Add a permission의

Amazon VPC Lattice

Amazon VPC Lattice: Add a permission의

Table of Contents

Amazon VPC Lattic란 무엇입니까?

Amazon VPC Lattice는 완전관리형 애플리케이션 네트워킹 서비스로, 애플리케이션용 서비스와 리소스를 연결, 보호 및 모니터링하는 데 사용합니다. VPC Lattice는 단일 Virtual Private Cloud(VPC)와 함께 사용하거나 계정 하나 이상의 여러 VPC에서 사용할 수 있습니다.

최신 애플리케이션은 HTTP API, 데이터베이스와 같은 리소스, DNS 및 IP 주소 엔드포인트로 구성된 사용자 지정 리소스와 같이 마이크로서비스라고도 하는 여러 개의 작은 모듈식 구성 요소로 구성될 수 있습니다. 현대화에는 장점이 있지만 이러한 마이크로서비스와 리소스를 연결할 때 네트워킹 복잡성과 과제가 발생할 수도 있습니다. 예를 들어 개발자가 서로 다른 팀에 분산되어 있는 경우 여러 계정 또는 VPCs.

VPC Lattice에서는 마이크로서비스를 서비스로 지칭하고 리소스를 리소스 구성으로만 나타냅니다. 다음은 VPC Lattice 사용 설명서에서 보고 사용할 용어입니다.

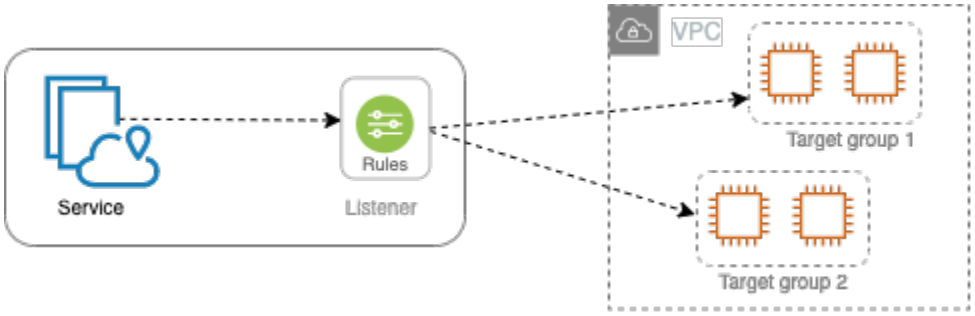
내용

핵심 구성 요소

Amazon VPC Lattice를 사용하려면 Lattice의 주요 구성 요소를 숙지해야 합니다.

서비스

> 독립적으로 배포할 수 있는 소프트웨어 단위로서 특정 작업이나 기능을 제공합니다. 서비스는 계정 또는 Virtual Private Cloud(VPC) 내에서 EC2 인스턴스 또는 ECS/EKS/Fargate 컨테이너에서 실행하거나 Lambda 함수로 실행할 수 있습니다. VPC Lattice 서비스는 대상 그룹, 리스너 및 규칙과 같은 구성 요소로 구성됩니다.



대상 그룹

애플리케이션이나 서비스를 실행하는 리소스 모음(일명 대상)입니다. Elastic Load Balancing에서 제공하는 대상 그룹과 비슷하지만 서로 바꿔서 사용할 수는 없습니다. 지원되는 대상 유형에는 EC2 인스턴스, IP 주소, Lambda 함수, Application Load Balancer, Amazon ECS 작업 및 Kubernetes 포드가 포함됩니다.

리스너

연결 요청을 확인하고 대상 그룹의 대상으로 이를 라우팅하는 프로세스입니다. 프로토콜과 포트 번호로 리스너를 구성합니다.

규칙

VPC Lattice 대상 그룹의 대상으로 요청을 전달하는 리스너의 기본 구성 요소입니다. 각 규칙은 우선 순위, 하나 이상의 작업, 하나 이상의 조건으로 구성됩니다. 규칙은 리스너가 클라이언트 요청을 라우팅하는 방법을 결정합니다.

Resource

리소스는 Amazon Relational Database Service(RDS) 데이터베이스, Amazon EC2 인스턴스, 애플리케이션 엔드포인트, 도메인 이름 대상 또는 IP 주소와 같은 엔터티입니다. AWS Resource Access Manager ()에서 리소스 공유를 생성하고, 리소스 게이트웨이를AWS RAM생성하고, 리소스 구성을 정의하여 VPC에서 리소스를 공유할 수 있습니다.

리소스 게이트웨이

리소스 게이트웨이는 리소스가 있는 VPC로 들어오는 지점입니다.

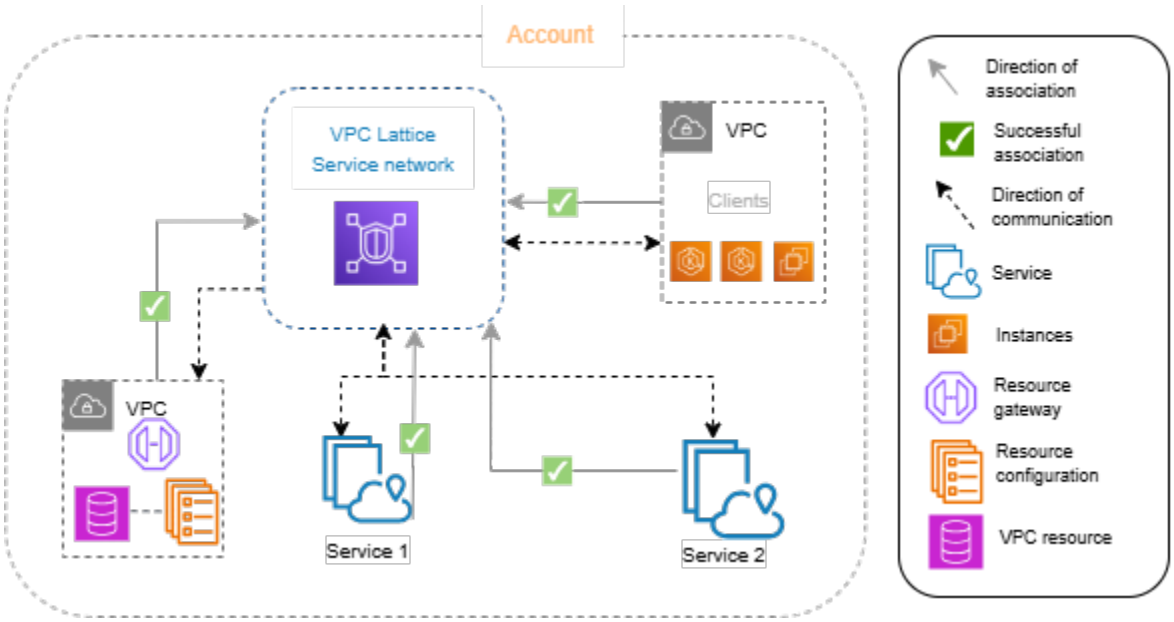
리소스 구성

리소스 구성은 단일 리소스 또는 리소스 그룹을 나타내는 논리적 객체입니다. 리소스는 IP 주소, 도메인 이름 대상 또는 Amazon RDS 데이터베이스일 수 있습니다.

서비스 네트워크

서비스 및 리소스 구성 모음의 논리적 경계입니다. 클라이언트는 서비스 네트워크와 연결된 VPC에 있을 수 있습니다. 동일한 서비스 네트워크에 연결된 클라이언트와 서비스는 권한이 있는 경우 서로 통신할 수 있습니다.

다음 그림에서는 VPC와 서비스가 동일한 서비스 네트워크에 연결되어 있으므로 클라이언트가 두 서비스와 통신할 수 있습니다.



서비스 디렉터리

소유하거나 계정과 공유하는 모든 VPC Lattice 서비스의 중앙 레지스트리입니다 AWS RAM.

인증 정책

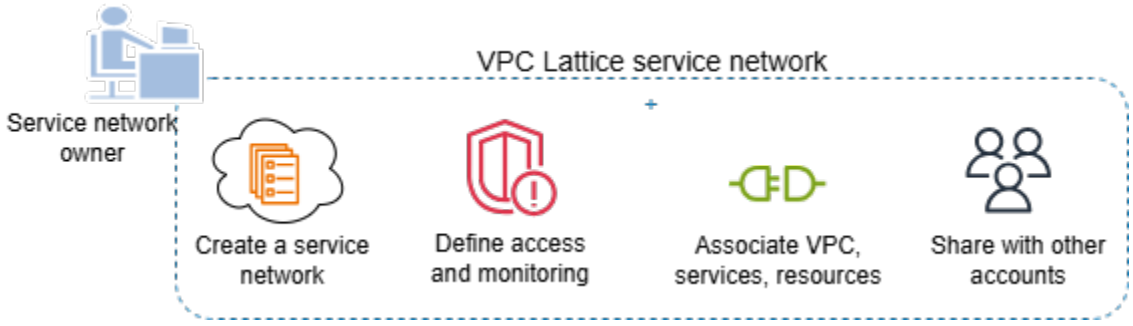
서비스에 대한 액세스를 정의하는 데 사용할 수 있는 세분화된 권한 부여 정책. 개별 서비스 또는 서비스 네트워크에 별도의 인증 정책을 추가할 수 있습니다. 예를 들어, Auto Scaling EC2 인스턴스 그룹에서 실행되는 결제 서비스가 AWS Lambda에서 실행되는 결제 서비스와 상호 작용하는 방식에 대한 정책을 생성할 수 있습니다.

인증 정책은 리소스 구성에서 지원되지 않습니다. 서비스 네트워크의 인증 정책은 서비스 네트워크의 리소스 구성에는 적용되지 않습니다.

역할 및 책임

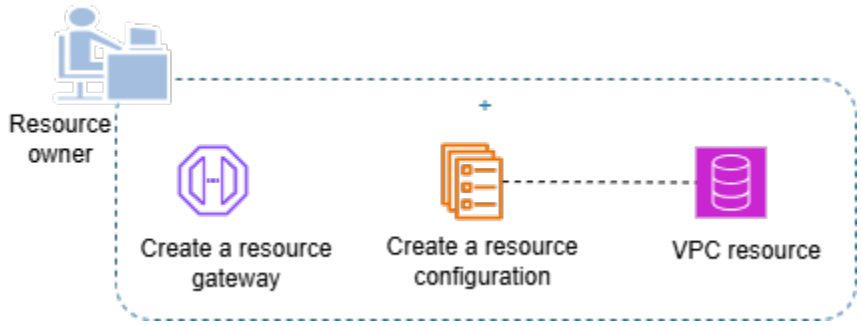
역할은 Amazon VPC Lattice 내에서 정보의 설정과 흐름을 누가 담당할지를 결정합니다. 일반적으로 서비스 네트워크 소유자와 서비스 소유자라는 두 가지 역할이 있으며, 둘의 책임은 중복될 수 있습니다.

서비스 네트워크 소유자 - 서비스 네트워크 소유자는 보통 조직의 네트워크 관리자 또는 클라우드 관리자입니다. 서비스 네트워크 소유자는 서비스 네트워크를 생성, 공유 및 제공합니다. 또한 VPC Lattice 내에서 서비스 네트워크 또는 서비스에 액세스할 수 있는 사용자를 관리합니다. 서비스 네트워크 소유자는 서비스 네트워크와 연결된 서비스에 대한 대략적인 액세스 설정을 정의할 수 있습니다. 인증 및 권한 부여 정책을 통해 클라이언트와 서비스 간의 통신을 관리하기 위해 이러한 제어를 사용합니다. 서비스 또는 리소스 구성이 서비스 네트워크 소유자의 계정과 공유되는 경우 서비스 네트워크 소유자는 서비스 또는 리소스 구성을 단일 또는 다중 서비스 네트워크와 연결할 수도 있습니다.



서비스 소유자 - 서비스 소유자는 일반적으로 조직의 소프트웨어 개발자입니다. 서비스 소유자는 VPC Lattice 내에서 서비스를 생성하고, 라우팅 규칙을 정의하고, 서비스를 서비스 네트워크와 연결합니다. 또한 세분화된 액세스 설정을 정의하여 인증되고 승인된 서비스와 클라이언트로만 액세스를 제한할 수 있습니다.

리소스 소유자 - 리소스 소유자는 일반적으로 조직의 소프트웨어 개발자이며 데이터베이스와 같은 리소스의 관리자 역할을 합니다. 리소스 소유자는 리소스에 대한 리소스 구성을 생성하고, 리소스 구성에 대한 액세스 설정을 정의하고, 리소스 구성을 서비스 네트워크와 연결합니다.



특성

VPC Lattice가 제공하는 핵심 기능은 다음과 같습니다.

서비스 검색

서비스 네트워크와 연결된 VPC의 모든 클라이언트 및 서비스는 동일한 서비스 네트워크 내의 다른 서비스와 통신할 수 있습니다. DNS는 VPC Lattice 엔드포인트를 통해 클라이언트-서비스와 서비스-서비스 트래픽을 전달합니다. 클라이언트는 서비스에 요청을 보내려고 할 때 서비스의 DNS 이름을 사용합니다. Route 53 Resolver는 트래픽을 VPC Lattice로 전송하고, VPC Lattice는 대상 서비스를 식별합니다.

연결

Client-to-service 및 client-to-resource 연결은 네트워크 인프라 내에서 AWS 설정됩니다. VPC를 서비스 네트워크에 연결하면 필요한 액세스 권한이 있는 경우 VPC 내의 모든 클라이언트가 서비스 네트워크의 서비스 및 리소스(리소스 구성을 통해)에 연결할 수 있습니다. VPC Lattice는 중복 CIDR 기술을 지원합니다.

온프레미스 액세스

VPC 엔드포인트(전원 제공)를 사용하여 VPC에서 서비스 네트워크에 대한 연결을 활성화할 수 있습니다 AWS PrivateLink. 서비스 네트워크 유형의 VPC 엔드포인트를 사용하면 Direct Connect 및 VPN을 통해 온프레미스 네트워크에서 서비스 네트워크의 서비스 및 리소스에 액세스할 수 있습니다. VPC 피어링을 통과하거나 VPC 엔드포인트를 통해 리소스 및 서비스에 액세스할 AWS Transit Gateway 수도 있는 트래픽입니다.

관찰성

VPC Lattice는 서비스 네트워크를 통과하는 각 요청과 응답에 대한 지표와 로그를 생성하여 애플리케이션을 모니터링하고 문제를 해결하는 데 도움이 됩니다. 기본적으로 지표는 서비스 소유자 계정에 게시됩니다. 서비스 소유자와 리소스 소유자는 로깅을 활성화하고 서비스 및 리소스에 대한 모든 클라이언트 액세스/요청에 대한 로그를 수신할 수 있습니다. 서비스 네트워크 소유자는 서비스

네트워크에 대한 로깅을 켜서 서비스 네트워크에 연결된 VPCs의 클라이언트에서 서비스 및 리소스에 대한 모든 액세스/요청을 로깅할 수도 있습니다.

VPC Lattice는 Amazon CloudWatch 로그 그룹, Firehose 전송 스트림 및 Amazon S3 버킷과 같은 서비스를 모니터링하고 문제를 해결하는 데 도움이 되는 도구와 함께 작동합니다.

보안

VPC Lattice는 네트워크의 여러 계층에서 방어 전략을 구현하는 데 사용할 수 있는 프레임워크를 제공합니다. 첫 번째 계층은 서비스, 리소스 구성, VPC 연결 및 서비스 네트워크 유형의 VPC 엔드포인트의 조합입니다. VPC 및 서비스 연결 또는 서비스 네트워크 유형의 VPC 엔드포인트가 없으면 클라이언트가 서비스에 액세스할 수 없습니다. 마찬가지로 VPC 및 리소스 구성과 서비스 연결 또는 서비스 네트워크 유형의 VPC 엔드포인트가 없으면 클라이언트가 리소스에 액세스할 수 없습니다.

두 번째 계층에서는 사용자가 VPC와 서비스 네트워크 간의 연결에 보안 그룹을 연결할 수 있습니다. 세 번째와 네 번째 계층은 서비스 네트워크 수준과 서비스 수준에서 개별적으로 적용할 수 있는 인증 정책입니다.

가용 영역 친화도

VPC Lattice는 트래픽 라우팅에 대한 가용 영역(AZ) 선호도를 지원합니다. 클라이언트가 VPC Lattice에 요청을 보내면 VPC Lattice는 클라이언트와 동일한 AZ의 서비스 또는 리소스에 대한 IP 주소로 응답합니다. 해당 AZ를 사용할 수 없는 경우 VPC Lattice는 다른 AZs. VPC Lattice에서 대상으로 라우팅은 AZs. 또한 VPC Lattice에는 AZ 간 데이터 전송 요금이 없습니다.

VPC Lattice에 액세스

다음 인터페이스 중 하나를 사용하여 VPC Lattice를 생성하고, 액세스하고, 관리할 수 있습니다.

- AWS Management Console – VPC Lattice에 액세스할 때 사용할 수 있는 웹 인터페이스를 제공합니다.
- AWS Command Line Interface (AWS CLI) - VPC Lattice를 포함한 다양한 AWS 서비스에 대한 명령을 제공합니다. AWS CLI 는 Windows, MacOS 및 Linux에서 지원됩니다. CLI에 대한 자세한 내용은 AWS Command Line Interface 단원을 참조하세요. API에 대한 자세한 내용은 Amazon VPC Lattice API 참조를 참조하세요.
- Kubernetes용 VPC Lattice 컨트롤러 – Kubernetes 클러스터의 VPC Lattice 리소스를 관리합니다. Kubernetes에서 VPC Lattice를 사용하는 방법에 대한 자세한 내용은 AWS 게이트웨이 API 컨트롤러 사용 설명서를 참조하세요.

- CloudFormation – AWS 리소스를 모델링하고 설정하는 데 도움을 줍니다. 자세한 내용은 Amazon VPC Lattice 리소스 유형 참조를 참조하세요.

VPC Lattice 서비스 엔드포인트

엔드포인트는 AWS 웹 서비스의 진입점 역할을 하는 URL입니다. VPC Lattice는 다음 엔드포인트 유형을 지원합니다.

- the section called “IPv4 엔드포인트”
- 듀얼 스택 엔드포인트(IPv4 및 IPv6를 모두 지원)

요청 시에, 사용할 엔드포인트를 지정할 수 있습니다. 엔드포인트를 지정하지 않으면 기본적으로 IPv4 엔드포인트가 사용됩니다. 다른 엔드포인트 유형을 사용하려면 요청에서 이를 지정해야 합니다. 이렇게 하는 방법의 예제는 the section called “엔드포인트 지정” 섹션을 참조하세요. 사용 가능한 엔드포인트 표는 Amazon VPC Lattice 엔드포인트를 참조하세요.

IPv4 엔드포인트

IPv4 엔드포인트는 IPv4 트래픽만 지원합니다. IPv4 엔드포인트는 모든 리전에 사용할 수 있습니다.

범용 엔드포인트 `vpc-lattice.amazonaws.com`을 지정하면 `us-east-1`의 엔드포인트를 사용합니다. 다른 리전을 사용하려면 연결된 엔드포인트를 지정해야 합니다. 예를 들어 `vpc-lattice.us-east-2.amazonaws.com`을 엔드포인트로 지정하면 요청이 `us-east-2` 엔드포인트로 전달됩니다.

IPv4 엔드포인트 이름에는 다음 명명 규칙이 사용됩니다.

- `vpc-lattice.`*`region`*`.amazonaws.com`

예를 들어 `eu-west-1` 리전의 IPv4 엔드포인트 이름은 `vpc-lattice.eu-west-1.amazonaws.com`입니다.

듀얼 스택(IPv4 및 IPv6) 엔드포인트

듀얼 스택 엔드포인트는 IPv4 트래픽과 IPv6 트래픽을 모두 지원합니다. 듀얼 스택 엔드포인트는 모든 리전에서 사용할 수 있습니다. 듀얼 스택 엔드포인트에 요청하는 경우, 엔드포인트 URL이 네트워크 및 클라이언트에서 사용하는 프로토콜에 따라 IPv6 또는 IPv4 주소로 확인됩니다.

이중 스택 엔드포인트 이름에는 다음 명명 규칙이 사용됩니다.

- vpc-lattice.*region*.api.aws

예를 들어 eu-west-1 리전의 이중 스택 엔드포인트 이름은 vpc-lattice.eu-west-1.api.aws입니다.

엔드포인트 지정

다음 예제에서는 용를 사용하여 us-east-2 리전의 엔드포인트 AWS CLI 를 지정하는 방법을 보여줍니다vpc-lattice.

- IPv4

```
aws vpc-lattice get-service --service-identifier svc-0285b53b2eEXAMPLE --region us-east-2 --endpoint-url https://vpc-lattice.us-east-2.amazonaws.com
```

- 듀얼 스택

```
aws vpc-lattice get-service --service-identifier svc-0285b53b2eEXAMPLE --region us-east-2 --endpoint-url https://vpc-lattice.us-east-2.api.aws
```

가격 책정

VPC Lattice를 사용하면 서비스가 프로비저닝된 시간, 각 서비스를 통해 전송되는 데이터의 양, 요청 수에 대한 비용을 지불할 수 있습니다. 리소스 소유자는 각 리소스와 주고받는 데이터에 대한 비용을 지불합니다. 서비스 네트워크 소유자는 서비스 네트워크에 연결된 리소스 구성에 대해 시간당 요금을 지불합니다. 서비스 네트워크에 연결된 VPC가 있는 소비자는 VPC에서 서비스 네트워크의 리소스로 송수신되는 데이터에 대한 비용을 지불합니다. 자세한 내용은 Amazon VPC Lattice 요금을 참조하세요.

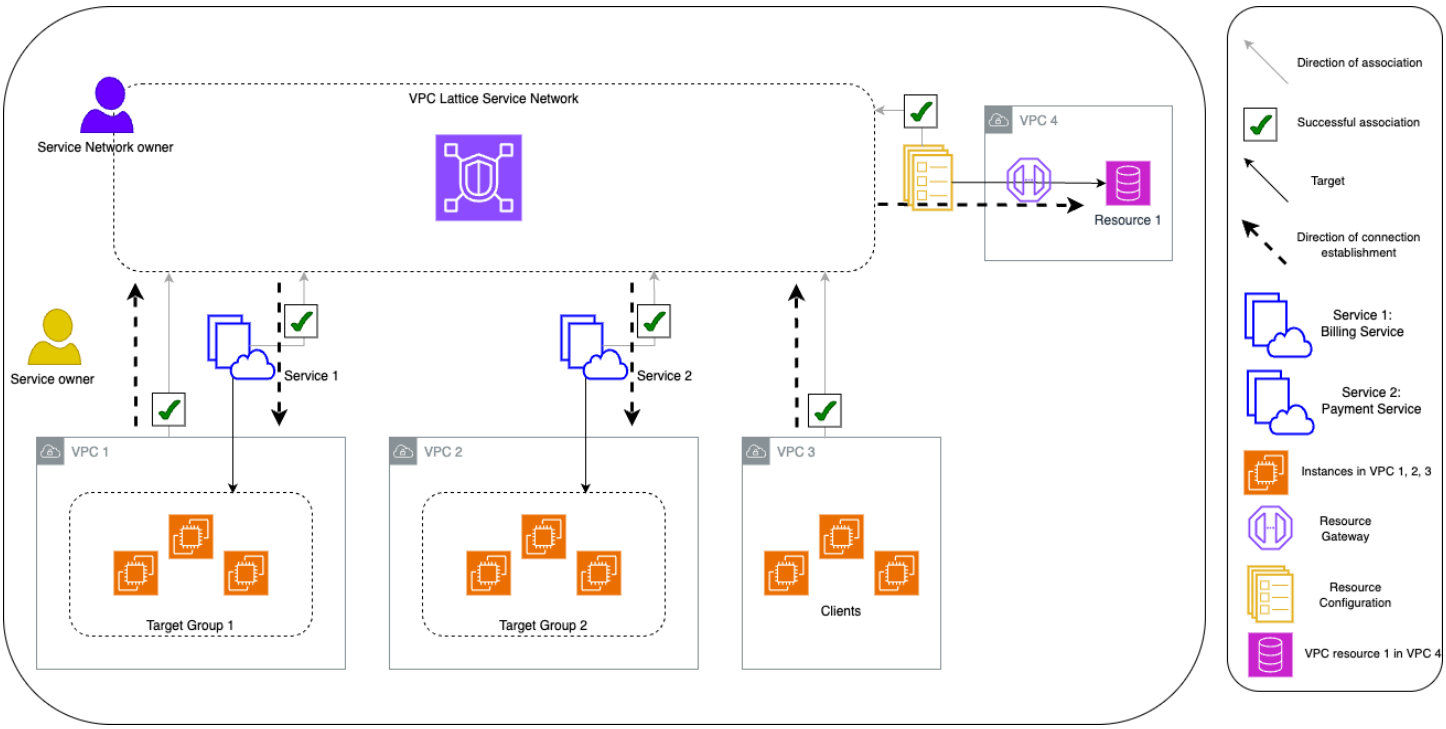
VPC Lattic의 작동 방식

VPC Lattice는 모든 서비스와 리소스를 쉽고 효과적으로 검색, 보호, 연결 및 모니터링할 수 있도록 설계되었습니다. VPC Lattice 내의 각 구성 요소는 서비스 네트워크와의 연결 및 액세스 설정을 기반으로 서비스 네트워크 내에서 단방향 또는 양방향으로 통신합니다. 액세스 설정은 이 통신에 필요한 인증 및 권한 부여 정책으로 구성됩니다.

다음 요약에서는 VPC Lattice 내 구성 요소 간 통신에 대해 설명합니다.

- VPC 연결과 서비스 네트워크 유형의 VPC 엔드포인트를 통해 VPC를 서비스 네트워크에 연결할 수 있는 두 가지 방법이 있습니다.
- 서비스 네트워크와 연결된 서비스 및 리소스는 VPCs가 서비스 네트워크에도 연결된 클라이언트로부터 요청을 수신할 수 있습니다.
- 클라이언트는 동일한 서비스 네트워크에 연결된 VPC에 있는 경우에만 서비스 네트워크와 연결된 서비스 및 리소스에 요청을 보낼 수 있습니다. VPC 피어링 연결, 전송 게이트웨이, Direct Connect 또는 VPN을 통과하는 클라이언트 트래픽은 VPC가 VPC 엔드포인트를 통해 서비스 네트워크에 연결된 경우에만 리소스 및 서비스에 도달할 수 있습니다.
- 서비스 네트워크와 연결된 VPCs의 서비스 대상도 클라이언트이며 서비스 네트워크와 연결된 다른 서비스 및 리소스에 요청을 보낼 수 있습니다.
- 서비스 네트워크와 연결되지 않은 VPCs의 서비스 대상은 클라이언트가 아니며 서비스 네트워크와 연결된 다른 서비스 및 리소스로 요청을 보낼 수 없습니다.
- 리소스가 있지만 VPCs가 서비스 네트워크와 연결되지 않은 VPC의 클라이언트는 클라이언트가 아니며 서비스 네트워크와 연결된 다른 서비스 및 리소스로 요청을 보낼 수 없습니다.

다음 흐름 다이어그램에서는 예시 시나리오를 사용하여 VPC Lattice 내 구성 요소 간의 정보 흐름과 통신 방향을 설명합니다. 서비스 네트워크와 관련된 서비스는 두 가지가 있습니다. 서비스와 모든 VPCs 모두 서비스 네트워크와 동일한 계정에 생성되었습니다. 두 서비스 모두 서비스 네트워크에서 오는 트래픽을 허용하도록 구성되어 있습니다.



서비스 1은 VPC 1의 대상 그룹 1에 등록된 인스턴스 그룹에서 실행되는 결제 애플리케이션입니다. 서비스 2는 VPC 2의 대상 그룹 2에 등록된 인스턴스 그룹에서 실행되는 결제 애플리케이션입니다. VPC 3은 동일한 계정에 있으며 클라이언트는 있지만 서비스는 없습니다. 리소스 1은 VPC 4에 고객 데이터가 있는 데이터베이스입니다.

다음 목록은 VPC Lattice의 일반적인 작업 워크플로를 순서대로 설명합니다.

1. 서비스 네트워크 생성

   서비스 네트워크 소유자는 서비스 네트워크를 생성합니다.

2. 서비스 생성

   서비스 소유자는 서비스 1과 서비스 2 등 각각의 서비스를 생성합니다. 생성 과정에서 서비스 소유자는 리스너를 추가하고 각 서비스의 대상 그룹으로 요청을 라우팅하기 위한 규칙을 정의합니다.

3. 라우팅 정의

   서비스 소유자는 각 서비스의 대상 그룹(대상 그룹 1 및 대상 그룹 2)을 생성합니다. 서비스가 실행되는 대상 인스턴스를 지정하여이 작업을 수행합니다. 또한 이러한 대상이 있는 VPC를 지정합니다.

   앞의 다이어그램에서 실선 화살표는 트래픽을 대상 그룹으로 라우팅하는 서비스와 리소스로 라우팅하는 리소스 구성을 나타냅니다.

VPC Lattice는 트래픽 라우팅에 대한 가용 영역(AZ) 선호도를 지원합니다. 클라이언트가 VPC Lattice에 요청을 보내면 VPC Lattice는 클라이언트와 동일한 AZ의 서비스 또는 리소스에 대한 IP 주소로 응답합니다. 해당 AZ를 사용할 수 없는 경우 VPC Lattice는 다른 AZs. VPC Lattice에서 대상으로 라우팅은 AZs. 또한 VPC Lattice에는 AZ 간 데이터 전송 요금이 없습니다.

4. 서비스를 서비스 네트워크와 연결

   서비스 네트워크 소유자 또는 서비스 소유자는 서비스를 서비스 네트워크와 연결합니다. 화살표로 연결을 표시하며 화살표에는 서비스의 서비스 네트워크를 가리키는 체크 표시가 있습니다. 서비스를 서비스 네트워크에 연결하면 서비스 네트워크에 연결된 VPCs의 서비스 네트워크 및 클라이언트와 연결된 다른 서비스에서 해당 서비스를 검색할 수 있게 됩니다.

   서비스 네트워크와 대상 그룹 간의 파선 화살표는 연결 설정 방향을 보여줍니다. 서비스 네트워크를 사용하여 트래픽 흐름을 클라이언트로 반환합니다. 반환 트래픽을 나타내는 화살표는이 다이어그램에 포함되지 않습니다.

5. 리소스 게이트웨이 생성

   리소스 소유자는 클라이언트에서 리소스 1로의 연결을 활성화할 수 있도록 VPC 4에 리소스 게이트웨이를 생성합니다.

6. 리소스 구성 생성

   리소스 소유자는 리소스 1을 나타내는 리소스 구성을 생성하고 리소스 1에 대한 리소스 게이트웨이를 지정합니다.

7. 리소스 구성을 서비스 네트워크와 연결

   서비스 네트워크 소유자 또는 리소스 소유자는 리소스 구성을 서비스 네트워크와 연결합니다. 연결은 리소스 구성에서 서비스 네트워크를 가리키는 확인 표시가 있는 화살표로 표시됩니다. 리소스 구성을 서비스 네트워크에 연결하면 해당 리소스 구성을 서비스 네트워크에 연결된 VPCs의 서비스 네트워크 및 클라이언트와 연결된 다른 서비스에서 검색할 수 있게 됩니다.

   서비스 네트워크에서 리소스로의 파선 화살표는 클라이언트로부터 요청을 수신하는 리소스를 나타냅니다. 서비스 네트워크를 사용하여 트래픽 흐름을 클라이언트로 반환합니다. 반환 트래픽을 나타내는 화살표는이 다이어그램에 포함되지 않습니다.

8. VPCs 서비스 네트워크에 연결

   VPCs VPC를 서비스 네트워크에 연결하거나 VPC 엔드포인트를 생성하는 두 가지 방법으로 서비스 네트워크에 연결할 수 있습니다. 여기서 서비스 네트워크 소유자는 VPC 1 및 VPC 3을 서비스 네트워크와 연결합니다. 연결은 서비스 네트워크를 가리키는 확인 표시가 있는 화살표를 사용하여 표시

됩니다. 이러한 연결을 사용하면 VPC의 모든 리소스가 클라이언트 역할을 할 수 있으며 서비스 네트워크 내의 서비스에 요청할 수 있습니다. VPC 1과 서비스 네트워크 간의 파선 화살표는 연결 설정 방향을 보여줍니다. 서비스 네트워크는 서비스 1 대상 그룹이 대상으로 하는 리소스에 대한 연결만 시작합니다. VPC 1의 모든 리소스는 클라이언트 역할을 하고 서비스 네트워크 서비스 및 리소스에 대한 연결을 시작할 수 있습니다.

VPC 2에는 연결을 나타내는 화살표 또는 확인 표시가 없습니다. 즉, 서비스 네트워크 소유자 또는 서비스 소유자는 VPC 2를 서비스 네트워크와 연결하지 않았습니다. 이 예시에서 서비스 2는 요청을 수신하고 동일한 요청을 사용하여 응답을 보내기만 하면 되기 때문입니다. 다시 말해, 서비스 2의 대상은 클라이언트가 아니므로 서비스 네트워크의 다른 서비스에 요청을 할 필요가 없습니다.

마찬가지로 VPC 4에는 연결을 나타내는 화살표 또는 확인 표시가 없습니다. 즉, 서비스 네트워크 소유자 또는 리소스 소유자가 VPC 4를 서비스 네트워크에 연결하지 않았습니다. 리소스 1은 동일한 요청을 사용하여 요청을 수신하고 응답을 보내기 때문입니다. 서비스 네트워크의 다른 서비스 및 리소스에는 요청할 수 없습니다.

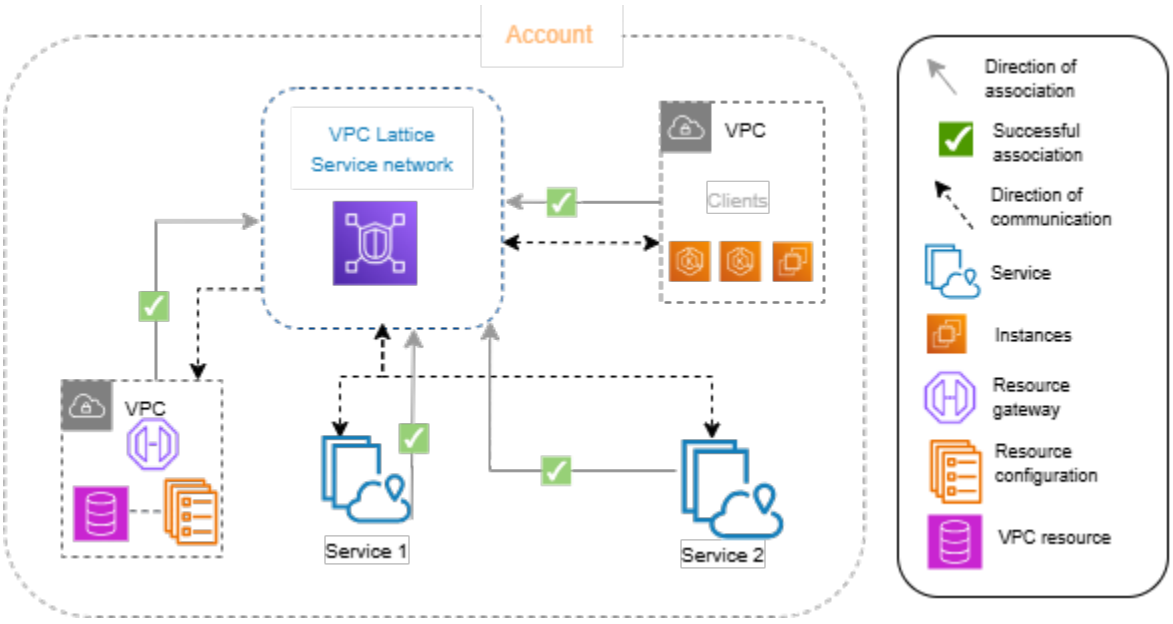
요약하면 다음 시나리오가 진행 다이어그램에 나와 있습니다.

- 수신이 있는 VPCs VPC Lattice에서 리소스로만 연결됩니다. VPC 2와 VPC 4는 이러한 시나리오를 나타냅니다.
- 송신 기능이 있는 VPC는 리소스에서 VPC Lattice로만 연결됩니다. VPC 3은이 시나리오를 나타냅니다.
- VPC Lattice에서 리소스로의 수신 연결과 리소스에서 VPC Lattice로의 송신 연결이 있는 VPC입니다. VPC 1은이 시나리오를 나타냅니다.

VPC Lattice의 서비스 네트워크

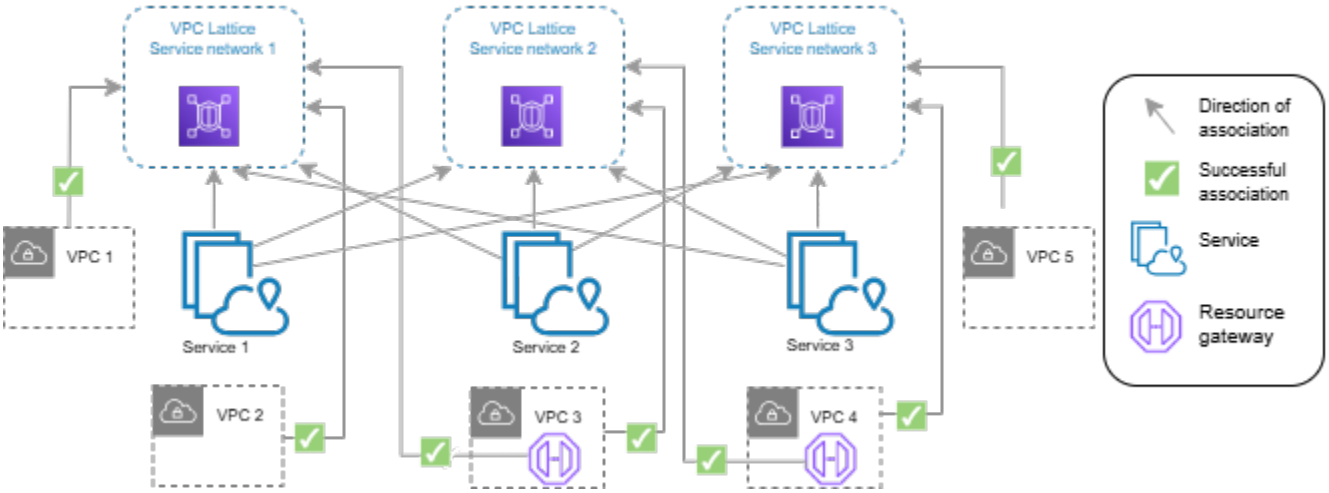
서비스 네트워크는 서비스 및 리소스 구성 모음의 논리적 경계입니다. 네트워크와 연결된 서비스 및 리소스 구성은 검색, 연결, 접근성 및 관찰성에 대한 권한을 부여받을 수 있습니다. 네트워크의 서비스 및 리소스 구성을 요청하려면 서비스 또는 클라이언트가 연결 또는 VPC 엔드포인트를 통해 서비스 네트워크에 연결된 VPC에 있어야 합니다.

다음 다이어그램은 Amazon VPC Lattice 내의 일반적인 서비스의 주요 구성 요소를 보여줍니다. 화살표의 체크 표시는 서비스와 VPC가 서비스 네트워크와 연결되어 있음을 나타냅니다. 서비스 네트워크에 연결된 VPC의 클라이언트는 서비스 네트워크를 통해 두 서비스와 통신할 수 있습니다.



하나 이상의 서비스 및 리소스 구성을 여러 서비스 네트워크에 연결할 수 있습니다. 하나의 서비스 네트워크에 여러 VPCs 연결할 수도 있습니다. 연결을 통해 VPC를 하나의 서비스 네트워크에만 연결할 수 있습니다. VPC를 여러 서비스 네트워크에 연결하려면 서비스 네트워크 유형의 VPC 엔드포인트를 사용할 수 있습니다. 서비스 네트워크 유형의 VPC 엔드포인트에 대한 자세한 내용은 AWS PrivateLink 사용 설명서를 참조하세요.

다음 다이어그램에서 화살표는 서비스와 서비스 네트워크 간의 연결뿐 아니라 VPC와 서비스 네트워크 간의 연결을 나타냅니다. 여러 서비스가 여러 서비스 네트워크에 연결되어 있고 여러 VPC가 각 서비스 네트워크에 연결되어 있음을 알 수 있습니다. 각 VPC는 서비스 네트워크에 정확히 하나의 연결을 갖습니다. 그러나 VPC 3과 VPC 4는 두 개의 서비스 네트워크에 연결됩니다. VPC 3은 VPC 엔드포인트를 통해 서비스 네트워크 1에 연결합니다. 마찬가지로 VPC 4는 VPC 엔드포인트를 통해 서비스 네트워크 2에 연결됩니다.



자세한 내용은 Amazon VPC Lattice의 할당량 단원을 참조하십시오.

내용

- VPC Lattice 서비스 네트워크 생성
- VPC Lattice 서비스 네트워크의 연결 관리
- VPC Lattice 서비스 네트워크의 액세스 설정 편집
- VPC Lattice 서비스 네트워크의 모니터링 세부 정보 편집
- VPC Lattice 서비스 네트워크의 태그 관리
- VPC Lattice 서비스 네트워크 삭제

VPC Lattice 서비스 네트워크 생성

콘솔을 사용하여 서비스 네트워크를 생성하고 필요할 경우 서비스, 연결, 액세스 설정, 액세스 로그로 서비스 네트워크를 구성할 수 있습니다.

콘솔을 사용하여 서비스 네트워크를 생성하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스 네트워크를 선택합니다.
3. 서비스 네트워크 생성을 선택합니다.
4. 식별자에 이름, 선택적 설명, 선택적 태그를 입력합니다. 이름은 3~63자 이내로 작성해야 합니다. 소문자, 숫자 및 하이픈만 포함될 수 있습니다. 이름은 글자 또는 숫자로 시작하고 끝나야 합니다.

하이픈을 연속하여 사용하지 마세요. 설명의 길이는 최대 256자입니다. 태그를 추가하려면 새 태그 추가를 선택하고 키 이름 및 키 값을 지정합니다.

5. (선택 사항) 서비스를 연결하려면 서비스 연결, 서비스에서 서비스를 선택합니다. 목록에는 계정에 있는 서비스와 다른 계정에서 사용자와 공유하는 모든 서비스가 포함됩니다. 목록에 서비스가 없는 경우 VPC Lattice 서비스 생성을 선택하여 서비스를 생성할 수 있습니다.

   또는 서비스 네트워크를 생성한 후 서비스를 연결하려면 the section called “서비스 네트워크 서비스 연결 관리”를 참조하세요.

6. (선택 사항) 리소스 구성을 연결하려면 리소스 구성 연결, 리소스 구성에서 리소스 구성 서비스를 선택합니다. 목록에는 계정에 있는 리소스 구성과 다른 계정에서 공유되는 모든 리소스 구성이 포함됩니다. 목록에 리소스 구성이 없는 경우 Amazon VPC Lattice 리소스 구성 생성을 선택하여 리소스 구성을 생성할 수 있습니다.

   또는 서비스 네트워크를 생성한 후 리소스 구성을 연결하려면 섹션을 참조하세요the section called “서비스 네트워크 리소스 연결 관리”.

7. (선택 사항) VPC를 연결하려면 VPC 연결 추가를 선택합니다. VPC에서 연결할 VPC를 선택하고 보안 그룹에서 최대 5개의 보안 그룹을 선택합니다. 보안 그룹을 생성하려면 보안 그룹 생성을 선택합니다.

   또는이 단계를 건너뛰고 VPC 엔드포인트(전원 제공)를 사용하여 VPC를 서비스 네트워크에 연결할 수 있습니다 AWS PrivateLink. 자세한 내용은 AWS PrivateLink 사용 설명서의 서비스 네트워크 액세스를 참조하세요.

8. 서비스 네트워크를 생성할 때 서비스 네트워크를 다른 계정과 공유할지 여부를 결정해야 합니다. 서비스 네트워크를 생성한 후에는 선택 항목을 변경할 수 없으며 변경할 수 없습니다. 공유를 허용하도록 선택하면를 통해 서비스 네트워크를 다른 계정과 공유할 수 있습니다 AWS Resource Access Manager.

   서비스 네트워크를 다른 계정과 공유하려면 AWS RAM 리소스 공유에서 리소스 공유를 선택합니다.

   리소스 공유를 생성하려면 AWS RAM 콘솔로 이동하여 리소스 공유 생성을 선택합니다.

9. 네트워크 액세스의 경우, 연결된 VPC의 클라이언트가 이 서비스 네트워크의 서비스에 액세스하도록 하려면 기본 인증 유형을 없음으로 둘 수 있습니다. 인증 정책을 적용하여 서비스에 대한 액세스를 제어하려면 AWS IAM을 선택하고 인증 정책에 대해 다음 중 하나를 수행합니다.

   - 입력 필드에 정책을 입력합니다. 복사하여 붙여넣을 수 있는 정책 예시를 보려면 정책 예시를 선택합니다.

- 정책 템플릿 적용을 선택하고 인증 및 비인증 액세스 허용 템플릿을 선택합니다. 이 템플릿을 사용하면 다른 계정의 클라이언트가 요청에 서명해서(인증되었다는 의미) 또는 익명으로(인증되지 않았다는 의미) 서비스에 액세스할 수 있습니다.
- 정책 템플릿 적용을 선택하고 인증된 액세스만 허용 템플릿을 선택합니다. 이 템플릿을 사용하면 다른 계정의 클라이언트가 요청에 서명해야만(인증되었다는 의미) 서비스에 액세스할 수 있습니다.

10. (선택 사항) 액세스 로그를 켜려면 액세스 로그 토글 스위치를 선택하고 다음과 같이 액세스 로그의 대상을 지정합니다.
    - CloudWatch 로그 그룹을 클릭하여 CloudWatch 로그 그룹을 선택합니다. 로그 그룹을 생성하려면 CloudWatch에서 로그 그룹 생성을 선택합니다.
    - S3 버킷을 선택하고 접두사를 포함한 S3 버킷 경로를 입력합니다. S3 버킷을 검색하려면 S3 찾아보기를 선택합니다.
    - Kinesis Data Firehose 전송 스트림을 클릭하고 전송 스트림을 선택합니다. 전송 스트림을 생성하려면 Kinesis에서 전송 스트림 생성을 선택합니다.
11. (선택 사항) 서비스 네트워크를 다른 계정과 공유하려면 AWS RAM 리소스 공유에서 리소스 공유를 선택합니다. 리소스 공유를 생성하려면 RAM 콘솔에서 리소스 공유 생성을 선택합니다.
12. 요약 섹션에서 구성을 검토한 다음 서비스 네트워크 생성을 선택합니다.

를 사용하여 서비스 네트워크를 생성하려면 AWS CLI

create-service-network 명령을 사용합니다. 이 명령은 기본 서비스 네트워크만 생성합니다. 모든 기능을 갖춘 서비스 네트워크를 만들려면 서비스 연결, VPC 연결 및 액세스 설정을 생성하는 명령도 사용해야 합니다.

VPC Lattice 서비스 네트워크의 연결 관리

서비스 또는 리소스 구성을 서비스 네트워크에 연결하면 서비스 네트워크에 연결된 VPCs의 클라이언트가 서비스 및 리소스 구성을 요청할 수 있습니다. VPC를 서비스 네트워크에 연결하면 해당 VPC 내의 모든 대상이 클라이언트가 되고 서비스 네트워크의 다른 서비스 및 리소스 구성과 통신할 수 있습니다.

서비스 네트워크 리소스 연결의 프라이빗 DNS 지원 속성은 서비스 네트워크 엔드포인트 및 서비스 네트워크 VPC 연결의 프라이빗 DNS 지원 속성을 재정의합니다.

서비스 네트워크 소유자가 서비스 네트워크 리소스 연결을 생성하고 프라이빗 DNS를 활성화하지 않는 경우 VPC Lattice는 서비스 네트워크 엔드포인트 또는 서비스 네트워크 VPCs 연결에서 프라이빗

DNS가 활성화되어 있더라도 서비스 네트워크가 연결된 VPC에서 해당 리소스 구성에 대한 프라이빗 호스팅 영역을 프로비저닝하지 않습니다.

내용

서비스 네트워크 서비스 연결 관리

계정에 있는 서비스 또는 다른 계정에서 사용자와 공유하는 서비스를 연결할 수 있습니다. 이 단계는 서비스 네트워크를 만드는 동안 수행할 수 있는 선택적 단계입니다. 하지만 서비스를 연결하기 전까지는 서비스 네트워크가 제대로 작동하지 않습니다. 계정에 필요한 액세스 권한이 있는 경우 서비스 소유자는 서비스를 서비스 네트워크에 연결할 수 있습니다. 자세한 내용은 VPC Lattice에 대한 자격 증명 기반 정책 예시 단원을 참조하십시오.

서비스 연결을 삭제하면 해당 서비스는 더 이상 서비스 네트워크의 다른 서비스에 연결할 수 없습니다.

콘솔을 사용하여 서비스 연결을 관리하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스 네트워크를 선택합니다.
3. 서비스 네트워크의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 서비스 연결 탭을 선택합니다.
5. 연결을 생성하려면 다음을 수행합니다.
    a. 연결 생성을 선택합니다.
    b. 서비스에서 서비스를 선택합니다. 서비스를 생성하려면 Amazon VPC Lattice 서비스 생성을 선택합니다.
    c. (선택 사항) 태그를 추가하려면 서비스 연결 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.
    d. 변경 사항 저장을 선택합니다.
6. 연결을 삭제하려면 연결의 확인란을 선택한 다음 작업, 서비스 연결 삭제를 선택합니다. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 삭제를 선택합니다.

를 사용하여 서비스 연결을 생성하려면 AWS CLI

create-service-network-service-association 명령을 사용합니다.

를 사용하여 서비스 연결을 삭제하려면 AWS CLI

delete-service-network-service-association 명령을 사용합니다.

서비스 네트워크 리소스 연결 관리

리소스 구성은 단일 리소스 또는 리소스 그룹을 나타내는 논리적 객체입니다. 계정에 상주하는 리소스 구성 또는 다른 계정에서 공유되는 리소스 구성을 연결할 수 있습니다. 이 단계는 서비스 네트워크를 만드는 동안 수행할 수 있는 선택적 단계입니다. 리소스 구성 소유자는 계정에 필요한 액세스 권한이 있는 경우 리소스 구성을 서비스 네트워크에 연결할 수 있습니다. 자세한 내용은 VPC Lattice의 자격 증명 기반 정책 예제를 참조하세요.

서비스 네트워크와 리소스 구성 간의 연결 관리

서비스 네트워크와 리소스 구성 간의 연결을 생성하거나 삭제할 수 있습니다.

콘솔을 사용하여 리소스 구성 연결을 관리하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 PrivateLink 및 Lattice에서 서비스 네트워크를 선택합니다.
3. 서비스 네트워크의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 리소스 구성 연결 탭을 선택합니다.
5. 연결을 생성하려면 다음을 수행합니다.
    a. 연결 생성을 선택합니다.
    b. 리소스 구성에서 리소스 구성을 선택합니다.
    c. DNS 이름에서 프라이빗 DNS 활성화를 선택하여 VPC Lattice가 리소스 구성의 도메인 이름을 기반으로 리소스 구성 연결에 프라이빗 호스팅 영역을 프로비저닝하도록 허용합니다.
    d. (선택 사항) 태그를 추가하려면 서비스 연결 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.
    e. 변경 사항 저장을 선택합니다.
6. 연결을 삭제하려면 연결의 확인란을 선택한 다음 작업, 삭제를 선택합니다. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 삭제를 선택합니다.

를 사용하여 리소스 구성 연결을 생성하려면 AWS CLI

create-service-network-resource-association 명령을 사용합니다.

를 사용하여 리소스 구성 연결을 삭제하려면 AWS CLI

delete-service-network-resource-association 명령을 사용합니다.

서비스 네트워크 VPC 연결 관리

클라이언트가 서비스 네트워크와 연결된 VPCs에 있는 경우 클라이언트는 서비스 네트워크와 연결된 리소스 구성에 지정된 서비스 및 리소스에 요청을 보낼 수 있습니다. VPC 피어링 연결 또는 전송 게이트웨이를 통과하는 클라이언트 트래픽은 서비스 네트워크 유형의 VPC 엔드포인트를 사용하는 서비스 네트워크를 통해서만 허용됩니다.

VPC 연결은 서비스 네트워크를 생성할 때 선택할 수 있는 단계입니다. 계정에 필요한 액세스 권한이 있는 경우 네트워크 소유자는 VPC를 서비스 네트워크에 연결할 수 있습니다. 자세한 내용은 VPC Lattice에 대한 자격 증명 기반 정책 예시 단원을 참조하십시오.

리소스 구성에 대한 VPC 연결을 생성할 때 프라이빗 DNS 기본 설정을 지정할 수 있습니다. 이 기본 설정을 사용하면 VPC Lattice가 리소스 소비자를 대신하여 프라이빗 호스팅 영역을 프로비저닝할 수 있습니다. 자세한 내용은 the section called “리소스 공급자의 사용자 지정 도메인 이름” 단원을 참조하십시오.

VPC 연결을 삭제하면 VPC의 클라이언트는 더 이상 서비스 네트워크의 서비스에 연결할 수 없습니다.

콘솔을 사용하여 VPC 연결을 관리하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스 네트워크를 선택합니다.
3. 서비스 네트워크의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. VPC 연결 탭을 선택합니다.
5. VPC 연결을 생성하려면 다음을 수행합니다.
    a. VPC 연결 생성을 선택합니다.
    b. VPC 연결 추가를 선택합니다.
    c. VPC에서 VPC를 선택하고 보안 그룹에서 최대 5개의 보안 그룹을 선택합니다. 보안 그룹을 생성하려면 보안 그룹 생성을 선택합니다.

d. (선택 사항) VPC Lattice가 리소스 구성의 도메인 이름을 기반으로 프라이빗 호스팅 영역을 프로비저닝하도록 허용하려면 DNS 이름에서 DNS 이름 활성화를 선택하고 다음을 수행합니다.

   i. 프라이빗 DNS 기본 설정에서 기본 설정을 선택합니다.

      모든 도메인을 선택하면 VPC Lattice는 리소스 구성의 사용자 지정 도메인 이름에 대해 프라이빗 호스팅 영역을 프로비저닝합니다.

   ii. (선택 사항) 확인 및 지정된 도메인 또는 지정된 도메인을 선택하는 경우 VPC Lattice가 호스팅 영역을 프로비저닝할 도메인 목록을 쉼표로 구분하여 입력합니다. VPC Lattice는 프라이빗 도메인 목록과 일치하는 경우에만 호스팅 영역을 프로비저닝합니다. 와일드카드 일치를 사용할 수 있습니다.

e. (선택 사항) 태그를 추가하려면 VPC 연결 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.

f. 변경 사항 저장을 선택합니다.

6. 연결의 보안 그룹을 편집하려면 연결의 확인란을 선택한 다음 작업, 보안 그룹 편집을 선택합니다. 필요에 따라 보안 그룹을 추가하고 제거합니다.
7. 연결을 삭제하려면 연결의 확인란을 선택한 다음 작업, VPC 연결 삭제를 선택합니다. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 삭제를 선택합니다.

를 사용하여 VPC 연결을 생성하려면 AWS CLI

create-service-network-vpc-association 명령을 사용합니다.

를 사용하여 VPC 연결의 보안 그룹을 업데이트하려면 AWS CLI

update-service-network-vpc-association 명령을 사용합니다.

를 사용하여 VPC 연결을 삭제하려면 AWS CLI

delete-service-network-vpc-association 명령을 사용합니다.

서비스 네트워크 VPC 엔드포인트 연결 관리

클라이언트는 VPC의 VPC 엔드포인트(제공자 AWS PrivateLink)를 통해 리소스 구성에 지정된 서비스 및 리소스에 요청을 보낼 수 있습니다. 서비스 네트워크 유형의 VPC 엔드포인트는 VPC를 서비스 네트워크에 연결합니다. VPC 피어링 연결, Transit Gateway, Direct Connect 또는 VPN을 통해 VPC 외부에서 오는 클라이언트 트래픽은 VPC 엔드포인트를 사용하여 서비스 및 리소스 구성에 도달할 수 있습니다. VPC 엔드포인트를 사용하면 VPC를 여러 서비스 네트워크에 연결할 수 있습니다. VPC에서 VPC

엔드포인트를 생성하면 VPC의 IP 주소(관리형 접두사 목록의 IP 주소 아님)가 서비스 네트워크에 대한 연결을 설정하는 데 사용됩니다.

리소스 구성에 대한 VPC 연결을 생성할 때 프라이빗 DNS 기본 설정을 지정할 수 있습니다. 이 기본 설정을 사용하면 VPC Lattice가 리소스 소비자를 대신하여 프라이빗 호스팅 영역을 프로비저닝할 수 있습니다. 자세한 내용은 the section called “리소스 공급자의 사용자 지정 도메인 이름” 단원을 참조하십시오.

콘솔을 사용하여 VPC 엔드포인트 연결을 관리하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스 네트워크를 선택합니다.
3. 서비스 네트워크의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 엔드포인트 연결 탭을 선택하여 서비스 네트워크에 연결된 VPC 엔드포인트를 봅니다.
5. VPC 엔드포인트의 엔드포인트 ID를 선택하여 세부 정보 페이지를 엽니다. 그런 다음 VPC 엔드포인트 연결을 수정하거나 삭제합니다.

콘솔을 사용하여 새 VPC 엔드포인트 연결을 생성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 엔드포인트를 선택합니다.
3. 엔드포인트 생성을 선택합니다.
4. 유형에서 서비스 네트워크를 선택합니다.
5. VPC에 연결할 서비스 네트워크를 선택합니다.
6. VPC, 서브넷 및 보안 그룹을 선택합니다.
7. (선택 사항) 프라이빗 DNS를 활성화하려면 프라이빗 DNS 활성화를 선택합니다.
8. (선택 사항) 태그를 추가하려면 VPC 연결 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.
9. 엔드포인트 생성을 선택합니다.

서비스 네트워크에 연결하는 방법에 대한 VPC 엔드포인트에 대한 자세한 내용은 AWS PrivateLink 사용 설명서의 서비스 네트워크 액세스를 참조하세요.

VPC Lattice 서비스 네트워크의 액세스 설정 편집

액세스 설정을 통해 서비스 네트워크에 대한 클라이언트 액세스를 구성하고 관리할 수 있습니다. 액세스 설정에는 인증 유형과 인증 정책이 포함됩니다. 인증 정책은 VPC Lattice 내의 서비스로 흐르는 트래픽을 인증하고 승인하는 데 도움이 됩니다. 서비스 네트워크의 액세스 설정은 서비스 네트워크에 연결된 리소스 구성에는 적용되지 않습니다.

인증 정책은 서비스 네트워크 수준, 서비스 수준 또는 둘 다에서 적용할 수 있습니다. 일반적으로 네트워크 소유자 또는 클라우드 관리자가 인증 정책을 적용합니다. 조직 내에서 인증된 호출을 허용하거나 특정 조건과 일치하는 익명 GET 요청을 허용하는 등 대략적인 권한 부여를 구현할 수 있습니다. 서비스 수준에서 서비스 소유자는 보다 제한적일 수 있는 세밀한 제어를 적용할 수 있습니다. 자세한 내용은 인증 정책을 사용하여 VPC Lattice 서비스에 대한 액세스 제어 단원을 참조하십시오.

콘솔을 사용하여 액세스 정책을 추가 또는 업데이트하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스 네트워크를 선택합니다.
3. 서비스 네트워크의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 액세스 탭을 선택하여 현재 액세스 설정을 확인합니다.
5. 액세스 설정을 업데이트하려면 액세스 설정 편집을 선택합니다.
6. 연결된 VPC의 클라이언트가 이 서비스 네트워크의 서비스에 액세스하도록 하려면 인증 유형으로 없음을 선택합니다.
7. 서비스 네트워크에 리소스 정책을 적용하려면 인증 유형으로 AWS IAM을 선택하고 인증 정책에 대해 다음 중 하나를 수행합니다.
   - 입력 필드에 정책을 입력합니다. 복사하여 붙여넣을 수 있는 정책 예시를 보려면 정책 예시를 선택합니다.
   - 정책 템플릿 적용을 선택하고 인증 및 비인증 액세스 허용 템플릿을 선택합니다. 이 템플릿을 사용하면 다른 계정의 클라이언트가 요청에 서명해서(인증되었다는 의미) 또는 익명으로(인증되지 않았다는 의미) 서비스에 액세스할 수 있습니다.
   - 정책 템플릿 적용을 선택하고 인증된 액세스만 허용 템플릿을 선택합니다. 이 템플릿을 사용하면 다른 계정의 클라이언트가 요청에 서명해야만(인증되었다는 의미) 서비스에 액세스할 수 있습니다.
8. 변경 사항 저장을 선택합니다.

를 사용하여 액세스 정책을 추가하거나 업데이트하려면 AWS CLI

put-auth-policy 명령을 사용합니다.

VPC Lattice 서비스 네트워크의 모니터링 세부 정보 편집

VPC Lattice는 모든 요청과 응답에 대한 지표와 로그를 생성하므로 애플리케이션을 보다 효율적으로 모니터링하고 문제를 해결할 수 있습니다.

액세스 로그를 활성화하고 로그의 대상 리소스를 지정할 수 있습니다. VPC Lattice는 CloudWatch Log 그룹, Firehose 전송 스트림, S3 버킷 등의 리소스로 로그를 전송할 수 있습니다.

콘솔을 사용하여 액세스 로그를 활성화하거나 로그 대상을 업데이트하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스 네트워크를 선택합니다.
3. 서비스 네트워크의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 모니터링 탭을 선택합니다. 액세스 로그에서 액세스 로그의 활성화 여부를 확인합니다.
5. 액세스 로그를 활성화하거나 비활성화하려면 액세스 로그 편집을 선택한 다음 액세스 로그 토글 스위치를 켜거나 끕니다.
6. 액세스 로그를 활성화할 때는 전송 대상 유형을 선택한 다음 액세스 로그의 대상을 생성하거나 선택해야 합니다. 전송 대상은 언제든지 변경할 수도 있습니다. 예제:
   - CloudWatch 로그 그룹을 클릭하여 CloudWatch 로그 그룹을 선택합니다. 로그 그룹을 생성하려면 CloudWatch에서 로그 그룹 생성을 선택합니다.
   - S3 버킷을 선택하고 접두사를 포함한 S3 버킷 경로를 입력합니다. S3 버킷을 검색하려면 S3 찾아보기를 선택합니다.
   - Kinesis Data Firehose 전송 스트림을 클릭하고 전송 스트림을 선택합니다. 전송 스트림을 생성하려면 Kinesis에서 전송 스트림 생성을 선택합니다.
7. 변경 사항 저장을 선택합니다.

를 사용하여 액세스 로그를 활성화하려면 AWS CLI

create-access-log-subscription 명령을 사용합니다.

를 사용하여 로그 대상을 업데이트하려면 AWS CLI

update-access-log-subscription 명령을 사용합니다.

를 사용하여 액세스 로그를 비활성화하려면 AWS CLI

delete-access-log-subscription 명령을 사용합니다.

VPC Lattice 서비스 네트워크의 태그 관리

태그를 사용하면 용도, 소유자, 환경 등 다양한 방식으로 대상 그룹을 분류할 수 있습니다.

각 서비스 네트워크에 태그를 여러 개 추가할 수 있습니다. 태그 키는 각 서비스 네트워크마다 고유해야 합니다. 서비스 네트워크와 이미 연결된 키를 통해 태그를 추가하면 해당 태그의 값이 업데이트됩니다. 허용되는 문자는 글자, 공백, 숫자(UTF-8 형식) 및 특수 문자 + - = . _ : / @입니다. 선행 또는 후행 공백을 사용하면 안 됩니다. 태그 값은 대소문자를 구분합니다.

콘솔을 사용하여 태그를 추가 또는 삭제하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스 네트워크를 선택합니다.
3. 서비스 네트워크의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 태그 탭을 선택합니다.
5. 태그를 추가하려면 태그 추가를 선택한 다음 태그 키와 태그 값을 입력합니다. 다른 태그를 추가하려면 새 태그 추가를 선택합니다. 태그 추가가 완료되면 변경 사항 저장을 선택합니다.
6. 태그를 삭제하려면 태그의 확인란을 선택한 다음 삭제를 선택합니다. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 삭제를 선택합니다.

를 사용하여 태그를 추가하거나 삭제하려면 AWS CLI

tag-resource 및 untag-resource 명령을 사용합니다.

VPC Lattice 서비스 네트워크 삭제

서비스 네트워크를 삭제하려면 먼저 서비스 네트워크에서 서비스, 리소스 구성, VPC 또는 VPC 엔드포인트와 연결할 수 있는 모든 연결을 삭제해야 합니다. 서비스 네트워크를 삭제하면 리소스 정책, 인증 정책, 액세스 로그 구독 등 서비스 네트워크와 관련된 모든 리소스도 삭제됩니다.

콘솔을 사용하여 네트워크 인터페이스를 삭제하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스 네트워크를 선택합니다.
3. 서비스 네트워크에 대한 확인란을 선택한 후 작업, 서비스 네트워크 삭제를 선택합니다.

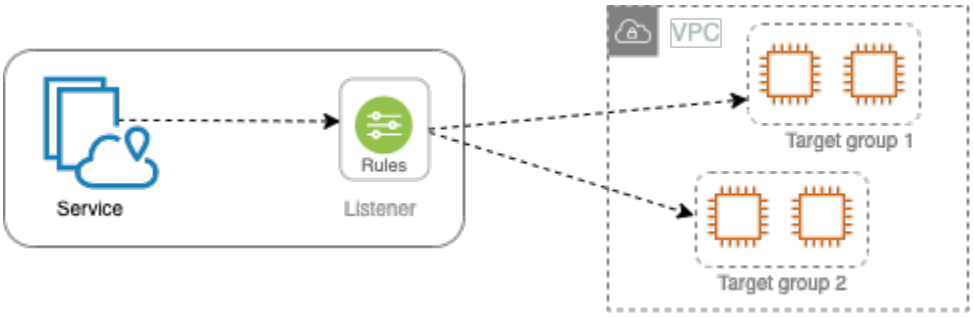
4. 확인 메시지가 나타나면 **confirm**을 입력한 다음 삭제를 선택합니다.

를 사용하여 서비스 네트워크를 삭제하려면 AWS CLI

delete-service-network 명령을 사용합니다.

VPC Lattice 내 서비스

VPC Lattice 내의 서비스는 독립적으로 배포할 수 있는 소프트웨어 단위로서 특정 작업이나 기능을 제공합니다. 서비스는 인스턴스 컨테이너나 계정 또는 Virtual Private Cloud(VPC) 내의 서버리스 함수에서 실행할 수 있습니다. 서비스에는 트래픽을 대상으로 라우팅하기 위해 구성할 수 있는 리스너가 있으며, 리스너 규칙이라는 규칙을 사용합니다. 지원되는 대상 유형에는 EC2 인스턴스, IP 주소, Lambda 함수, Application Load Balancer, Amazon ECS 작업 및 Kubernetes 포드가 포함됩니다. 자세한 내용은 VPC Lattice의 대상 그룹 단원을 참조하십시오. 서비스를 여러 서비스 네트워크와 연결할 수 있습니다. 다음 다이어그램은 VPC Lattice 내의 일반적인 서비스의 주요 구성 요소를 보여줍니다.



서비스에 이름과 설명을 지정하여 서비스를 생성할 수 있습니다. 하지만 서비스에 대한 트래픽을 제어하고 모니터링하려면 액세스 설정과 모니터링 세부 정보를 포함시키는 것이 중요합니다. 서비스에서 대상으로 트래픽을 보내려면 리스너를 설정하고 규칙을 구성해야 합니다. 서비스 네트워크에서 서비스로 트래픽이 흐를 수 있도록 하려면 서비스를 서비스 네트워크와 연결해야 합니다.

대상 연결에는 유휴 제한 시간과 전체 연결 제한 시간이 있습니다. 유휴 연결 제한 시간은 1분이며, 1분이 지나면 연결이 종료됩니다. 최대 지속 시간은 10분이며, 그 이후에는 연결을 통한 새 스트림이 허용되지 않으며 기존 스트림을 종료하는 프로세스가 시작됩니다.

작업

- 1단계: VPC Lattice 서비스 생성
- 2단계: 라우팅 정의
- 3단계: 네트워크 연결 생성
- 4단계: 검토 및 생성
- VPC Lattice 서비스의 연결 관리
- VPC Lattice 서비스에 대한 액세스 설정 편집
- VPC Lattice 서비스의 모니터링 세부 정보 편집
- VPC Lattice 서비스의 태그 관리

- VPC Lattice 서비스의 사용자 지정 도메인 이름 구성
- VPC Lattice용 자체 인증서 가져오기(BYOC)
- VPC Lattice 서비스 삭제

1단계: VPC Lattice 서비스 생성

액세스 설정과 모니터링 세부 정보가 포함된 기본 VPC Lattice 서비스를 생성합니다. 하지만 라우팅 구성을 정의하고 서비스 네트워크와 연결할 때까지는 서비스가 제대로 작동하지 않습니다.

콘솔을 사용하여 기본 서비스를 생성하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 생성을 선택합니다.
4. 식별자에서 다음을 수행합니다.

   a. 서비스의 이름을 입력합니다. 이름은 3~40자여야 하며 소문자, 숫자 및 하이픈을 사용해야 합니다. 문자나 숫자로 시작하고 끝나야 합니다. 이중 하이픈은 사용하지 마세요.

   b. (선택 사항) 서비스 네트워크 설명을 입력합니다. 생성 중 또는 생성 후 설명을 설정하거나 변경할 수 있습니다. 설명의 길이는 최대 256자입니다.
5. 서비스의 사용자 지정 도메인 이름을 지정하려면 사용자 지정 도메인 구성 지정을 선택하고 사용자 지정 도메인 이름을 입력합니다.

   HTTPS 리스너의 경우 VPC Lattice가 TLS 종료를 수행하는 데 사용할 인증서를 선택할 수 있습니다. 지금 인증서를 선택하지 않으면 서비스에 대한 HTTPS 리스너를 생성할 때 인증서를 선택할 수 있습니다.

   TCP 리스너의 경우 서비스에 대한 사용자 지정 도메인 이름을 지정해야 합니다. 인증서를 지정하면 사용되지 않습니다. 대신 애플리케이션에서 TLS 종료를 수행합니다.
6. 서비스 액세스에서, 서비스 네트워크와 연결된 VPC의 클라이언트가 서비스에 액세스하도록 하려면 없음을 선택합니다. 인증 정책을 적용하여 서비스에 대한 액세스를 제어하려면 AWS IAM을 선택합니다. 서비스에 리소스 정책을 적용하려면 인증 정책에 대해 다음 중 하나를 수행합니다.
   - 입력 필드에 정책을 입력합니다. 복사하여 붙여넣을 수 있는 정책 예시를 보려면 정책 예시를 선택합니다.

- 정책 템플릿 적용을 선택하고 인증 및 비인증 액세스 허용 템플릿을 선택합니다. 이 템플릿을 사용하면 다른 계정의 클라이언트가 요청에 서명해서(인증되었다는 의미) 또는 익명으로(인증되지 않았다는 의미) 서비스에 액세스할 수 있습니다.
- 정책 템플릿 적용을 선택하고 인증된 액세스만 허용 템플릿을 선택합니다. 이 템플릿을 사용하면 다른 계정의 클라이언트가 요청에 서명해야만(인증되었다는 의미) 서비스에 액세스할 수 있습니다.

7. (선택 사항) 액세스 로그를 활성화하려면 액세스 로그 토글 스위치를 켜고 다음과 같이 액세스 로그의 대상을 지정합니다.
   - CloudWatch 로그 그룹을 클릭하여 CloudWatch 로그 그룹을 선택합니다. 로그 그룹을 생성하려면 CloudWatch에서 로그 그룹 생성을 선택합니다.
   - S3 버킷을 선택하고 접두사를 포함한 S3 버킷 경로를 입력합니다. S3 버킷을 검색하려면 S3 찾아보기를 선택합니다.
   - Kinesis Data Firehose 전송 스트림을 클릭하고 전송 스트림을 선택합니다. 전송 스트림을 생성하려면 Kinesis에서 전송 스트림 생성을 선택합니다.
8. (선택 사항) 서비스를 다른 계정과 공유하려면 AWS RAM 리소스 공유에서 리소스 공유를 선택합니다. 리소스 공유를 생성하려면 RAM 콘솔에서 리소스 공유 생성을 선택합니다.
9. 구성을 검토하고 서비스를 생성하려면 검토 및 생성으로 건너뛰기를 선택합니다. 그렇지 않으면 다음을 선택하여 서비스의 라우팅 구성을 정의합니다.

2단계: 라우팅 정의

서비스에서 지정된 대상으로 트래픽을 전송할 수 있도록 리스너를 사용하여 라우팅 구성을 정의하세요.

사전 조건

리스너를 추가하려면 먼저 VPC Lattice 대상 그룹을 생성해야 합니다. 자세한 내용은 the section called “대상 그룹 생성” 단원을 참조하십시오.

콘솔을 사용하여 서비스의 라우팅을 정의하는 방법

1. 리스너 추가를 선택합니다.
2. 리스너 이름에서 사용자 지정 리스너 이름을 제공하거나 리스너의 프로토콜과 포트를 리스너 이름으로 사용할 수 있습니다. 지정할 사용자 지정 이름은 최대 63자까지 가능하며 계정의 모든 서비스마다 고유해야 합니다. 유효한 문자는 a~z, 0~9, 하이픈(-)입니다. 하이픈은 첫 문자 또는 마지

막 문자로 사용할 수도 없고 다른 하이픈 바로 뒤에 사용할 수도 없습니다. 리스너를 생성한 후에는 리스너의 이름을 변경할 수 없습니다.

3. 프로토콜을 선택한 다음 포트 번호를 입력합니다.
4. 기본 동작에서, 트래픽을 수신할 VPC Lattice 대상 그룹을 선택하고 이 대상 그룹에 할당할 가중치를 선택합니다. 필요한 경우 기본 작업에 다른 대상 그룹을 추가할 수 있습니다. 작업 추가를 선택한 다음 다른 대상 그룹을 선택하고 가중치를 지정합니다.
5. (선택 사항) 다른 규칙을 추가하려면 규칙 추가를 선택한 다음, 규칙의 이름, 우선 순위, 조건 및 작업을 입력합니다.

   각 규칙에 1에서 100 사이의 우선 순위 번호를 부여할 수 있습니다. 리스너는 우선 순위가 동일한 규칙을 여러 개 자질 수 없습니다. 규칙은 가장 낮은 값에서 가장 높은 값에 이르기까지 우선 순위에 따라 평가됩니다. 기본 규칙은 마지막에 평가됩니다.

   조건에서 경로 일치 조건의 경로 패턴을 입력합니다. 각 문자열의 최대 크기는 200자입니다. 이 비교는 대소문자를 구분하지 않습니다.
6. (선택 사항) 태그를 추가하려면 리스너 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.
7. 구성을 검토하고 서비스를 생성하려면 검토 및 생성으로 건너뛰기를 선택합니다. 그렇지 않으면 다음을 선택하여 서비스를 서비스 네트워크에 연결합니다.

3단계: 네트워크 연결 생성

클라이언트가 서비스 네트워크와 통신할 수 있도록 서비스를 서비스 네트워크와 연결합니다.

콘솔을 사용하여 서비스를 서비스 네트워크에 연결하는 방법

1. VPC Lattice 서비스 네트워크에서 서비스 네트워크를 선택합니다. 서비스 네트워크를 생성하려면 VPC Lattice 네트워크 생성을 선택합니다. 서비스를 여러 서비스 네트워크와 연결할 수 있습니다.
2. (선택 사항) 태그를 추가하려면 서비스 네트워크 연결 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.
3. 다음을 선택합니다.

4단계: 검토 및 생성

콘솔을 사용하여 구성을 검토하고 서비스를 생성하는 방법

1. 서비스 구성을 검토합니다.
2. 서비스 구성의 일부를 수정해야 하는 경우 편집을 선택합니다.
3. 구성 검토 또는 편집을 마쳤으면 VPC Lattice 서비스 생성을 선택합니다.
4. 서비스에 사용자 지정 도메인 이름을 지정한 경우 서비스가 생성된 후 DNS 라우팅을 구성해야 합니다. 자세한 내용은 the section called “사용자 지정 도메인 이름 구성” 단원을 참조하십시오.

VPC Lattice 서비스의 연결 관리

서비스를 서비스 네트워크와 연결하면 클라이언트(서비스 네트워크와 연결된 VPC의 리소스)가 이 서비스에 요청을 보낼 수 있습니다. 계정에 있는 서비스 또는 다른 계정에서 사용자와 공유하는 서비스를 연결할 수 있습니다. 서비스를 생성할 때 이 단계는 선택 사항입니다. 하지만 서비스를 생성한 후에는 서비스 네트워크와 연결할 때까지 다른 서비스와 통신할 수 없습니다. 계정에 필요한 액세스 권한이 있는 경우 서비스 소유자는 서비스를 서비스 네트워크에 연결할 수 있습니다. 자세한 내용은 VPC Lattic의 작동 방식 단원을 참조하십시오.

콘솔을 사용하여 서비스 네트워크 연결을 관리하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 서비스 네트워크 연결 탭을 선택합니다.
5. 연결을 생성하려면 다음을 수행합니다.
   a. 연결 생성을 선택합니다.
   b. VPC Lattice 서비스 네트워크에서 서비스 네트워크를 선택합니다. 서비스 네트워크를 생성하려면 VPC Lattice 네트워크 생성을 선택합니다.
   c. (선택 사항) 태그를 추가하려면 서비스 연결 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.
   d. 변경 사항 저장을 선택합니다.
6. 연결을 삭제하려면 연결의 확인란을 선택한 다음 작업, 네트워크 연결 삭제를 선택합니다. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 삭제를 선택합니다.

를 사용하여 서비스 네트워크 연결을 생성하려면 AWS CLI

create-service-network-service-association 명령을 사용합니다.

를 사용하여 서비스 네트워크 연결을 삭제하려면 AWS CLI

delete-service-network-service-association 명령을 사용합니다.

VPC Lattice 서비스에 대한 액세스 설정 편집

액세스 설정을 통해 서비스에 대한 클라이언트 액세스를 구성하고 관리할 수 있습니다. 액세스 설정에는 인증 유형과 인증 정책이 포함됩니다. 인증 정책은 VPC Lattice 내의 서비스로 흐르는 트래픽을 인증하고 승인하는 데 도움이 됩니다.

인증 정책은 서비스 네트워크 수준, 서비스 수준 또는 둘 다에서 적용할 수 있습니다. 서비스 수준에서 서비스 소유자는 보다 제한적일 수 있는 세밀한 제어를 적용할 수 있습니다. 일반적으로 네트워크 소유자 또는 클라우드 관리자가 인증 정책을 적용합니다. 예를 들어 조직 내에서 인증된 통화를 허용하거나 특정 조건에 맞는 익명 GET 요청을 허용하는 등, 세분화되지 않은 인증을 구현할 수 있습니다. 자세한 내용은 인증 정책을 사용하여 VPC Lattice 서비스에 대한 액세스 제어 단원을 참조하십시오.

콘솔을 사용하여 액세스 정책을 추가 또는 업데이트하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 액세스 탭을 선택하여 현재 액세스 설정을 확인합니다.
5. 액세스 설정을 업데이트하려면 액세스 설정 편집을 선택합니다.
6. 연결된 서비스 네트워크의 VPC에 있는 클라이언트가 서비스에 액세스하도록 하려면 인증 유형으로 없음을 선택합니다.
7. 리소스 정책을 적용하여 서비스에 대한 액세스를 제어하려면 인증 유형으로 AWS IAM을 선택하고 인증 정책에 대해 다음 중 하나를 수행합니다.
   - 입력 필드에 정책을 입력합니다. 복사하여 붙여넣을 수 있는 정책 예시를 보려면 정책 예시를 선택합니다.
   - 정책 템플릿 적용을 선택하고 인증 및 비인증 액세스 허용 템플릿을 선택합니다. 이 템플릿을 사용하면 다른 계정의 클라이언트가 요청에 서명해서(인증되었다는 의미) 또는 익명으로(인증되지 않았다는 의미) 서비스에 액세스할 수 있습니다.

- 정책 템플릿 적용을 선택하고 인증된 액세스만 허용 템플릿을 선택합니다. 이 템플릿을 사용하면 다른 계정의 클라이언트가 요청에 서명해야만(인증되었다는 의미) 서비스에 액세스할 수 있습니다.

8. 변경 사항 저장을 선택합니다.

를 사용하여 액세스 정책을 추가하거나 업데이트하려면 AWS CLI

put-auth-policy 명령을 사용합니다.

VPC Lattice 서비스의 모니터링 세부 정보 편집

VPC Lattice는 모든 요청과 응답에 대한 지표와 로그를 생성하므로 애플리케이션을 보다 효율적으로 모니터링하고 문제를 해결할 수 있습니다.

액세스 로그를 활성화하고 로그의 대상 리소스를 지정할 수 있습니다. VPC Lattice는 CloudWatch Log 그룹, Firehose 전송 스트림, S3 버킷 등의 리소스로 로그를 전송할 수 있습니다.

콘솔을 사용하여 액세스 로그를 활성화하거나 로그 대상을 업데이트하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 모니터링 탭을 선택한 다음 로그를 선택합니다. 액세스 로그에서 액세스 로그의 활성화 여부를 확인합니다.
5. 액세스 로그를 활성화하거나 비활성화하려면 액세스 로그 편집을 선택한 다음 액세스 로그 토글 스위치를 켜거나 끕니다.
6. 액세스 로그를 활성화할 때는 전송 대상 유형을 선택한 다음 액세스 로그의 대상을 생성하거나 선택해야 합니다. 전송 대상은 언제든지 변경할 수도 있습니다. 예제:
   - CloudWatch 로그 그룹을 클릭하여 CloudWatch 로그 그룹을 선택합니다. 로그 그룹을 생성하려면 CloudWatch에서 로그 그룹 생성을 선택합니다.
   - S3 버킷을 선택하고 접두사를 포함한 S3 버킷 경로를 입력합니다. S3 버킷을 검색하려면 S3 찾아보기를 선택합니다.
   - Kinesis Data Firehose 전송 스트림을 클릭하고 전송 스트림을 선택합니다. 전송 스트림을 생성하려면 Kinesis에서 전송 스트림 생성을 선택합니다.
7. 변경 사항 저장을 선택합니다.

를 사용하여 액세스 로그를 활성화하려면 AWS CLI

create-access-log-subscription 명령을 사용합니다.

를 사용하여 로그 대상을 업데이트하려면 AWS CLI

update-access-log-subscription 명령을 사용합니다.

를 사용하여 액세스 로그를 비활성화하려면 AWS CLI

delete-access-log-subscription 명령을 사용합니다.

VPC Lattice 서비스의 태그 관리

태그를 사용하면 용도, 소유자 또는 환경 등 다양한 방식으로 서비스를 분류할 수 있습니다.

각 서비스에 여러 태그를 추가할 수 있습니다. 태그 키는 각 서비스에 대해 고유해야 합니다. 서비스와 이미 연결된 키가 있는 태그를 추가하면 해당 태그의 값이 업데이트됩니다. 허용되는 문자는 글자, 공백, 숫자(UTF-8 형식) 및 특수 문자 + - = . _ : / @입니다. 선행 또는 후행 공백을 사용하면 안 됩니다. 태그 값은 대소문자를 구분합니다.

콘솔을 사용하여 태그를 추가 또는 삭제하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 태그 탭을 선택합니다.
5. 태그를 추가하려면 태그 추가를 선택한 다음 태그 키와 태그 값을 입력합니다. 다른 태그를 추가하려면 새 태그 추가를 선택합니다. 태그 추가가 완료되면 변경 사항 저장을 선택합니다.
6. 태그를 삭제하려면 태그의 확인란을 선택한 다음 삭제를 선택합니다. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 삭제를 선택합니다.

를 사용하여 태그를 추가하거나 삭제하려면 AWS CLI

tag-resource 및 untag-resource 명령을 사용합니다.

VPC Lattice 서비스의 사용자 지정 도메인 이름 구성

새 서비스를 생성하면 VPC Lattice는 다음 구문을 사용하여 서비스에 대한 고유한 FQDN(정규화된 도메인 이름)을 생성합니다.

```
service_name-service_id.partition_id.vpc-lattice-svcs.region.on.aws
```

그러나 VPC Lattice가 제공하는 도메인 이름은 사용자가 기억하기 쉽지 않습니다. 사용자 지정 도메인 이름은 사용자에게 제공할 수 있는 더 간단하고 직관적인 URL입니다. 서비스에 대해 VPC Lattice 생성 DNS 이름 대신 사용자 지정 도메인 이름(예: `www.parking.example.com`)을 사용하려면 VPC Lattice 서비스를 생성할 때 구성할 수 있습니다. 클라이언트가 이러한 사용자 지정 도메인 이름을 사용해 요청을 하면 DNS 서버는 이를 VPC Lattice 새성 도메인 이름으로 해석합니다.

필수 조건

- 서비스에 등록된 도메인 이름이 있어야 합니다. 등록된 도메인 이름이 아직 없는 경우 Amazon Route 53 또는 다른 등록 대행 기관을 통해 등록할 수 있습니다.
- HTTPS 요청을 받으려면 AWS Certificate Manager에서 자체 인증서를 제공해야 합니다. VPC Lattice는 대체 인증서로 기본 인증서를 지원하지 않습니다. 따라서 사용자 지정 도메인 이름에 해당하는 SSL/TLS 인증서를 제공하지 않으면 사용자 지정 도메인 이름에 대한 모든 HTTPS 연결이 실패합니다. 자세한 내용은 VPC Lattice용 자체 인증서 가져오기(BYOC) 단원을 참조하십시오.

제한 사항 및 고려 사항

- 서비스에는 사용자 지정 도메인 이름을 두 개 이상 사용할 수 없습니다.
- 서비스를 생성한 후에는 사용자 지정 도메인 이름을 수정할 수 없습니다.
- 사용자 지정 도메인 이름은 서비스 네트워크별로 고유해야 합니다. 즉, 동일한 서비스 네트워크에 (다른 서비스용으로) 이미 존재하는 사용자 지정 도메인 이름으로는 서비스를 생성할 수 없습니다.

다음 절차에서는 서비스에 대한 사용자 지정 도메인 이름을 구성하는 방법을 보여줍니다.

AWS Management Console

서비스에 대한 사용자 지정 도메인 이름을 구성하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.

3. 서비스 생성을 선택합니다. 1단계: 서비스 생성으로 이동합니다.
4. 사용자 지정 도메인 구성 섹션에서 사용자 지정 도메인 구성 지정을 선택합니다.
5. 사용자 지정 도메인 이름을 입력합니다.
6. HTTPS 요청을 처리하려면 사용자 지정 SSL/TLS 인증서에서 사용자 지정 도메인 이름과 일치하는 SSL/TLS 인증서를 선택합니다. 아직 인증서가 없거나 인증서를 지금 추가하고 싶지 않은 경우에는 HTTPS 리스너를 생성할 때 인증서를 추가할 수 있습니다. 하지만 인증서가 없으면 사용자 지정 도메인 이름이 HTTPS 요청을 처리할 수 없습니다. 자세한 내용은 HTTPS 리스너 추가 단원을 참조하십시오.
7. 서비스 생성을 위한 다른 모든 정보를 추가했으면 생성을 선택합니다.

AWS CLI

서비스에 대한 사용자 지정 도메인 이름을 구성하려면

create-service 명령을 사용합니다.

```
aws vpc-lattice create-service --name service_name --custom-domain-
name your_custom_domain_name --type https --certificate-arn arn:aws:acm:us-
east-1:123456789012:certificate/12345678-1234-1234-1234-123456789012
```

위 명령의 `--name`에서 서비스 이름을 입력합니다. `--custom-domain-name`에서 서비스의 도메인 이름(예: `parking.example.com`)을 입력합니다. `--certificate-arn`에서 ACM에 있는 인증서의 ARN을 입력합니다. 인증서 ARN은 AWS Certificate Manager의 계정에서 사용할 수 있습니다.

사용자 지정 도메인 이름을 서비스와 연결

먼저, 아직 등록을 하지 않았다면 사용자 지정 도메인 이름부터 등록합니다. 국제인터넷주소관리기구(ICANN)가 인터넷 도메인 이름을 관리합니다. 도메인 이름 등록을 관리하는 ICANN 인증 조직인 도메인 이름 등록 대행자를 이용해 도메인 이름을 등록합니다. 등록 대행자 웹 사이트는 도메인 이름 등록에 대한 자세한 지침과 요금 정보를 제공합니다. 자세한 내용은 다음 자료를 참조하세요.

- Amazon Route 53을 사용하여 도메인 이름을 등록하려면 Amazon Route 53 개발자 안내서의 Route 53을 사용하여 도메인 이름 등록을 참조하세요.
- 인증된 등록 대행자의 목록은 Accredited Registrar Directory를 참조하세요.

그런 다음 도메인 등록 기관과 같은 DNS 서비스를 사용하여 쿼리를 서비스로 라우팅하는 레코드를 생성합니다. 자세한 내용은 DNS 서비스에 대한 설명서를 참조하세요. 또는 DNS 서비스로 Route 53을 사용할 수도 있습니다.

Route 53을 사용하는 경우 별칭 레코드 또는 CNAME 레코드를 사용하여 쿼리를 서비스로 라우팅할 수 있습니다. 영역 정점이라고도 하는 DNS 네임스페이스의 상단 노드에서 별칭 레코드를 생성할 수 있으므로 별칭 레코드를 사용하는 것이 좋습니다.

Route 53를 사용하는 경우, 먼저 도메인을 위해 인터넷에서 트래픽을 라우팅하는 방법이 포함된 호스팅 영역을 생성해야 합니다. 프라이빗 또는 퍼블릭 호스팅 영역을 생성한 후와 같은 사용자 지정 도메인 이름이와 같은 VPC Lattice 자동 생성 도메인 이름에 `parking.example.com`매핑되도록 레코드를 생성합니다`my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws`. 이 매핑이 없으면 사용자 지정 도메인 이름이 VPC Lattice에서 작동하지 않습니다.

다음 절차에서는 Route 53을 사용하여 프라이빗 또는 퍼블릭 호스팅 영역을 생성하는 방법을 보여줍니다.

AWS Management Console

Route 53을 사용하여 서비스로 쿼리를 라우팅하는 별칭 레코드를 생성하려면 Amazon VPC Lattice 서비스 도메인 엔드포인트로 트래픽 라우팅을 참조하세요.

값에 대해와 같이 서비스에 `my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws` 대해 VPC Lattice에서 생성한 도메인 이름을 사용합니다. 이 자동 생성된 도메인 이름은 서비스 페이지의 VPC Lattice 콘솔에서 찾을 수 있습니다.

AWS CLI

호스팅 영역에서 별칭 레코드를 생성하려면

1. 서비스에 대해 VPC Lattice에서 생성한 도메인 이름을 가져옵니다(예: `my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws`).
2. 별칭을 설정하려면 다음 명령을 사용하세요.

```
aws route53 change-resource-record-sets --hosted-zone-id your-hosted-zone-ID --
change-batch file://~/Desktop/change-set.json
```

`change-set.json` 파일의 경우 다음 JSON 예시의 콘텐츠가 포함된 JSON 파일을 생성하고 로컬 시스템에 저장합니다. 위 명령의 *`file://~/Desktop/change-set.json`*을 로컬 컴퓨터에 저장된 JSON 파일의 경로로 바꿉니다. 참고로, 다음 JSON의 "유형"은 A 또는 AAAA 레코드 유형일 수 있습니다.

```
{
    "Comment": "my-custom-domain-name.com alias",
    "Changes": [
        {
            "Action": "CREATE",
            "ResourceRecordSet": {
                "Name": "my-custom-domain-name.com",
                "Type": "alias-record-type",
                "AliasTarget": {
                    "HostedZoneId": "your-hosted-zone-ID",
                    "DNSName": "lattice-generated-domain-name",
                    "EvaluateTargetHealth": true
                }
            }
        }
    ]
}
```

VPC Lattice용 자체 인증서 가져오기(BYOC)

HTTPS 요청을 처리하려면 사용자 지정 도메인 이름을 설정하기 전에 AWS Certificate Manager (ACM)에서 SSL/TLS 인증서를 준비해야 합니다. 이 인증서에는 서비스의 사용자 지정 도메인 이름과 일치하는 SAN(주체 대체 이름) 또는 CN(일반 이름)이 있어야 합니다. SAN이 있는 경우 SAN 목록에서만 일치 항목이 있는지 확인합니다. SAN이 없는 경우 CN에서 일치 항목이 있는지 확인합니다.

VPC Lattice는 SNI(서버 이름 표시)를 사용하여 HTTPS 요청을 처리합니다. DNS는 사용자 지정 도메인 이름 및 이 도메인 이름과 일치하는 인증서를 기반으로 HTTPS 요청을 VPC Lattice 서비스로 라우팅합니다. 도메인 이름에 대한 SSL/TLS 인증서를 ACM에서 요청하거나 ACM으로 가져오려면 AWS Certificate Manager 사용 설명서의 인증서 발급 및 관리 및 인증서 가져오기를 참조하세요. ACM에서 자체 인증서를 요청하거나 가져올 수 없는 경우 VPC Lattice에서 생성한 도메인 이름과 인증서를 사용하세요.

VPC Lattice는 서비스당 하나의 사용자 지정 인증서만 허용합니다. 그러나 사용자 지정 인증서를 사용자 지정 도메인 여러 개에 사용할 수 있습니다. 즉, 사용자 지정 도메인 이름으로 생성한 모든 VPC Lattice 서비스에 동일한 인증서를 사용할 수 있습니다.

ACM 콘솔을 사용하여 인증서를 보려면 인증서를 열고 인증서 ID를 선택합니다. 관련 리소스에서 해당 인증서와 연결된 VPC Lattice 서비스를 확인해야 합니다.

제한 사항 및 고려 사항

- VPC Lattice는 연결된 인증서의 SAN(주체 대체 이름) 또는 CN(일반 이름)에서 단일 수준 깊이의 와일드카드 일치를 허용합니다. 사용자 지정 도메인 이름 `parking.example.com`으로 서비스를 생성하고 고유 인증서를 SAN `*.example.com`과 연결하는 경우를 예로 들 수 있습니다. `parking.example.com`에 대한 요청이 들어오면 VPC Lattice는 SAN을 apex 도메인 `example.com`이 있는 모든 도메인 이름과 일치시킵니다. 하지만 사용자 지정 도메인 `parking.different.example.com`이 있고 인증서에 SAN `*.example.com`이 있는 경우에는 요청이 실패합니다.
- VPC Lattice는 단일 수준의 와일드카드 도메인 일치를 지원합니다. 즉, 와일드카드는 첫 번째 수준의 하위 도메인으로만 사용할 수 있으며 하나의 하위 도메인 수준만 보호합니다. 예를 들어 인증서의 SAN이 `*.example.com`인 경우 `parking.*.example.com`은 지원되지 않습니다.
- VPC Lattice는 도메인 이름당 하나의 와일드카드를 지원합니다. 즉, `*.*.example.com`은 유효하지 않습니다. 자세한 내용은 AWS Certificate Manager 사용 설명서에서 공인 인증서 요청을 참조하세요.
- VPC Lattice는 2048비트 RSA 키가 있는 인증서만 지원합니다.
- ACM의 SSL/TLS 인증서는 연결하려는 VPC Lattice 서비스와 동일한 리전에 있어야 합니다.

인증서의 프라이빗 키 보호

ACM을 사용하여 SSL/TLS 인증서를 요청하면 ACM은 퍼블릭/프라이빗 키 페어를 생성합니다. 인증서를 가져오면 키 페어를 생성합니다. 퍼블릭 키는 인증서의 일부가 됩니다. 프라이빗 키를 안전하게 저장하기 위해 ACM은 별칭 aws/acm을 사용하여 KMS 키 AWS KMS라는 다른 키를 생성합니다.는 이 키를 AWS KMS 사용하여 인증서의 프라이빗 키를 암호화합니다. 자세한 내용은 AWS Certificate Manager 사용 설명서에서 AWS Certificate Manager의 데이터 보호를 참조하세요.

VPC Lattice는 에서만 액세스할 수 있는 서비스인 AWS TLS 연결 관리자 AWS 서비스를 사용하여 인증서의 프라이빗 키를 보호하고 사용합니다. ACM 인증서를 사용하여 VPC Lattice 서비스를 생성하면 VPC Lattice는 인증서를 AWS TLS Connection Manager와 연결합니다. 관리형 AWS 키에 AWS KMS 대해에서 권한 부여를 생성하여이 작업을 수행합니다. 이 권한 부여를 통해 TLS Connection Manager는를 사용하여 인증서의 프라이빗 키를 해독 AWS KMS 할 수 있습니다. TLS 연결 관리자는 인증서와 해독된(일반 텍스트) 프라이빗 키를 사용하여 VPC Lattice 서비스의 클라이언트와 보안 연결(SSL/TLS 세션)을 설정합니다. 인증서가 통합 서비스에서 연결 해제되면 권한 부여가 사용 중지됩니다. 자세한 내용은 AWS Key Management Service 개발자 안내서의 권한 부여를 참조하세요.

자세한 내용은 저장 중 암호화 단원을 참조하십시오.

VPC Lattice 서비스 삭제

VPC Lattice 서비스를 삭제하려면 먼저 서비스 네트워크와 서비스의 모든 연결을 삭제해야 합니다. 서비스를 삭제하면 리소스 정책, 인증 정책, 리스너, 리스너 규칙, 액세스 로그 구독 등 서비스와 관련된 모든 리소스도 삭제됩니다.

콘솔을 사용하여 프리셋을 삭제하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 페이지에서 삭제하려는 서비스를 선택한 다음 작업, 서비스 삭제를 선택합니다.
4. 확인 메시지가 나타나면 삭제를 선택합니다.

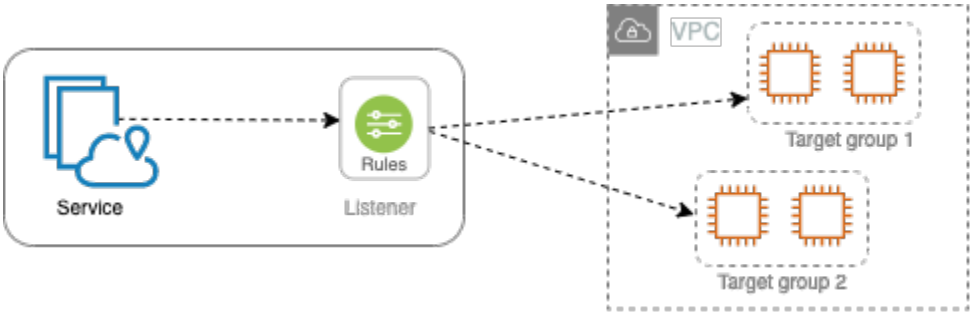
를 사용하여 서비스를 삭제하려면 AWS CLI

delete-service 명령을 사용합니다.

VPC Lattice의 대상 그룹

VPC Lattice 대상 그룹은 애플리케이션 또는 서비스를 실행하는 대상 또는 컴퓨팅 리소스의 모음입니다. 지원되는 대상 유형에는 EC2 인스턴스, IP 주소, Lambda 함수, Application Load Balancer, Amazon ECS 작업 및 Kubernetes 포드가 포함됩니다. 기존 서비스를 대상 그룹에 연결할 수도 있습니다. VPC Lattice에서 Kubernetes를 사용하는 방법에 대한 자세한 내용은 AWS 게이트웨이 API 컨트롤러 사용 설명서를 참조하세요.

각 대상 그룹은 하나 이상의 등록된 대상에 요청을 라우팅하는 데 사용됩니다. 리스너 규칙을 생성할 때 대상 그룹 및 조건을 지정합니다. 규칙 조건이 충족되면 해당하는 대상 그룹으로 트래픽이 전달됩니다. 서로 다른 유형의 요청에 대해 서로 다른 대상 그룹을 생성할 수 있습니다. 예를 들어, 일반 요청을 위한 하나의 대상 그룹을 생성하고 경로나 헤더 값과 같은 특정 규칙 조건을 포함하는 요청에 대해 다른 대상 그룹을 생성합니다.



대상 그룹 기준으로 서비스에 대한 상태 확인 설정을 정의합니다. 대상 그룹을 만들거나 나중에 변경할 때 재정의하지 않는 이상 각 대상 그룹은 기본 상태 확인 설정을 사용합니다. 리스너에 대한 규칙에 대상 그룹을 지정한 후, 서비스는 대상 그룹에 등록된 모든 대상의 상태를 지속적으로 모니터링합니다. 서비스는 정상 상태로 등록된 대상으로 요청을 라우팅합니다.

서비스 리스너에 대한 규칙에서 대상 그룹을 지정하려면 대상 그룹이 서비스와 동일한 계정에 있어야 합니다.

VPC Lattice 대상 그룹은 Elastic Load Balancing에서 제공하는 대상 그룹과 비슷하지만 서로 바꿔서 사용할 수는 없습니다.

내용

- VPC Lattice 대상 그룹 생성
- VPC Lattice 대상 그룹에 대상 등록
- VPC Lattice 대상 그룹에 대한 상태 확인
- 라우팅 구성

- 라우팅 알고리즘
- 대상 유형
- IP 주소 유형
- VPC Lattice의 HTTP 대상
- VPC Lattice에서 대상으로서 Lambda 함수
- VPC Lattice의 대상인 Application Load Balancer
- 프로토콜 버전
- VPC Lattice 대상 그룹에 대한 태그
- VPC Lattice 대상 그룹 삭제

VPC Lattice 대상 그룹 생성

대상 그룹에 대상을 등록합니다. 기본적으로 VPC Lattice 서비스는 대상 그룹에 대해 지정한 프로토콜과 포트 번호를 사용하여 등록된 대상으로 요청을 전송합니다. 또는 대상 그룹에 각 대상을 등록할 때 이 포트를 재정의할 수 있습니다.

대상 그룹의 대상으로 트래픽을 라우팅하려면 리스너 또는 리스너에 대한 규칙을 생성할 때 작업에 대상 그룹을 지정합니다. 자세한 내용은 VPC Lattice 서비스를 위한 리스너 규칙 단원을 참조하십시오. 여러 리스너에서 동일한 대상 그룹을 지정할 수 있지만, 이러한 리스너는 동일한 서비스에 속해야 합니다. 대상 그룹을 서비스와 함께 사용하려면 대상 그룹이 다른 서비스용으로 리스너에서 사용되고 있지 않은지 확인해야 합니다.

언제든지 대상 그룹에서 대상을 추가하거나 삭제할 수 있습니다. 자세한 내용은 VPC Lattice 대상 그룹에 대상 등록 단원을 참조하십시오. 대상 그룹에 대한 상태 확인 설정을 변경할 수도 있습니다. 자세한 내용은 VPC Lattice 대상 그룹에 대한 상태 확인 단원을 참조하십시오.

대상 그룹 생성

다음과 같이 대상 그룹을 생성하고 필요할 경우 대상을 등록할 수 있습니다.

콘솔을 사용하여 대상 그룹을 생성하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 대상 그룹을 선택합니다.
3. 대상 그룹 생성을 선택합니다.
4. 대상 유형 선택으로 다음 중 하나를 선택합니다.

- 인스턴스 ID로 대상을 등록하려면 인스턴스를 선택합니다.
- IP 주소로 대상을 등록하려면 IP 주소를 선택합니다.
- Lambda 함수를 대상으로 등록하려면 Lambda 함수를 선택합니다.
- Application Load Balancer를 대상으로 등록하려면 Application Load Balancer를 선택합니다.

5. 대상 그룹 이름에 대상 그룹의 이름을 입력합니다. 이 이름은 각 AWS 리전의 계정에 대해 고유해야 하며, 최대 32자여야 하고, 영숫자 또는 하이픈만 포함해야 하며, 하이픈으로 시작하거나 끝나지 않아야 합니다.
6. (선택 사항) 프로토콜과 포트에서 필요에 따라 기본값을 변경합니다. 기본 프로토콜은 HTTPS이고 기본 포트는 443입니다.

   대상 유형이 Lambda 함수인 경우 프로토콜이나 포트를 지정할 수 없습니다.
7. IP 주소 유형에서 IPv4를 선택하여 대상을 IPv4 주소로 등록하거나 IPv6를 선택하여 대상을 IPv6 주소로 등록합니다. 대상 그룹을 생성한 후에는 이 설정을 변경할 수 없습니다.

   이 옵션은 대상 유형이 IP 주소인 경우에 사용할 수 있습니다.
8. VPC에서 Virtual Private Cloud(VPC)를 선택합니다.

   대상 유형이 Lambda 함수인 경우 이 옵션을 사용할 수 없습니다.
9. (선택 사항) 프로토콜 버전에서 필요에 따라 기본값을 수정합니다. 기본값은 HTTP1입니다.

   대상 유형이 Lambda 함수인 경우 이 옵션을 사용할 수 없습니다.
10. 상태 확인에서 필요에 따라 기본 설정을 수정합니다. 자세한 내용은 VPC Lattice 대상 그룹에 대한 상태 확인 단원을 참조하십시오.

    대상 유형이 Lambda 함수인 경우 상태 확인을 사용할 수 없습니다.
11. Lambda 이벤트 구조 버전에서 버전을 선택합니다. 자세한 내용은 the section called “VPC Lattice 서비스에서 이벤트 수신” 단원을 참조하십시오.

    이 옵션은 대상 유형이 Lambda 함수인 경우에만 사용할 수 있습니다.
12. (선택 사항) 태그를 추가하려면 태그를 확장하고 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
13. 다음을 선택합니다.
14. 대상 등록에서 이 단계를 건너뛰거나 다음과 같이 대상을 추가할 수 있습니다.
    - 대상 유형이 인스턴스인 경우 인스턴스를 선택하고 포트를 입력한 다음 아래에 보류 중인 것으로 포함을 선택합니다.

- 대상 유형이 IP 주소인 경우 다음을 수행합니다.

  a. 네트워크 선택에서 대상 그룹에 대해 선택한 VPC를 그대로 유지하거나 기타 프라이빗 IP 주소를 선택합니다.

  b. IP 지정 및 포트 정의에서 IP 주소를 입력하고 포트를 입력합니다. 기본 포트는 대상 그룹 포트입니다.

  c. 아래에서 보류 중인 것으로 포함을 선택하세요.

- 대상 유형이 Lambda 함수인 경우 Lambda 함수를 선택합니다. Lambda 함수를 생성하려면 새 Lambda 함수 생성을 선택합니다.
- 대상 유형이 Application Load Balancer인 경우 Application Load Balancer를 선택합니다. Application Load Balancer를 생성하려면 Application Load Balancer 생성을 선택합니다.

15. 대상 그룹 생성을 선택합니다.

    VPC Lattice가 대상을 등록하는 데 몇 분 정도 걸릴 수 있습니다. 자세한 내용은 Route 53 및 퍼블릭 해석기에서 DNS 변경 사항이 전파되는 데 시간이 오래 걸리는 이유는 무엇인가요?를 참조하세요.

를 사용하여 대상 그룹을 생성하려면 AWS CLI

create-target-group 명령을 사용하여 대상 그룹을 생성하고 register-targets 명령을 사용하여 대상을 추가합니다.

공유 서브넷

참여자는 공유 VPC에서 VPC Lattice 대상 그룹을 생성할 수 있습니다. 공유 서브넷에는 다음 규칙이 적용됩니다.

- 리스너, 대상 그룹, 대상 등 VPC Lattice 서비스의 모든 부분을 동일한 계정에서 생성해야 합니다. VPC Lattice 서비스 소유자가 소유하거나 소유자와 공유하는 서브넷에서 생성할 수 있습니다.
- 대상 그룹에 등록된 대상은 대상 그룹과 동일한 계정에서 생성해야 합니다.
- VPC 소유자만 VPC를 서비스 네트워크와 연결할 수 있습니다. 서비스 네트워크와 연결된 공유 VPC의 참여자 리소스는 서비스 네트워크와 연결된 서비스에 요청을 보낼 수 있습니다. 하지만 관리자는 보안 그룹, 네트워크 ACL 또는 인증 정책을 사용하여 이를 방지할 수 있습니다.

VPC Lattice의 공유 가능한 리소스에 대한 자세한 내용은 VPC Lattice 엔터티 공유 섹션을 참조하세요.

VPC Lattice 대상 그룹에 대상 등록

서비스는 클라이언트에 대해 단일 접점의 역할을 하며 정상적으로 등록된 대상 간에 수신 트래픽을 자동으로 분산합니다. 하나 이상의 대상 그룹에 각 대상을 등록할 수 있습니다.

애플리케이션에 대한 요구가 증가하면 이를 처리하기 위해 하나 이상의 대상 그룹에 추가 대상을 등록할 수 있습니다. 서비스는 등록 프로세스가 완료되고 대상이 초기 상태 확인을 통과하자마자 새로 등록된 대상에 대한 라우팅 요청을 시작합니다.

애플리케이션에 대한 요구가 감소하거나 대상을 서비스해야 하는 경우에는 대상 그룹에서 대상 등록을 취소할 수 있습니다. 대상을 등록 취소하면 대상 그룹에서 제거되지만 대상에 영향을 미치지는 않습니다. 등록이 취소되는 즉시 서비스는 대상으로의 요청 라우팅을 중지합니다. 진행 중인 요청이 완료될 때까지 해당 대상은 `DRAINING` 상태를 유지합니다. 요청 수신을 다시 시작할 준비가 되면 대상 그룹에 대상을 다시 등록할 수 있습니다.

대상 그룹의 대상 유형에 따라 해당 대상 그룹에 대상을 등록하는 방법이 결정됩니다. 자세한 내용은 대상 유형 단원을 참조하십시오.

다음 콘솔 절차를 사용하여 대상을 등록하거나 등록 취소합니다. 또한 AWS CLI의 register-targets 및 deregister-targets 명령을 사용할 수 있습니다.

내용

- 인스턴스 ID로 대상 등록 또는 등록 취소
- IP 주소로 대상 등록 또는 등록 취소
- Lambda 함수 등록 또는 등록 취소
- Application Load Balancer 등록 또는 등록 취소

인스턴스 ID로 대상 등록 또는 등록 취소

대상 인스턴스는 대상 그룹에 대해 지정한 Virtual Private Cloud(VPC)에 있어야 합니다. 또한 인스턴스를 등록할 때 인스턴스가 `running` 상태여야 합니다.

인스턴스 ID로 대상을 등록하는 경우 오토 스케일링에 서비스를 사용할 수 있습니다. 오토 스케일링에 대상 그룹을 연결하고 해당 그룹이 확장되면, 오토 스케일링에서 시작한 인스턴스가 대상 그룹에 자동으로 등록됩니다. 오토 스케일링에서 대상 그룹을 분리하면 인스턴스가 대상 그룹에서 자동으로 등록 취소됩니다. 자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서에서 VPC Lattice 대상 그룹과 함께 오토 스케일링으로 트래픽 라우팅을 참조하세요.

콘솔을 사용하여 인스턴스 ID별로 대상을 등록 또는 등록 취소하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 대상 그룹을 선택합니다.
3. 대상 그룹의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 대상 탭을 선택합니다.
5. 인스턴스를 등록하려면 대상 등록을 선택합니다. 인스턴스를 선택하고 포트를 입력한 다음 아래에 보류 중인 것으로 포함을 선택합니다. 인스턴스 추가를 마쳤으면 대상 등록을 선택합니다.
6. 인스턴스의 등록을 취소하려면 인스턴스를 선택한 다음 등록 취소를 선택합니다.

IP 주소로 대상 등록 또는 등록 취소

대상 IP 주소는 대상 그룹에 대해 지정한 VPC 서브넷에서 가져와야 합니다. 동일한 VPC에 다른 서비스의 IP 주소는 등록할 수 없습니다. VPC 엔드포인트 또는 공개적으로 라우팅 가능한 IP 주소는 등록할 수 없습니다.

콘솔을 사용하여 IP 주소로 대상을 등록 또는 등록 취소하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 대상 그룹을 선택합니다.
3. 대상 그룹의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 대상 탭을 선택합니다.
5. IP 주소를 등록하려면 대상 등록을 선택합니다. 각 IP 주소에 대해 네트워크를 선택하고 IP 주소 및 포트를 입력한 다음 아래에 보류 중인 것으로 포함을 선택합니다. 주소 지정을 마치면 대상 등록을 선택합니다.
6. IP 주소의 등록을 취소하려면 IP 주소를 선택한 다음 등록 취소를 선택합니다.

Lambda 함수 등록 또는 등록 취소

대상 그룹에 단일 Lambda 함수를 등록할 수 있습니다. 트래픽을 Lambda 함수에 더 이상 전송할 필요가 없는 경우 해당 함수의 등록을 취소할 수 있습니다. Lambda 함수의 등록을 취소한 후에는 처리 중인 요청이 HTTP 5XX 오류와 함께 실패합니다. 대상 그룹에 대한 Lambda 함수를 바꾸는 대신 새 대상 그룹을 생성하는 것이 더 좋습니다.

콘솔을 사용하여 Lambda 함수를 등록 또는 등록 취소하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 대상 그룹을 선택합니다.
3. 대상 그룹의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 대상 탭을 선택합니다.
5. 등록된 Lambda 함수가 없는 경우 대상 등록을 선택합니다. Lambda 함수를 선택한 다음 대상 등록을 선택합니다.
6. Lambda 함수의 등록을 취소하려면 등록 취소를 선택합니다. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 등록 취소를 선택합니다.

Application Load Balancer 등록 또는 등록 취소

각 대상 그룹에 단일 Application Load Balancer를 등록할 수 있습니다. 트래픽을 로드 밸런서에 더 이상 전송할 필요가 없는 경우 로드 밸런서의 등록을 취소할 수 있습니다. 로드 밸런서의 등록을 취소한 후에는 처리 중인 요청이 HTTP 5XX 오류와 함께 실패합니다. 대상 그룹에 대한 Application Load Balancer를 바꾸는 대신 새 대상 그룹을 생성하는 것이 더 좋습니다.

콘솔을 사용하여 Application Load Balancer를 등록하거나 등록 취소하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 대상 그룹을 선택합니다.
3. 대상 그룹의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 대상 탭을 선택합니다.
5. 등록된 Application Load Balancer가 없는 경우 대상 등록을 선택합니다. Application Load Balancer를 선택하고 대상 등록을 선택합니다.
6. Application Load Balancer의 등록을 취소하려면 등록 취소를 선택합니다. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 등록 취소를 선택합니다.

VPC Lattice 대상 그룹에 대한 상태 확인

서비스는 등록된 대상으로 요청을 주기적으로 전송하여 상태를 확인합니다. 이러한 테스트를 바로 상태 확인이라고 합니다.

각 VPC Lattice 서비스는 정상 상태의 대상으로만 요청을 라우팅합니다. 각각의 서비스는 대상이 등록된 대상 그룹에 대한 상태 확인 설정을 사용하여 각 대상의 상태를 확인합니다. 대상이 등록된 후에는 상태 확인을 통과해야만 정상 상태로 간주됩니다. 각각의 상태 확인이 완료되고 나면 서비스는 상태 확인을 위해 설정된 연결을 종료합니다.

제한 사항 및 고려 사항

- 대상 그룹 프로토콜 버전이 HTTP1인 경우 상태 확인이 기본적으로 활성화됩니다.
- 대상 그룹 프로토콜 버전이 HTTP2인 경우 상태 확인은 기본적으로 활성화되지 않습니다. 하지만 상태 확인을 활성화하고 프로토콜 버전을 HTTP1 또는 HTTP2로 수동으로 설정할 수 있습니다.
- 상태 확인은 gRPC 대상 그룹 프로토콜 버전을 지원되지 않습니다. 하지만 상태 확인을 활성화하는 경우 상태 확인 프로토콜 버전을 HTTP1 또는 HTTP2로 지정해야 합니다.
- 상태 확인은 Lambda 대상 그룹을 지원하지 않습니다.
- 상태 확인은 Application Load Balancer 대상 그룹을 지원하지 않습니다. 하지만 Elastic Load Balancing을 사용하여 Application Load Balancer의 대상에 대한 상태 확인을 활성화할 수 있습니다. 자세한 내용은 Application Load Balancer 사용 설명서의 대상 그룹 상태 확인을 참조하세요.

상태 확인 설정

다음 표에 설명된 대로 대상 그룹의 대상에 대한 상태 확인을 구성합니다. 테이블에 사용되는 설정 이름은 API에 사용되는 이름입니다. 서비스는 지정된 포트, 프로토콜 및 ping 경로를 사용하여 HealthCheckIntervalSeconds 초마다 모든 등록 대상에 상태 확인 요청을 전송합니다. 각 상태 확인 요청은 독립적이며 결과는 전체 간격 동안 지속됩니다. 대상이 응답하는 데 걸리는 시간은 다음 상태 확인 요청의 간격에 영향을 미치지 않습니다. 상태 확인이 UnhealthyThresholdCount 연속 실패를 초과하면 서비스는 대상을 서비스에서 제외합니다. 상태 확인이 HealthyThresholdCount 연속 성공을 초과하면 서비스는 대상을 다시 서비스합니다.

설정	설명
HealthCheckProtocol	대상에 대한 상태 확인을 수행할 때 서비스가 사용하는 프로토콜입니다. HTTP, HTTPS 등의 프로토콜이 여기에 해당됩니다. HTTP 프로토콜이 기본 설정값입니다.
HealthCheckPort	대상에 대한 상태 확인을 수행할 때 서비스가 사용하는 포트입니다. 각 대상이 서비스에서 트래

설정	설명
	픽을 수신하는 포트를 사용하도록 기본 설정되어 있습니다.
HealthCheckPath	대상에 대한 상태 확인을 위한 대상입니다. 프로토콜 버전이 HTTP1 또는 HTTP2인 경우 유효한 URI(/path?query)를 참조하세요. 기본값은 /입니다.
HealthCheckTimeoutSeconds	상태 확인 실패를 의미하는 대상으로부터 응답이 없는 기간(초 단위)입니다. 범위는 1~120초입니다. 대상 유형이 `INSTANCE` 또는 `IP`인 경우 기본값은 5초입니다. 이 설정을 기본값으로 재설정하려면 0을 지정합니다.
HealthCheckIntervalSeconds	개별 인스턴스의 상태 확인 간의 대략적인 간격(초 단위)입니다. 범위는 5~300초입니다. 대상 유형이 `INSTANCE` 또는 `IP`인 경우 기본값은 30초입니다. 이 설정을 기본값으로 재설정하려면 0을 지정합니다.
HealthyThresholdCount	비정상 상태의 대상이 정상으로 간주되기까지 필요한 연속적인 상태 확인 성공 횟수입니다. 범위는 2~10회입니다. 기본값은 5입니다. 이 설정을 기본값으로 재설정하려면 0을 지정합니다.
UnhealthyThresholdCount	대상을 비정상 상태로 간주하기까지 필요한 연속적인 상태 확인 실패 횟수입니다. 범위는 2~10회입니다. 기본값은 2입니다. 이 설정을 기본값으로 재설정하려면 0을 지정합니다.

설정	설명
Matcher	대상으로부터 응답 성공을 확인할 때 사용하는 코드입니다. 이를 콘솔에서 성공 코드라고 합니다. 프로토콜 버전이 HTTP1 또는 HTTP2,인 경우 가능한 값은 200~499입니다. 값 범위(예: "200-299")에서 여러 값(예: "200,202")을 지정할 수 있습니다. 기본값은 200입니다. gRPC용 상태 확인 프로토콜 버전은 현재 지원되지 않습니다. 그러나 대상 그룹 프로토콜 버전이 gRPC인 경우 상태 확인 구성에서 HTTP1 또는 HTTP2 프로토콜 버전을 지정할 수 있습니다.

대상의 상태 확인

대상 그룹에 등록된 대상의 상태를 확인할 수 있습니다.

콘솔을 사용하여 대상의 상태를 확인하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 대상 그룹을 선택합니다.
3. 대상 그룹의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 대상 탭에서 상태 열은 각 대상의 상태를 나타냅니다. 상태가 `Healthy` 이외의 값인 경우에는 상태 세부 정보 열에 자세한 정보가 포함됩니다.

를 사용하여 대상의 상태를 확인하려면 AWS CLI

list-targets 명령을 사용합니다. 이 명령의 출력 화면에는 대상 상태 설명이 포함됩니다. 상태가 `Healthy` 이외의 값인 경우에는 출력 화면에도 사유 코드가 포함됩니다.

비정상 대상에 대한 이메일 알림을 받으려면

CloudWatch 경보를 통해 Lambda 함수를 시작하여 비정상 대상에 대한 세부 정보를 전송합니다.

상태 확인 설정 변경

대상 그룹에 대한 상태 확인 설정을 언제든지 변경할 수도 있습니다.

콘솔을 사용하여 상태 확인 설정을 변경하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 대상 그룹을 선택합니다.
3. 대상 그룹의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 상태 확인 탭에 있는 상태 확인 설정 섹션에서 편집을 선택합니다.
5. 필요에 따라 상태 확인 설정을 변경합니다.
6. 변경 사항 저장을 선택합니다.

를 사용하여 상태 확인 설정을 수정하려면 AWS CLI

update-target-group 명령을 사용합니다.

라우팅 구성

기본적으로 서비스는 대상 그룹을 생성할 때 지정한 프로토콜과 포트 번호를 사용하여 대상으로 요청을 라우팅합니다. 또는 대상 그룹에 등록할 때 대상으로 트래픽을 라우팅하는 데 사용되는 포트를 재정의할 수 있습니다.

대상 그룹은 다음과 같은 프로토콜 및 포트를 지원합니다.

- 프로토콜: HTTP, HTTPS, TCP
- 포트: 1-65535

대상 그룹이 HTTPS 프로토콜로 구성되거나 HTTPS 상태 확인을 사용하는 경우 대상에 대한 TLS 연결은 리스너의 보안 정책을 사용합니다. VPC Lattice는 대상에 설치하는 인증서를 사용하여 대상과 TLS 연결을 설정합니다. VPC Lattice는 이러한 인증서를 검증하지 않습니다. 따라서 자체 서명된 인증서 또는 만료된 인증서를 사용할 수 있습니다. VPC Lattice와 대상 간의 트래픽은 패킷 수준에서 인증되므로 대상의 인증서가 유효하지 않더라도 man-in-the-middle 공격 또는 스푸핑 위험이 없습니다.

TCP 대상 그룹은 TLS 리스너에서만 지원됩니다.

라우팅 알고리즘

기본적으로 라운드 로빈 라우팅 알고리즘은 정상 상태의 대상에 요청을 라우팅하는 데 사용됩니다.

VPC Lattice 서비스는 요청을 받으면 다음 프로세스를 사용합니다.

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

대상 그룹에 비정상인 대상만 포함되는 경우 정상 상태와 관계없이 모든 해당 대상에 요청이 라우팅됩니다. 즉, 모든 대상이 동시에 상태 확인에 실패하면 VPC Lattice 서비스가 열리지 않습니다. 오류 시 열림이 적용되면 상태에 관계없이 라운드 로빈 알고리즘에 따라 모든 대상에 대한 트래픽이 허용됩니다.

VPC Lattice는 트래픽 라우팅에 대한 가용 영역(AZ) 선호도를 지원합니다. 클라이언트가 VPC Lattice에 요청을 보내면 VPC Lattice는 클라이언트와 동일한 AZ의 서비스 또는 리소스에 대한 IP 주소로 응답합니다. 해당 AZ를 사용할 수 없는 경우 VPC Lattice는 다른 AZs. VPC Lattice에서 대상으로 라우팅은 AZs. 또한 VPC Lattice에는 AZ 간 데이터 전송 요금이 없습니다.

대상 유형

대상 그룹을 생성할 때 대상 유형을 지정합니다. 이 값에 따라 이 대상 그룹에 대상을 등록할 때 지정하는 대상의 유형이 결정됩니다. 대상 그룹을 생성한 후에는 대상 유형을 변경할 수 없습니다.

가능한 대상 유형은 다음과 같습니다.

`INSTANCE`

대상이 인스턴스 ID에 의해 지정됩니다.

`IP`

대상이 IP 주소입니다.

`LAMBDA`

대상이 Lambda 함수입니다.

`ALB`

대상이 Application Load Balancer입니다.

고려 사항

- 대상 유형이 IP인 경우 대상 그룹에 대한 VPC 서브넷의 IP 주소를 지정해야 합니다. 이 VPC 외부에서 IP 주소를 등록해야 하는 경우, ALB 유형의 대상 그룹을 생성하고 IP 주소를 Application Load Balancer에 등록합니다.
- 대상 유형이 IP인 경우 VPC 엔드포인트 또는 공개적으로 라우팅 가능한 IP 주소를 등록할 수 없습니다.
- 대상 유형이 LAMBDA인 경우 단일 Lambda 함수를 등록할 수 있습니다. 서비스가 Lambda 함수에 대한 요청을 수신하면 Lambda 함수를 호출합니다. 서비스에 여러 Lambda 함수를 등록하려면 여러 대상 그룹을 사용해야 합니다.
- 대상 유형이 인 경우 단일 내부 Application Load Balancer를 최대 2개의 VPC Lattice 서비스의 대상으로 등록할 ALB수 있습니다. 이렇게 하려면 Application Load Balancer를 두 개의 서로 다른 VPC Lattice 서비스에서 사용하는 별도의 두 대상 그룹에 등록해야 합니다. 또한 대상 Application Load Balancer에 대상 그룹 포트와 포트가 일치하는 리스너가 하나 이상 있어야 합니다.
- 시작 시 VPC Lattice 대상 그룹에 ECS 작업을 자동으로 등록할 수 있습니다. 대상 그룹에는 대상 유형이 IP여야 합니다. 자세한 내용은 Amazon Elastic Container Service 개발자 안내서의 Amazon ECS 서비스와 함께 VPC Lattice 사용을 참조하세요.

  또는 Amazon ECS 서비스에 대한 Application Load Balancer를 유형의 VPC Lattice 대상 그룹에 등록합니다ALB. 자세한 내용은 Amazon Elastic Container Service 개발자 안내서의 로드 밸런싱을 사용하여 Amazon ECS 서비스 트래픽 분산을 참조하세요.
- EKS 포드를 대상으로 등록하려면 Kubernetes 서비스에서 IP 주소를 가져오는 AWS 게이트웨이 API 컨트롤러를 사용하세요.
- 대상 그룹 프로토콜이 TCP인 경우 지원되는 유일한 대상 유형은 INSTANCE, IP또는 입니다ALB.

IP 주소 유형

대상 유형이 IP인 대상 그룹을 생성할 때 대상 그룹의 IP 주소 유형을 지정할 수 있습니다. 이는 로드 밸런서가 대상에 요청 및 상태 확인을 보내는 데 사용하는 주소 유형을 지정합니다. 가능한 값은 IPv4와 IPv6입니다. 기본값은 IPV4입니다.

고려 사항

- IP 주소 유형이 IPv6인 대상 그룹을 생성하는 경우 대상 그룹에 지정하는 VPC의 주소 범위는 IPv6이어야 합니다.

- 대상 그룹에 등록한 IP 주소는 대상 그룹의 IP 주소 유형과 일치해야 합니다. 예를 들어 IP 주소 유형이 IPv4인 경우 대상 그룹에 IPv6 주소를 등록할 수 없습니다.
- 대상 그룹에 등록하는 IP 주소는 대상 그룹에 지정한 VPC의 IP 주소 범위 내에 있어야 합니다.

VPC Lattice의 HTTP 대상

HTTP 요청 및 HTTP 응답은 헤더 필드를 사용하여 HTTP 메시지에 대한 정보를 전송합니다. HTTP 헤더가 자동으로 추가됩니다. 헤더 필드는 콜론으로 구분된 이름-값 페어이며 CR(캐리지 리턴) 및 LF(줄 바꿈)로 구분됩니다. HTTP 헤더 필드의 표준 집합은 RFC 2616, 메시지 헤더에 정의되어 있습니다. 자동으로 추가되고 애플리케이션에서 널리 사용되는 비표준 HTTP 헤더도 제공되고 있습니다. 예를 들어 접두사 `x-forwarded`가 붙은 비표준 HTTP 헤더가 있습니다.

x-forwarded 헤더

Amazon VPC Lattice는 다음의 `x-forwarded` 헤더를 추가합니다.

`x-forwarded-for`

소스 IP 주소.

`x-forwarded-port`

대상 포트.

`x-forwarded-proto`

네트워크 연결 프로토콜(`http` | `https`).

호출자 ID 헤더

Amazon VPC Lattice는 다음과 같은 호출자 ID 헤더를 추가합니다.

`x-amzn-lattice-identity`

자격 증명 정보. AWS 인증에 성공하면 다음 필드가 표시됩니다.

- `Principal` – 인증된 보안 주체.
- `PrincipalOrgID` – 인증된 보안 주체의 조직 ID.
- `PrincipalOrgPath` - 인증된 보안 주체의 조직 경로입니다.

- SessionName – 인증된 사용자의 사용자 이름.

Roles Anywhere 보안 인증을 사용하고 인증에 성공하면 다음 필드가 표시됩니다.

- X509Issuer/OU – 발급자(OU).
- X509SAN/DNS – 주체 대체 이름(DNS).
- X509SAN/NameCN – 발급자 대체 이름(이름/CN).
- X509SAN/URI – 주체 대체 이름(URI).
- X509Subject/CN – 주체 이름(CN).

`x-amzn-lattice-identity-tags`

보안 주체 ID 및 보안 주체 태그입니다. 형식은 다음과 같습니다.

```
principal=principal;principalorgid=orgid;principalorgpath=orgpath;principal-tag1=value1; ...;principal-tag99=value99
```

VPC Lattice는 백슬래시(\)가 있는 값의 세미콜론(;)을 이스케이프 처리합니다.

`x-amzn-lattice-network`

VPC. 형식은 다음과 같습니다.

```
SourceVpcArn=arn:aws:ec2:region:account:vpc/id
```

`x-amzn-lattice-target`

대상. 형식은 다음과 같습니다.

```
ServiceArn=arn;ServiceNetworkArn=arn;TargetGroupArn=arn
```

VPC Lattice의 리소스 ARN에 대한 자세한 내용은 Amazon VPC Lattice에서 정의한 리소스 유형을 참조하세요.

발신자 자격 증명 헤더는 스푸핑할 수 없습니다. VPC Lattice는 수신 요청에서 이러한 헤더를 제거합니다. 이러한 자격 증명 헤더는 다음 형식을 사용하여 빈 값을 지원하는 맵을 표시합니다. 구문 분석 시 이러한 헤더에 있는 KEYs의 특정 순서에 의존해서는 안 되며, 언제든지 새 KEYs 추가될 수 있으며 빈 값을 처리할 준비가 되어 있어야 합니다.

형식은 다음과 같습니다.

```
key-0=value-0;key-1=value-1;....;key-n=value-n;
```

VPC Lattice에서 대상으로서 Lambda 함수

Lambda 함수를 VPC Lattice 대상 그룹에 대상으로 등록하고 Lambda 함수에 대한 대상 그룹에 요청을 전달하도록 리스너 규칙을 구성할 수 있습니다. 서비스가 Lambda 함수를 대상으로 사용하는 대상 그룹에 요청을 전달하면 Lambda 함수를 호출하고 요청 콘텐츠를 Lambda 함수에 JSON 형식으로 전달합니다.

제한 사항

- Lambda 함수와 대상 그룹은 동일한 계정 및 동일한 리전에 있어야 합니다.
- Lambda 함수에 전송할 수 있는 요청 본문의 최대 크기는 6MB입니다.
- Lambda 함수가 전송할 수 있는 응답 JSON의 최대 크기는 6MB입니다.
- 프로토콜은 HTTP 또는 HTTPS여야 합니다.

Lambda 함수 준비

다음 권장 사항은 와 함께 VPC Lattice 서비스에 Lambda 함수를 사용하는 경우에 적용됩니다.

Lambda 함수를 호출할 권한

대상 그룹을 생성하고 AWS Management Console 또는를 사용하여 Lambda 함수를 등록하면 AWS CLI VPC Lattice는 사용자를 대신하여 Lambda 함수 정책에 필요한 권한을 추가합니다.

다음 API 직접 호출을 사용하여 직접 권한을 추가할 수도 있습니다.

```
aws lambda add-permission \
  --function-name lambda-function-arn-with-alias-name \
  --statement-id vpc-lattice \
  --principal vpc-lattice.amazonaws.com \
  --action lambda:InvokeFunction \
  --source-arn target-group-arn
```

Lambda 함수 버전 관리

대상 그룹당 하나의 Lambda 함수를 등록할 수 있습니다. Lambda 함수를 변경할 수 있는지 확인하고 VPC Lattice 서비스가 항상 현재 버전의 Lambda 함수를 호출하도록 하려면 Lambda 함수를 VPC

Lattice 서비스에 등록할 때 함수 별칭을 생성하고 별칭을 함수 ARN에 포함시킵니다. 자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수 버전 및 Lambda 함수의 별칭 생성을 참조하세요.

Lambda 함수에 대한 대상 그룹 생성

라우팅 요청에서 사용되는 대상 그룹을 만듭니다. 요청 콘텐츠가 해당 콘텐츠를 이 대상 그룹에 전달하는 작업이 포함된 리스너 규칙과 일치하는 경우 VPC Lattice 서비스는 등록된 Lambda 함수를 호출합니다.

콘솔을 사용하여 대상 그룹을 생성하고 Lambda 함수를 등록하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 대상 그룹을 선택합니다.
3. 대상 그룹 생성을 선택합니다.
4. 대상 유형에서 Lambda 함수를 선택합니다.
5. 대상 그룹 이름에 대상 그룹의 이름을 입력합니다.
6. Lambda 이벤트 구조 버전에서 버전을 선택합니다. 자세한 내용은 the section called “VPC Lattice 서비스에서 이벤트 수신” 단원을 참조하십시오.
7. (선택 사항) 태그를 추가하려면 태그를 확장하고 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
8. 다음을 선택합니다.
9. Lambda 함수에 대해 다음 중 하나를 수행합니다.
   - 기존 Lambda 함수를 선택합니다.
   - 새 Lambda 함수를 생성하고 선택합니다.
   - 나중에 Lambda 함수를 등록합니다.
10. 대상 그룹 생성을 선택합니다.

를 사용하여 대상 그룹을 생성하고 Lambda 함수를 등록하려면 AWS CLI

create-target-group 및 register-targets 명령을 사용합니다.

VPC Lattice 서비스에서 이벤트 수신

VPC Lattice 서비스는 HTTP 및 HTTPS를 통한 요청에 대한 Lambda 호출을 지원합니다. 서비스는 JSON 형식으로 이벤트를 전송하고 모든 요청에 `X-Forwarded-For` 헤더를 추가합니다.

Base64 인코딩

Base64 서비스는 content-encoding 헤더가 존재하고 콘텐츠 유형이 다음 중 하나가 아닌 경우 본문을 인코딩합니다.

- text/*
- application/json
- application/xml
- application/javascript

content-encoding 헤더가 없으면 Base64 인코딩은 콘텐츠 유형에 따라 다릅니다. 위 콘텐츠 유형의 경우 서비스는 Base64 인코딩 없이 본문을 있는 그대로 전송합니다.

이벤트 구조 형식

LAMBDA 유형의 대상 그룹을 생성하거나 업데이트할 때 Lambda 함수가 수신하는 이벤트 구조의 버전을 지정할 수 있습니다. 가능한 버전은 V1 및 V2입니다.

Example예시 이벤트: V2

```
{
    "version": "2.0",
    "path": "/?query1=value1&query2=value2",
    "method": "GET|POST|HEAD|...",
    "headers": {
        "header-key": ["header-value", ...],
        ...
    },
    "queryStringParameters": {
        "key": ["value", ...]
    },
    "body": "request-body",
    "isBase64Encoded": true|false,
    "requestContext": {
        "serviceNetworkArn": "arn:aws:vpc-
lattice:region:123456789012:servicenetwork/sn-0bf3f2882e9cc805a",
        "serviceArn": "arn:aws:vpc-
lattice:region:123456789012:service/svc-0a40eebed65f8d69c",
        "targetGroupArn": "arn:aws:vpc-
lattice:region:123456789012:targetgroup/tg-6d0ecf831eec9f09",
        "identity": {
```

```
            "sourceVpcArn":
 "arn:aws:ec2:region:123456789012:vpc/vpc-0b8276c84697e7339",
            "type": "AWS_IAM",
            "principal": "arn:aws:iam::123456789012:assumed-role/my-role/my-session",
            "principalOrgID": "o-50dc6c495c0c9188",
            "sessionName": "i-0c7de02a688bde9f7",
            "x509IssuerOu": "string",
            "x509SanDns": "string",
            "x509SanNameCn": "string",
            "x509SanUri": "string",
            "x509SubjectCn": "string"
        },
        "region": "region",
        "timeEpoch": "1690497599177430"
    }
}
```

`body`

요청의 본문. 프로토콜이 HTTP, HTTPS 또는 gRPC인 경우에만 표시됩니다.

`headers`

요청의 HTTP 상태. 프로토콜이 HTTP, HTTPS 또는 gRPC인 경우에만 표시됩니다.

`identity`

자격 증명 정보. 가능한 필드는 다음과 같습니다.

- `principal` – 인증된 보안 주체. AWS 인증이 성공한 경우에만 표시됩니다.
- `principalOrgID` – 인증된 보안 주체의 조직 ID. AWS 인증이 성공한 경우에만 표시됩니다.
- `sessionName` – 인증된 사용자의 사용자 이름. AWS 인증이 성공한 경우에만 표시됩니다.
- `sourceVpcArn` – 요청이 시작된 VPC의 ARN. 소스 VPC를 식별할 수 있는 경우에만 표시됩니다.
- `type` - 인증 정책이 사용되고 AWS 인증이 성공한 `AWS_IAM` 경우 값은 입니다.

Roles Anywhere 보안 인증을 사용하고 인증에 성공하면 가능한 필드는 다음과 같습니다.

- `x509IssuerOu` – 발급자(OU).
- `x509SanDns` – 주체 대체 이름(DNS).
- `x509SanNameCn` – 발급자 대체 이름(이름/CN).
- `x509SanUri` – 주체 대체 이름(URI).

- x509SubjectCn – 주체 이름(CN).

isBase64Encoded

본문이 base64로 인코딩되었는지 여부를 나타냅니다. 프로토콜이 HTTP, HTTPS 또는 gRPC이고 요청 본문이 아직 문자열이 아닌 경우에만 표시됩니다.

method

요청의 HTTP 메서드. 프로토콜이 HTTP, HTTPS 또는 gRPC인 경우에만 표시됩니다.

path

쿼리 문자열 파라미터를 포함하는 클라이언트의 요청 경로입니다. 프로토콜이 HTTP, HTTPS 또는 gRPC인 경우에만 표시됩니다.

queryStringParameters

HTP 쿼리 문자열 파라미터. 프로토콜이 HTTP, HTTPS 또는 gRPC인 경우에만 표시됩니다.

serviceArn

요청을 수신하는 서비스의 ARN.

serviceNetworkArn

요청을 전달하는 서비스 네트워크의 ARN.

targetGroupArn

요청을 수신하는 대상 그룹의 ARN.

timeEpoch

시간(초 단위).

Example예시 이벤트: V1

```
{
    "raw_path": "/path/to/resource?query1=value1&query2=value2",
    "method": "GET|POST|HEAD|...",
    "headers": {"header-key": "header-value", ... },
    "query_string_parameters": {"key": "value", ...},
    "body": "request-body",
    "is_base64_encoded": true|false
```

```
}
```

VPC Lattice 서비스에 응답

Lambda 함수의 응답에는 Base64 인코딩 상태, 상태 코드 및 헤더가 포함됩니다. 본문을 생략할 수 있습니다.

응답의 본문에 바이너리 콘텐츠를 포함시키려면 콘텐츠를 Base64로 인코딩하고 `isBase64Encoded`를 `true`를 설정해야 합니다. 서비스는 콘텐츠를 디코딩하여 바이너리 콘텐츠를 수신하고 이 콘텐츠를 HTTP 응답의 본문으로 클라이언트에 전송합니다.

VPC Lattice 서비스는 `Connection` 또는 `Transfer-Encoding`과 같은 hop-by-hop 헤더를 적용하지 않습니다. 응답을 클라이언트에 전송하기 전에 서비스가 컴퓨팅하기 때문에 `Content-Length` 헤더를 생략할 수 있습니다.

다음은 Lambda 함수의 응답 예시입니다.

```
{
    "isBase64Encoded": false,
    "statusCode": 200,
    "headers": {
        "Set-cookie": "cookies",
        "Content-Type": "application/json"
    },
    "body": "Hello from Lambda (optional)"
}
```

다중 값 헤더

VPC Lattice는 클라이언트의 요청 또는 값이 여러 개인 헤더를 포함하거나 동일한 헤더를 여러 번 포함하는 Lambda 함수의 응답을 지원합니다. VPC Lattice는 모든 값을 대상으로 전달합니다.

다음 예제에는 값이 다른 라는 두 개의 헤더header1가 있습니다.

```
header1 = value1
header1 = value2
```

V2 이벤트 구조를 사용하면 VPC Lattice가 목록의 값을 전송합니다. 예제:

```
"header1": ["value1", "value2"]
```

V1 이벤트 구조를 사용하면 VPC Lattice가 값을 단일 문자열로 결합합니다. 예제:

```
"header1": "value1, value2"
```

다중 값 쿼리 문자열 파라미터

VPC Lattice는 동일한 키에 대해 여러 값이 있는 쿼리 파라미터를 지원합니다.

다음 예제에는 값이 다른 라는 두 개의 파라미터QS1가 있습니다.

```
http://www.example.com?&QS1=value1&QS1=value2
```

V2 이벤트 구조를 사용하면 VPC Lattice가 목록의 값을 전송합니다. 예제:

```
"QS1": ["value1", "value2"]
```

V1 이벤트 구조의 경우 VPC Lattice는 전달된 마지막 값을 사용합니다. 예제:

```
"QS1": "value2"
```

Lambda 함수 등록 취소

트래픽을 Lambda 함수에 더 이상 전송할 필요가 없는 경우 해당 함수의 등록을 취소할 수 있습니다. Lambda 함수의 등록을 취소한 후에는 처리 중인 요청이 HTTP 5XX 오류와 함께 실패합니다.

Lambda 함수를 바꾸려면 새 대상 그룹을 생성하고, 새 함수를 새 대상 그룹에 등록한 다음, 새 대상 그룹을 기존 대상 그룹 대신 사용하도록 리스너 규칙을 업데이트하는 것이 좋습니다.

콘솔을 사용하여 Lambda 함수의 등록을 취소하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 대상 그룹을 선택합니다.
3. 대상 그룹의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 대상 탭에서 등록 취소를 선택합니다.
5. 확인 메시지가 나타나면 **confirm**을 입력한 다음 등록 취소를 선택합니다.

를 사용하여 Lambda 함수의 등록을 취소하려면 AWS CLI

deregister-targets 명령을 사용합니다.

VPC Lattice의 대상인 Application Load Balancer

VPC Lattice 대상 그룹을 생성하고 단일 내부 Application Load Balancer를 대상으로 등록하고 트래픽을 이 대상 그룹으로 전달하도록 VPC Lattice 서비스를 구성할 수 있습니다. 이 시나리오에서는 Application Load Balancer가 트래픽이 도달하는 즉시 라우팅 결정을 인계합니다. 이 구성을 사용하면 Application Load Balancer의 계층 7 요청 기반 라우팅 기능을 VPC Lattice가 지원하는 기능(예: IAM 인증 및 권한 부여, VPC와 계정 간의 연결)과 함께 사용할 수 있습니다.

제한 사항

- VPC Lattice 대상 그룹 유형 `ALB`의 대상으로 단일 내부 Application Load Balancer를 등록할 수 있습니다.
- Application Load Balancer를 서로 다른 두 VPC Lattice 서비스에서 사용하는 최대 2개의 VPC Lattice 대상 그룹에 대한 대상으로 등록할 수 있습니다.
- VPC Lattice는 `ALB` 유형 대상 그룹에 대한 상태 확인을 제공하지 않습니다. 하지만 Elastic Load Balancing에서 대상에 대한 로드 밸런서 수준에서 독립적으로 상태 확인을 구성할 수 있습니다. 자세한 내용은 Application Load Balancer 사용 설명서의 대상 그룹 상태 확인을 참조하세요.

사전 조건

Application Load Balancer를 생성하여 VPC Lattice 대상 그룹에 대상으로 등록합니다. 로드 밸런서는 다음 기준을 충족해야 합니다.

- 로드 밸런서 스키마는 내부입니다.
- Application Load Balancer는 VPC Lattice 대상 그룹과 동일한 계정에 있어야 하며 활성 상태여야 합니다.
- Application Load Balancer는 VPC Lattice 대상 그룹과 동일한 VPC에 있어야 합니다.
- Application Load Balancer에서 HTTPS 리스너를 사용하여 TLS를 종료할 수 있지만 VPC Lattice 서비스가 로드 밸런서와 동일한 SSL/TLS 인증서를 사용하는 경우에 한합니다.
- `X-Forwarded-For` 요청 헤더에 VPC Lattice 서비스의 클라이언트 IP를 보존하려면 Application Load Balancer의 속성 `routing.http.xff_header_processing.mode`를 `Preserve`로 설정해야 합니다. 값이 `Preserve`인 경우, 로드 밸런서가 HTTP 요청의 `X-Forwarded-For` 헤더를 보존하고, 요청을 변경 없이 대상에 보냅니다.

자세한 내용은 Application Load Balancers 사용 설명서의 Application Load Balancer 생성을 참조하세요.

1단계: ALB 유형의 대상 그룹 생성

대상 그룹을 생성하려면 다음 절차를 따르세요. VPC Lattice는 ALB 대상 그룹에 대한 상태 확인을 지원하지 않습니다. 하지만 Application Load Balancer의 대상 그룹에 대한 상태 확인은 구성할 수 있습니다. 자세한 내용은 Application Load Balancer 사용 설명서의 대상 그룹 상태 확인을 참조하세요.

대상 그룹을 생성하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 대상 그룹을 선택합니다.
3. 대상 그룹 생성을 선택합니다.
4. 대상 그룹 세부 정보 지정 페이지의 기본 구성에서 대상 유형으로 Application Load Balancer를 선택합니다.
5. 대상 그룹 이름에 대상 그룹의 이름을 입력합니다.
6. 프로토콜에서 , 또는 **HTTPHTTPS**를 선택합니다**TCP**. 대상 그룹 프로토콜은 내부 Application Load Balancer의 리스너 프로토콜과 일치해야 합니다.
7. 포트에서 대상 그룹에 대한 포트를 지정합니다. 이 포트는 내부 Application Load Balancer의 리스너 포트와 일치해야 합니다. 또는 여기에서 지정하는 대상 그룹 포트와 일치하도록 내부 Application Load Balancer에 리스너 포트를 추가할 수 있습니다.
8. VPC 에서 내부 Application Load Balancer를 생성할 때 선택한 것과 동일한 Virtual Private Cloud(VPC)를 선택합니다. 이것은 VPC Lattice 리소스가 포함된 VPC여야 합니다.
9. 프로토콜 버전에서 Application Load Balancer가 지원하는 프로토콜 버전을 선택합니다.
10. (선택 사항) 필수 태그를 추가합니다.
11. 다음을 선택합니다.

2단계: Application Load Balancer를 대상으로 등록

로드 밸런서를 지금 또는 나중에 대상으로 등록할 수 있습니다.

Application Load Balancer를 대상으로 등록하는 방법

1. 지금 등록을 선택합니다.

2. Application Load Balancer에서 내부 Application Load Balancer를 선택합니다.
3. 포트에서 기본값을 유지하거나 필요에 따라 다른 포트를 지정합니다. 이 포트는 Application Load Balancer의 기존 리스너 포트와 일치해야 합니다. 일치하는 포트 없이 작업을 계속하면 트래픽이 Application Load Balancer에 도달하지 않습니다.
4. 대상 그룹 생성을 선택합니다.

프로토콜 버전

기본적으로, 서비스는 HTTP/1.1을 사용하여 대상에 요청을 보냅니다. 프로토콜 버전을 사용하면 HTTP/2 또는 gRPC를 사용하여 대상에 요청을 보낼 수 있습니다.

다음 표에는 요청 프로토콜과 대상 그룹 프로토콜 버전의 조합에 대한 결과가 요약되어 있습니다.

요청 프로토콜	프로토콜 버전	결과
HTTP/1.1	HTTP/1.1	성공
HTTP/2	HTTP/1.1	성공
gRPC	HTTP/1.1	오류
HTTP/1.1	HTTP/2	오류
HTTP/2	HTTP/2	성공
gRPC	HTTP/2	대상이 gRPC를 지원하는 경우 성공
HTTP/1.1	gRPC	오류
HTTP/2	gRPC	POST 요청 시 성공
gRPC	gRPC	성공

gRPC 프로토콜 버전에 대한 고려 사항

- 지원되는 유일한 리스너 프로토콜은 HTTPS입니다.
- 지원되는 유일한 대상 유형은 `INSTANCE` 및 `IP`입니다.

- 서비스는 gRPC 요청을 구문 분석하고 패키지, 서비스 및 메서드를 기반으로 gRPC 호출을 적절한 대상 그룹으로 라우팅합니다.
- Lambda 함수를 대상으로 사용할 수 없습니다.

HTTP/2 프로토콜 버전에 대한 고려 사항

- 지원되는 유일한 리스너 프로토콜은 HTTPS입니다. 대상 그룹 프로토콜로 HTTP 또는 HTTPS를 선택할 수 있습니다.
- 지원되는 유일한 리스너 규칙은 순방향 및 고정 응답입니다.
- 지원되는 유일한 대상 유형은 `INSTANCE` 및 `IP`입니다.
- 서비스는 클라이언트에서 오는 스트리밍을 지원합니다. 서비스는 대상으로 가는 스트리밍을 지원하지 않습니다.

VPC Lattice 대상 그룹에 대한 태그

태그를 사용하면 용도, 소유자 또는 환경 등에 따라 대상 그룹을 다양한 방식으로 분류할 수 있습니다.

각 대상 그룹에 여러 태그를 추가할 수 있습니다. 태그 키는 대상 그룹별로 고유해야 합니다. 대상 그룹에 이미 연결된 키를 통해 태그를 추가하면 해당 태그의 값이 업데이트됩니다.

사용이 끝난 태그는 삭제할 수 있습니다.

제한 사항

- 리소스당 최대 태그 수 - 50개
- 최대 키 길이 - 유니코드 문자 127자
- 최대 값 길이 - 유니코드 문자 255자
- 태그 키와 값은 대소문자를 구분합니다. 허용되는 문자는 UTF-8로 표현할 수 있는 문자, 공백 및 숫자와 특수 문자 + - = . _ : / @입니다. 선행 또는 후행 공백을 사용하면 안 됩니다.
- 태그 이름 또는 값에 `aws:` 접두사는 AWS 사용하도록 예약되어 있으므로 사용하지 마십시오. 이 접두사가 지정된 태그 이름이나 값은 편집하거나 삭제할 수 없습니다. 이 접두사가 지정된 태그는 리소스당 태그 수 제한에 포함되지 않습니다.

콘솔을 사용하여 대상 그룹 태그를 업데이트하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.

2. 탐색 창의 VPC Lattice에서 대상 그룹을 선택합니다.
3. 대상 그룹의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 태그 탭을 선택합니다.
5. 태그를 추가하려면 태그 추가를 선택한 다음 태그 키와 태그 값을 입력합니다. 다른 태그를 추가하려면 새 태그 추가를 선택합니다. 태그 추가가 완료되면 변경 사항 저장을 선택합니다.
6. 태그를 삭제하려면 태그의 확인란을 선택한 다음 삭제를 선택합니다. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 삭제를 선택합니다.

를 사용하여 대상 그룹의 태그를 업데이트하려면 AWS CLI

tag-resource 및 untag-resource 명령을 사용합니다.

VPC Lattice 대상 그룹 삭제

리스너 규칙의 전달 작업에서 참조하지 않는 대상 그룹을 삭제할 수 있습니다. 대상 그룹을 삭제해도 대상 그룹에 등록된 대상에는 영향을 미치지 않습니다. 등록된 EC2 인스턴스가 더 이상 필요하지 않은 경우 중지 또는 종료할 수 있습니다.

콘솔을 사용하여 대상 그룹을 삭제하는 방법

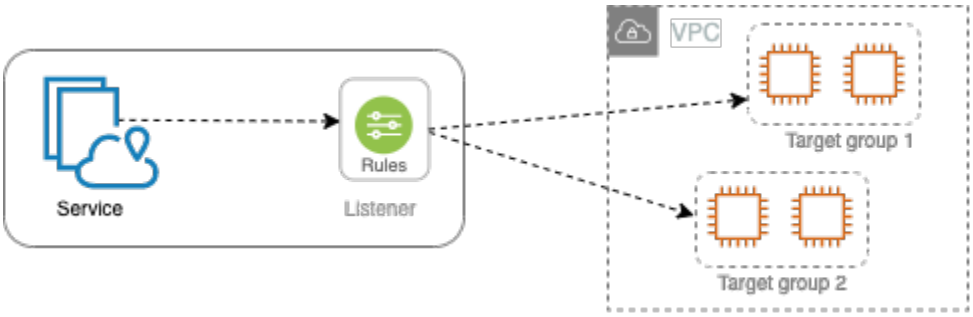
1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창에서 대상 그룹을 선택합니다.
3. 대상 그룹에 대한 확인란을 선택한 후 작업, 삭제를 선택합니다.
4. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 삭제를 선택합니다.

를 사용하여 대상 그룹을 삭제하려면 AWS CLI

delete-target-group 명령을 사용합니다.

VPC Lattice 서비스를 위한 리스너

VPC Lattice 서비스 사용을 시작하기 전에 리스너를 추가해야 합니다. 리스너는 구성한 프로토콜 및 포트를 사용하여 연결 요청을 확인하는 프로세스입니다. 리스너에 대해 정의한 규칙에 따라 서비스가 등록된 대상으로 요청을 라우팅하는 방법이 결정됩니다.



내용

리스너 구성

리스너는 다음과 같은 프로토콜 및 포트를 지원합니다.

- 프로토콜: HTTP, HTTPS, TLS
- 포트: 1-65535

리스너 프로토콜이 HTTPS인 경우 VPC Lattice는 VPC Lattice에서 생성한 FQDN과 연결된 TLS 인증서를 프로비저닝하고 관리합니다. VPC Lattice는 HTTP/1.1 및 HTTP/2에서 TLS를 지원합니다. HTTPS 리스너로 서비스를 구성하면 VPC Lattice는 ALPN(Application-Layer Protocol Negotiation)을 사용하여 HTTP 프로토콜을 자동으로 결정합니다. ALPN이 없는 경우 VPC Lattice는 HTTP/1.1을 기본값으로 사용합니다. 자세한 내용은 HTTPS 리스너 단원을 참조하십시오.

VPC Lattice는 HTTP, HTTPS, HTTP/1.1, HTTP/2에서 수신 대기하고 이러한 프로토콜 및 버전의 모든 대상과 통신할 수 있습니다. 리스너와 대상 그룹 프로토콜이 일치할 필요는 없습니다. VPC Lattice는 프로토콜과 버전 간 업그레이드와 다운그레이드의 전체 프로세스를 관리합니다. 자세한 내용은 프로토콜 버전 단원을 참조하십시오.

애플리케이션이 VPC Lattice 대신 암호화된 트래픽을 복호화하도록 TLS 리스너를 생성할 수 있습니다. 자세한 내용은 TLS 리스너 단원을 참조하십시오.

VPC Lattice는 기본적으로 WebSockets를 지원하지 않습니다. 그러나 TLS 리스너를 사용하거나 VPC Lattice 리소스를 통해 라우팅하여 여전히 Websocket 기반 서비스에 연결할 수 있습니다.

VPC Lattice 서비스를 위한 HTTP 리스너

리스너는 연결 요청을 확인하는 프로세스입니다. VPC Lattice 서비스를 생성할 때 리스너를 정의할 수 있습니다. 언제라도 서비스에 리스너를 추가할 수 있습니다.

이 페이지의 정보는 서비스용 HTTP 리스너를 생성하는 데 도움이 됩니다. 다른 프로토콜을 사용하는 리스너 생성에 대한 자세한 내용은 HTTPS 리스너 및 섹션을 참조하세요TLS 리스너.

사전 조건

- 기본 리스너 규칙에 전달 작업을 추가하려면 사용 가능한 VPC Lattice 대상 그룹을 지정해야 합니다. 자세한 내용은 VPC Lattice 대상 그룹 생성 단원을 참조하십시오.
- 여러 리스너에서 동일한 대상 그룹을 지정할 수 있지만, 이러한 리스너는 동일한 서비스에 속해야 합니다. 대상 그룹을 VPC Lattice 서비스에서 사용하려면 대상 그룹이 VPC Lattice 서비스용으로 리스너에서 사용되고 있지 않은지 확인해야 합니다.

HTTP 리스너 추가

언제라도 서비스에 리스너 및 규칙을 추가할 수 있습니다. 리스너에서 클라이언트에서 서비스로의 연결을 위한 프로토콜 및 포트 번호와 기본 리스너 규칙에 대한 VPC Lattice 대상 그룹을 구성합니다. 자세한 내용은 리스너 구성 단원을 참조하십시오.

콘솔을 사용하여 HTTP 리스너를 추가하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.

3. 서비스 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 라우팅 탭에서 리스너 추가를 선택합니다.
5. 리스너 이름에서 사용자 지정 리스너 이름을 제공하거나 리스너의 프로토콜과 포트를 리스너 이름으로 사용할 수 있습니다. 지정할 사용자 지정 이름은 최대 63자까지 가능하며 계정의 모든 서비스마다 고유해야 합니다. 유효한 문자는 a~z, 0~9, 하이픈(–)입니다. 하이픈은 첫 문자 또는 마지막 문자로 사용할 수도 없고 다른 하이픈 바로 뒤에 사용할 수도 없습니다. 이름은 생성한 후에는 변경할 수 없습니다.
6. 프로토콜: 포트에서 HTTP를 선택하고 포트 번호를 입력합니다.
7. 기본 동작에서, 트래픽을 수신할 VPC Lattice 대상 그룹을 선택하고 이 대상 그룹에 할당할 가중치를 선택합니다. 대상 그룹에 할당하는 가중치는 트래픽을 수신할 우선 순위를 설정합니다. 예를 들어, 두 대상 그룹의 가중치가 같으면 각 대상 그룹은 트래픽의 절반을 수신합니다. 대상 그룹을 하나만 지정한 경우에는 트래픽의 100%가 하나의 대상 그룹으로 전송됩니다.

   필요한 경우 기본 작업에 다른 대상 그룹을 추가할 수 있습니다. 작업 추가를 선택한 다음 대상 그룹을 선택하고 가중치를 지정합니다.
8. (선택 사항) 다른 규칙을 추가하려면 규칙 추가를 선택한 다음, 규칙의 이름, 우선 순위, 조건 및 작업을 입력합니다.

   각 규칙에 1에서 100 사이의 우선 순위 번호를 부여할 수 있습니다. 리스너는 우선 순위가 동일한 규칙을 여러 개 자질 수 없습니다. 규칙은 가장 낮은 값에서 가장 높은 값에 이르기까지 우선 순위에 따라 평가됩니다. 기본 규칙은 마지막에 평가됩니다. 자세한 내용은 리스너 규칙 단원을 참조하십시오.
9. (선택 사항) 태그를 추가하려면 리스너 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.
10. 구성을 검토하고 추가를 선택합니다.

를 사용하여 HTTP 리스너를 추가하려면 AWS CLI

기본 규칙으로 리스너를 생성하려면 create-listener 명령을, 추가 리스너 규칙을 생성하려면 create-rule 명령을 사용합니다.

VPC Lattice 서비스를 위한 HTTPS 리스너

리스너는 연결 요청을 확인하는 프로세스입니다. 리스너는 서비스를 생성할 때 정의합니다. 언제든지 VPC Lattice에서 서비스에 리스너를 추가할 수 있습니다.

TLS 버전 1.2 또는 TLS 버전 1.3을 사용하여 VPC Lattice와의 HTTPS 연결을 직접 종료하는 HTTPS 리스너를 생성할 수 있습니다. VPC Lattice는 VPC Lattice에서 생성한 FQDN(정규화된 도메인 이름)과 연결된 TLS 인증서를 프로비저닝하고 관리합니다. VPC Lattice는 HTTP/1.1 및 HTTP/2에서 TLS를 지원합니다. HTTPS 리스너로 서비스를 구성하면 VPC Lattice는 ALPN(Application-Layer Protocol Negotiation)을 통해 HTTP 프로토콜을 자동으로 결정합니다. ALPN이 없는 경우 VPC Lattice는 HTTP/1.1을 기본값으로 사용합니다.

VPC Lattice는 멀티 테넌시 아키텍처를 사용하므로 동일한 엔드포인트에서 여러 서비스를 호스팅할 수 있습니다. VPC Lattice에서는 모든 클라이언트 요청에 대해 SNI(서버 이름 표시)와 함께 TLS를 사용합니다. Encrypted Client Hello(ECH) 및 Encrypted Server Name Indication(ESNI)은 지원되지 않습니다.

VPC Lattice는 HTTP, HTTPS, HTTP/1.1, HTTP/2에서 수신 대기하고 이러한 프로토콜 및 버전의 모든 대상과 통신할 수 있습니다. 이러한 리스너와 대상 그룹 구성은 일치하지 않아도 됩니다. VPC Lattice는 프로토콜과 버전 간 업그레이드와 다운그레이드의 전체 프로세스를 관리합니다. 자세한 내용은 프로토콜 버전 단원을 참조하십시오.

애플리케이션이 트래픽을 해독하도록 하려면 대신 TLS 리스너를 생성합니다. TLS 패스스루를 사용하면 VPC Lattice는 TLS를 종료하지 않습니다. 자세한 내용은 TLS 리스너 단원을 참조하십시오.

목차

보안 정책

VPC Lattice는 TLSv1.2 프로토콜과 SSL/TLS 암호 목록을 조합한 보안 정책을 사용합니다. 프로토콜은 클라이언트와 서버 간에 보안 연결을 설정하여 클라이언트와 VPC Lattice 간에 전달되는 모든 데이터를 안전하게 보호합니다. 암호는 코딩된 메시지를 생성하기 위해 암호화 키를 사용하는 암호화 알고리즘입니다. 프로토콜은 여러 개의 암호를 사용해 데이터를 암호화합니다. 연결 협상이 이루어지는 동안 클라이언트와 VPC Lattice는 각각이 지원하는 암호 및 프로토콜 목록을 선호도 순으로 표시합니다. 기본적으로 서버의 목록에서 클라이언트의 암호 중 하나와 일치하는 첫 번째 암호가 보안 연결을 위해 선택됩니다.

VPC Lattice는 다음 TLS 1.2 SSL/TLS 암호를이 기본 설정 순서로 사용합니다.

- `ECDHE-RSA-AES128-GCM-SHA256`

- ECDHE-RSA-AES128-SHA
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-SHA
- AES128-GCM-SHA256
- AES128-SHA
- AES256-GCM-SHA384
- AES256-SHA

또한 VPC Lattice는 다음 TLS 1.3 SSL/TLS 암호를 이러한 기본 설정 순서로 사용합니다.

- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256

ALPN 정책

ALPN(Application-Layer Protocol Negotiation)은 최초 TLS 핸드셰이크 hello 메시지를 통해 전송되는 TLS 확장입니다. ALPN을 사용하면 애플리케이션 계층이 HTTP/1 및 HTTP/2 같은 보안 연결을 통해 사용해야 하는 프로토콜을 협상할 수 있습니다.

클라이언트가 ALPN 연결을 시작하면 VPC Lattice 서비스는 클라이언트 ALPN 기본 설정 목록을 해당 ALPN 정책과 비교합니다. 클라이언트가 ALPN 정책의 프로토콜을 지원하는 경우 VPC Lattice 서비스는 ALPN 정책의 기본 설정 목록을 기반으로 연결을 설정합니다. 그렇지 않으면 서비스는 ALPN을 사용하지 않습니다.

VPC Lattice는 다음과 같은 ALPN 정책을 지원합니다.

HTTP2Preferred

HTTP/1.1보다 HTTP/2를 선호합니다. ALPN 기본 설정 목록은 h2, http/1.1입니다.

HTTPS 리스너 추가

리스너에서 클라이언트에서 서비스로의 연결을 위한 프로토콜 및 포트 번호와 기본 리스너 규칙에 대한 대상 그룹을 구성합니다. 자세한 내용은 리스너 구성 단원을 참조하십시오.

사전 조건

- 기본 리스너 규칙에 전달 작업을 추가하려면 사용 가능한 VPC Lattice 대상 그룹을 지정해야 합니다. 자세한 내용은 VPC Lattice 대상 그룹 생성 단원을 참조하십시오.
- 여러 리스너에서 동일한 대상 그룹을 지정할 수 있지만, 이러한 리스너는 동일한 VPC Lattice 서비스에 속해야 합니다. 대상 그룹을 VPC Lattice 서비스에서 사용하려면 대상 그룹이 VPC Lattice 서비스용으로 리스너에서 사용되고 있지 않은지 확인해야 합니다.
- VPC Lattice에서 제공하는 인증서를 사용하거나 자체 인증서를 가져올 수 있습니다 AWS Certificate Manager. 자세한 내용은 the section called “BYOC” 단원을 참조하십시오.

콘솔을 사용하여 HTTPS 리스너를 추가하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 라우팅 탭에서 리스너 추가를 선택합니다.
5. 리스너 이름에서 사용자 지정 리스너 이름을 제공하거나 리스너의 프로토콜과 포트를 리스너 이름으로 사용할 수 있습니다. 지정할 사용자 지정 이름은 최대 63자까지 가능하며 계정의 모든 서비스마다 고유해야 합니다. 유효한 문자는 a~z, 0~9, 하이픈(-)입니다. 하이픈은 첫 문자 또는 마지막 문자로 사용할 수도 없고 다른 하이픈 바로 뒤에 사용할 수도 없습니다. 리스너를 생성한 후에는 리스너의 이름을 변경할 수 없습니다.
6. 프로토콜: 포트에서 HTTPS를 선택하고 포트 번호를 입력합니다.
7. 기본 동작에서, 트래픽을 수신할 VPC Lattice 대상 그룹을 선택하고 이 대상 그룹에 할당할 가중치를 선택합니다. 대상 그룹에 할당하는 가중치는 트래픽을 수신할 우선 순위를 설정합니다. 예를 들어, 두 대상 그룹의 가중치가 같으면 각 대상 그룹은 트래픽의 절반을 수신합니다. 대상 그룹을 하나만 지정한 경우에는 트래픽의 100%가 하나의 대상 그룹으로 전송됩니다.

   필요한 경우 기본 작업에 다른 대상 그룹을 추가할 수 있습니다. 작업 추가를 선택한 다음 대상 그룹을 선택하고 가중치를 지정합니다.
8. (선택 사항) 다른 규칙을 추가하려면 규칙 추가를 선택한 다음, 규칙의 이름, 우선 순위, 조건 및 작업을 입력합니다.

   각 규칙에 1에서 100 사이의 우선 순위 번호를 부여할 수 있습니다. 리스너는 우선 순위가 동일한 규칙을 여러 개 자질 수 없습니다. 규칙은 가장 낮은 값에서 가장 높은 값에 이르기까지 우선 순위에 따라 평가됩니다. 기본 규칙은 마지막에 평가됩니다. 자세한 내용은 리스너 규칙 단원을 참조하십시오.

9. (선택 사항) 태그를 추가하려면 리스너 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.
10. HTTPS 리스너 인증서 설정에서, 서비스를 생성할 때 사용자 지정 도메인 이름을 지정하지 않은 경우 VPC Lattice는 자동으로 TLS 인증서를 생성하여 리스너를 통한 트래픽 흐름을 보호합니다.

    사용자 지정 도메인 이름으로 서비스를 생성했지만 일치하는 인증서를 지정하지 않은 경우 이제 사용자 지정 SSL/TLS 인증서에서 인증서를 선택하여 서비스를 생성할 수 있습니다. 그렇지 않으면 서비스를 생성할 때 지정한 인증서가 이미 선택되어 있습니다.
11. 구성을 검토하고 추가를 선택합니다.

를 사용하여 HTTPS 리스너를 추가하려면 AWS CLI

기본 규칙으로 리스너를 생성하려면 create-listener 명령을, 추가 리스너 규칙을 생성하려면 create-rule 명령을 사용합니다.

VPC Lattice 서비스를 위한 TLS 리스너

리스너는 연결 요청을 확인하는 프로세스입니다. VPC Lattice 서비스를 생성할 때 리스너를 정의할 수 있습니다. 언제라도 서비스에 리스너를 추가할 수 있습니다.

VPC Lattice가 암호화된 트래픽을 복호화하지 않고 애플리케이션으로 전달하도록 TLS 리스너를 생성할 수 있습니다.

VPC Lattice가 암호화된 트래픽을 복호화하고 암호화되지 않은 트래픽을 애플리케이션으로 전송하도록 하려면 대신 HTTPS 리스너를 생성합니다. 자세한 내용은 HTTPS 리스너 단원을 참조하십시오.

고려 사항

TLS 리스너에는 다음 고려 사항이 적용됩니다.

- VPC Lattice 서비스에는 사용자 지정 도메인 이름이 있어야 합니다. 서비스 사용자 지정 도메인 이름은 서비스 이름 표시(SNI) 일치로 사용됩니다. 서비스를 생성할 때 인증서를 지정한 경우 해당 인증서는 사용되지 않습니다.
- TLS 리스너에 허용되는 유일한 규칙은 기본 규칙입니다.
- TLS 리스너의 기본 작업은 TCP 대상 그룹에 대한 전달 작업이어야 합니다.
- 기본적으로 TCP 대상 그룹에 대해서는 상태 확인이 비활성화됩니다. TCP 대상 그룹에 대한 상태 확인을 활성화하는 경우 프로토콜 및 프로토콜 버전을 지정해야 합니다.

- TLS 리스너는 client-hello 메시지의 SNI 필드를 사용하여 요청을 라우팅합니다. 일치하는 조건이 client-hello와 정확히 일치하는 경우 대상에 와일드카드 및 SAN 인증서를 사용할 수 있습니다.
- 모든 트래픽은 클라이언트에서 대상으로 암호화된 상태로 유지되므로 VPC Lattice는 HTTP 헤더를 읽을 수 없으며 HTTP 헤더를 삽입하거나 제거할 수 없습니다. 따라서 TLS 리스너의 경우 다음과 같은 제한 사항이 있습니다.
  - 연결 시간은 10분으로 제한됩니다.
  - 인증 정책은 익명 보안 주체로 제한됩니다.
  - Lambda 대상은 지원되지 않습니다.
- Websocket 연결은 TLS 리스너를 사용하여 , VPC Lattice 서비스에 연결할 수 있습니다. 다음과 같은 제한 사항이 있습니다.
  - 연결 시간은 10분으로 제한됩니다.
  - 인증 정책은 익명 보안 주체로 제한됩니다.
  - Lambda 대상은 지원되지 않습니다.
- 암호화된 클라이언트 Hello(ECH)는 지원되지 않습니다.
- 암호화된 서버 이름 표시(ESNI)는 지원되지 않습니다.

TLS 리스너 추가

리스너에서 클라이언트에서 서비스로의 연결을 위한 프로토콜 및 포트 번호와 기본 리스너 규칙에 대한 대상 그룹을 구성합니다. 자세한 내용은 리스너 구성 단원을 참조하십시오.

콘솔을 사용하여 TLS 리스너를 추가하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 라우팅 탭에서 리스너 추가를 선택합니다.
5. 리스너 이름에서 사용자 지정 리스너 이름을 제공하거나 리스너의 프로토콜과 포트를 리스너 이름으로 사용할 수 있습니다. 지정할 사용자 지정 이름은 최대 63자까지 가능하며 계정의 모든 서비스마다 고유해야 합니다. 유효한 문자는 a~z, 0~9, 하이픈(-)입니다. 하이픈은 첫 문자 또는 마지막 문자로 사용할 수도 없고 다른 하이픈 바로 뒤에 사용할 수도 없습니다. 리스너를 생성한 후에는 리스너의 이름을 변경할 수 없습니다.
6. 프로토콜에서 TCP를 선택합니다. 포트에 포트 번호를 입력합니다.

7. 대상 그룹에 전달에서 TCP 프로토콜을 사용하여 트래픽을 수신하는 VPC Lattice 대상 그룹을 선택하고이 대상 그룹에 할당할 가중치를 선택합니다. 선택적으로 다른 대상 그룹을 추가할 수 있습니다. 대상 그룹 추가를 선택한 다음 대상 그룹을 선택하고 가중치를 입력합니다.
8. (선택 사항) 태그를 추가하려면 리스너 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.
9. 구성을 검토하고 추가를 선택합니다.

를 사용하여 TLS 리스너를 추가하려면 AWS CLI

create-listener 명령을 사용하여 기본 규칙으로 리스너를 생성합니다. TLS_PASSTHROUGH 프로토콜을 지정합니다.

VPC Lattice 서비스를 위한 리스너 규칙

각 리스너는 기본 규칙이 있으며 추가 규칙은 정의할 수 있습니다. 각 규칙은 우선 순위, 하나 이상의 작업, 하나 이상의 조건으로 구성됩니다. 언제든 규칙을 추가하거나 편집할 수 있습니다.

내용

기본 규칙

리스너를 생성할 때 기본 규칙에 대한 작업을 정의합니다. 기본 규칙은 조건을 가질 수 없습니다. 리스너의 규칙에 대한 조건이 충족되지 않으면 기본 규칙에 대해 작업이 수행됩니다.

규칙 우선 순위

각 규칙마다 우선 순위가 있습니다. 규칙은 가장 낮은 값에서 가장 높은 값에 이르기까지 우선 순위에 따라 평가됩니다. 기본 규칙은 마지막에 평가됩니다. 기본이 아닌 규칙의 우선 순위는 언제든지 변경이 가능합니다. 기본 규칙의 우선 순위는 변경할 수 없습니다.

규칙 작업

VPC Lattice 서비스의 리스너는 전달 작업과 고정 응답 작업을 지원합니다.

전달 작업

`forward` 작업을 사용하여 하나 이상의 VPC Lattice 대상 그룹에 요청을 라우팅할 수 있습니다. `forward` 작업에 대해 여러 대상 그룹을 지정하는 경우 각 대상 그룹에 대해 가중치를 지정해야 합니다. 각 대상 그룹 가중치는 0과 999 사이의 값입니다. 가중 대상 그룹이 있는 리스너 규칙과 일치하는 요청은 가중치를 기준으로 이러한 대상 그룹에 배포됩니다. 예를 들어, 각각 가중치가 10인 두 개의 대상 그룹을 지정하면 각 대상 그룹은 요청을 절반씩 받습니다. 가중치가 10인 대상 그룹과 가중치가 20인 대상 그룹 두 개를 지정하면 가중치가 20인 대상 그룹이 다른 대상 그룹보다 두 배 많은 요청을 받습니다.

고정 응답 작업

`fixed-response` 작업을 사용하여 클라이언트 요청을 삭제하고 사용자 지정 HTTP 응답을 반환할 수 있습니다. 이 작업을 사용하여 404 또는 500 응답 코드를 반환할 수 있습니다.

Example에 대한 고정 응답 작업의 예 AWS CLI

규칙을 만들거나 업데이트할 때 작업을 지정할 수 있습니다. 다음 작업은 지정된 상태 코드가 있는 고정 응답을 보냅니다.

```
"action": {
    "fixedResponse": {
        "statusCode": 404
},
```

규칙 조건

각 규칙 조건에는 유형과 구성 정보가 있습니다. 규칙에 대한 조건이 충족되면 작업이 수행됩니다.

규칙에 대해 지원되는 일치 기준은 다음과 같습니다.

헤더 일치

라우팅은 각 요청의 HTTP 헤더를 기반으로 합니다. HTTP 헤더 조건을 사용하여 요청의 HTTP 헤더를 기반으로 요청을 라우팅하는 규칙을 구성할 수 있습니다. 표준 또는 사용자 지정 HTTP 헤더 필드의 이름을 지정할 수 있습니다. 헤더 이름과 일치 평가는 대소문자를 구분하지 않습니다. 대소

문자 구분을 켜서 이 설정을 변경할 수 있습니다. 와일드카드 문자는 헤더 이름에서는 지원되지 않습니다. 헤더 일치에는 접두사 일치, 정확한 일치, 포함 일치가 지원됩니다.

메서드 일치

라우팅은 각 요청의 HTTP 요청 메서드를 기반으로 합니다.

HTTP 요청 메서드 조건을 사용하여 요청의 HTTP 요청 메서드를 기반으로 요청을 라우팅하는 규칙을 구성할 수 있습니다. 표준 또는 사용자 지정 HTTP 메서드를 지정할 수 있습니다. 메서드는 대소문자를 구분합니다. 메서드 이름이 정확히 일치해야 합니다. 와일드카드 문자는 지원되지 않습니다.

경로 일치

라우팅은 요청 URL의 경로 패턴 일치를 기반으로 합니다.

경로 조건을 사용하여 요청의 URL을 기반으로 요청을 라우팅하는 규칙을 정의할 수 있습니다. 와일드카드 문자는 지원되지 않습니다. 경로의 접두사 일치와 정확한 일치가 지원됩니다.

규칙 추가

언제든 리스너 규칙을 추가할 수 있습니다.

콘솔을 사용하여 리스너 규칙을 추가하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 라우팅 탭에서 리스너 편집을 선택합니다.
5. 리스너 규칙을 확장하고 규칙 추가를 선택합니다.
6. 규칙 이름에 규칙 이름을 입력합니다.
7. 우선 순위에 1에서 100 사이의 우선 순위를 입력합니다. 규칙은 가장 낮은 값에서 가장 높은 값에 이르기까지 우선 순위에 따라 평가됩니다. 기본 규칙은 마지막에 평가됩니다.
8. 조건에서 경로 일치 조건의 경로 패턴을 입력합니다. 각 문자열의 최대 크기는 200자입니다. 이 비교는 대소문자를 구분하지 않습니다. 와일드카드 문자는 지원되지 않습니다.

   헤더 일치 또는 메서드 일치 규칙 조건을 추가하려면 AWS CLI 또는 AWS SDK를 사용합니다.
9. 작업에서 VPC Lattice 대상 그룹을 선택합니다.
10. 변경 사항 저장을 선택합니다.

를 사용하여 규칙을 추가하려면 AWS CLI

create-rule 명령을 사용합니다.

규칙 업데이트

언제든 리스너 규칙을 업데이트할 수 있습니다. 대상 그룹의 우선 순위, 조건, 대상 그룹, 가중치 등을 수정할 수 있습니다. 규칙 이름은 수정할 수 없습니다.

콘솔을 사용하여 리스너 규칙을 업데이트하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 라우팅 탭에서 리스너 편집을 선택합니다.
5. 필요에 따라 규칙 우선 순위, 조건 및 작업을 수정합니다.
6. 업데이트를 검토하고 변경 사항 저장을 선택합니다.

를 사용하여 규칙을 업데이트하려면 AWS CLI

update-rule 명령을 사용합니다.

규칙 삭제

리스너에 대한 기본이 아닌 규칙은 언제든지 삭제할 수 있습니다. 리스너에 대한 기본 규칙은 삭제할 수 없습니다. 리스너를 삭제하면 모든 리스너 규칙이 삭제됩니다.

콘솔을 사용하여 리스너 규칙을 삭제하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 라우팅 탭에서 리스너 편집을 선택합니다.
5. 규칙을 찾고 제거를 선택합니다.
6. 변경 사항 저장을 선택합니다.

를 사용하여 규칙을 삭제하려면 AWS CLI

delete-rule 명령을 사용합니다.

VPC Lattice 서비스의 리스너 삭제

언제든 리스너를 삭제할 수 있습니다. 리스너를 삭제하면 모든 리스너 규칙이 자동으로 삭제됩니다.

콘솔을 사용하여 리스너를 삭제하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스를 선택합니다.
3. 서비스 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 라우팅 탭에서 리스너 삭제를 선택합니다.
5. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 삭제를 선택합니다.

를 사용하여 리스너를 삭제하려면 AWS CLI

delete-listener 명령을 사용하세요.

Amazon VPC Lattice의 VPC 리소스

VPC 리소스를 조직의 다른 팀 또는 외부 독립 소프트웨어 공급업체(ISV) 파트너와 공유할 수 있습니다. VPC 리소스는 Amazon RDS 데이터베이스, 도메인 이름 또는 IP 주소와 같은 AWS네이티브 리소스일 수 있습니다. 리소스는 VPC 또는 온프레미스 네트워크에 있을 수 있으며 로드 밸런싱할 필요가 없습니다. AWS RAM 를 사용하여 리소스에 액세스할 수 있는 보안 주체를 지정합니다. 리소스에 액세스할 수 있는 리소스 게이트웨이를 생성합니다. 또한 공유하려는 리소스 또는 리소스 그룹을 나타내는 리소스 구성을 생성합니다.

리소스를 공유하는 보안 주체는 VPC 엔드포인트를 사용하여 이러한 리소스에 비공개로 액세스할 수 있습니다. 리소스 VPC 엔드포인트를 사용하여 하나의 리소스에 액세스하거나 VPC Lattice 서비스 네트워크의 여러 리소스를 풀링하고 서비스 네트워크 VPC 엔드포인트를 사용하여 서비스 네트워크에 액세스할 수 있습니다.

다음 섹션에서는 VPC Lattice에서 VPC 리소스를 생성하고 관리하는 방법을 설명합니다.

주제

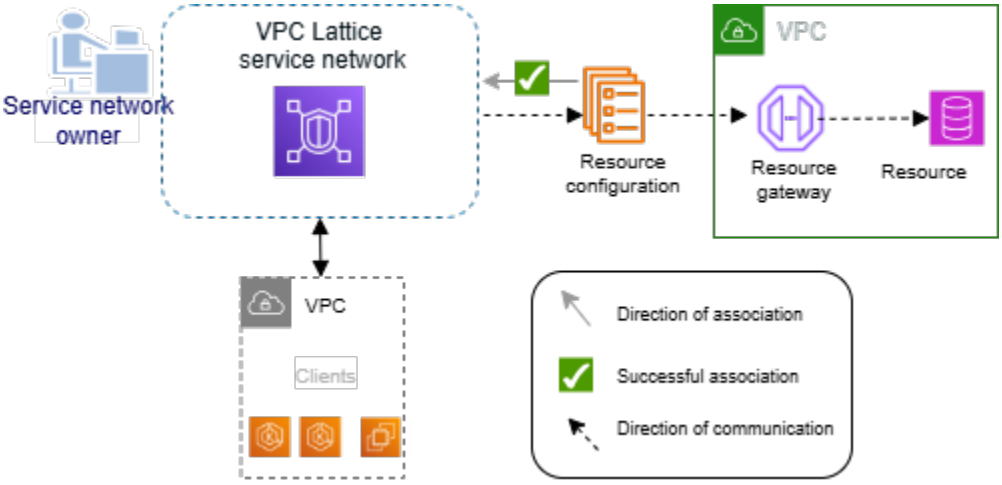
- VPC Lattice의 리소스 게이트웨이
- VPC 리소스에 대한 리소스 구성

VPC Lattice의 리소스 게이트웨이

리소스 게이트웨이는 리소스가 있는 VPC로 트래픽을 수신하는 지점입니다. 여러 가용 영역에 걸쳐 구성됩니다.

다른 VPC나 계정에서 VPC 내 리소스에 액세스할 계획이라면, 해당 VPC에는 반드시 리소스 게이트웨이가 있어야 합니다. 공유하는 모든 리소스는 리소스 게이트웨이와 연결됩니다. 다른 VPC 또는 계정의 클라이언트가 사용자 VPC에 있는 리소스에 액세스하면 해당 리소스는 트래픽이 그 VPC 내의 리소스 게이트웨이에서 로컬로 들어오는 것처럼 인식합니다. 트래픽의 소스 IP 주소는 가용 영역에 있는 리소스 게이트웨이의 IP 주소입니다. 각각 리소스가 여러 개인 여러 리소스 구성을 리소스 게이트웨이에 연결할 수 있습니다.

다음 다이어그램은 클라이언트가 리소스 게이트웨이를 통해 리소스에 액세스하는 방법을 보여줍니다.



내용

고려 사항

리소스 게이트웨이에는 다음과 같은 고려 사항이 적용됩니다.

- 리소스가 모든 가용 영역에서 액세스 가능하도록 하려면, 리소스 게이트웨이를 가능한 많은 가용 영역에 걸쳐 생성하는 것이 좋습니다.
- VPC 엔드포인트와 리소스 게이트웨이는 최소 1개 이상 가용 영역이 겹쳐야 합니다.
- 하나의 VPC에는 최대 100개의 리소스 게이트웨이를 생성할 수 있습니다. 자세한 내용은 VPC Lattice 할당량을 참조하세요.
- VPC Lattice는 리소스 게이트웨이에 새 ENIs 추가할 수 있습니다.
- 공유 VPC 서브넷이 있는 리소스 게이트웨이:
  - 리소스 게이트웨이는 VPC를 소유한 계정만 공유 VPC 서브넷에 배포할 수 있습니다.
  - 리소스 게이트웨이에 대한 리소스 구성은 리소스 게이트웨이를 소유한 계정에서만 생성할 수 있습니다.

보안 그룹

리소스 게이트웨이에 보안 그룹을 연결할 수 있습니다. 리소스 게이트웨이에 대한 보안 그룹 규칙은 리소스 게이트웨이에서 리소스로 향하는 아웃바운드 트래픽을 제어합니다.

데이터베이스 리소스로 향하는 리소스 게이트웨이 트래픽에 권장되는 아웃바운드 규칙

리소스 게이트웨이에서 리소스로 트래픽이 흐르도록 하려면, 리소스에서 허용하는 리스너 프로토콜과 포트 범위에 대한 아웃바운드 규칙을 생성해야 합니다.

Destination	프로토콜	포트 범위	설명
#### CIDR ##	TCP	3306	리소스 게이트웨이에서 데이터베이스로 향하는 트래픽을 허용합니다.

IP 주소 유형

리소스 게이트웨이에는 IPv4, IPv6 또는 듀얼 스택 주소가 있을 수 있습니다. 리소스 게이트웨이의 IP 주소 유형은 여기에 설명된 대로 리소스 게이트웨이의 서브넷과 리소스의 IP 주소 유형과 호환되어야 합니다.

- IPv4 - 리소스 게이트웨이 네트워크 인터페이스에 IPv4 주소를 할당합니다. 이 옵션은 선택한 모든 서브넷에 IPv4 주소 범위가 있고, 리소스에도 IPv4 주소가 있는 경우에만 지원됩니다. 이 옵션을 사용하면 리소스 게이트웨이 ENI당 IPv4 주소 수를 구성할 수 있습니다.
- IPv6 - 리소스 게이트웨이 네트워크 인터페이스에 IPv6 주소를 할당합니다. 이 옵션은 선택한 모든 서브넷이 IPv6 전용 서브넷이고, 리소스에도 IPv6 주소가 있는 경우에만 지원됩니다. 이 옵션을 사용하면 IPv6 주소가 자동으로 할당되므로 관리할 필요가 없습니다.
- 듀얼 스택 - 리소스 게이트웨이 네트워크 인터페이스에 IPv4 및 IPv6 주소를 모두 할당합니다. 이 옵션은 선택한 모든 서브넷에 IPv4와 IPv6 주소 범위가 모두 있고, 리소스에 IPv4 또는 IPv6 주소가 있는 경우에만 지원됩니다. 이 옵션을 사용하면 리소스 게이트웨이 ENI당 IPv4 주소 수를 구성할 수 있습니다.

리소스 게이트웨이의 IP 주소 유형은 클라이언트의 IP 주소 유형 또는 리소스에 액세스하는 VPC 엔드포인트와 독립적입니다.

ENI당 IPv4 주소

리소스 게이트웨이가 IPv4 또는 듀얼 스택 IP 주소 유형을 사용하는 경우, 리소스 게이트웨이의 각 ENI에 할당할 IPv4 주소 수를 구성할 수 있습니다. 리소스 게이트웨이를 생성할 때 1에서 62까지의 IPv4 주소 중에서 선택합니다. 한 번 설정한 IPv4 주소 수는 변경할 수 없습니다.

IPv4 주소는 네트워크 주소 변환에 사용되며, 리소스에 대한 동시 IPv4 연결의 최대 수를 결정합니다. 각 IPv4 주소는 대상 IP당 최대 55,000개의 동시 연결을 지원할 수 있습니다. 기본적으로 모든 리소스 게이트웨이에는 ENI당 16개의 IPv4 주소가 할당됩니다.

리소스 게이트웨이가 IPv6 주소 유형을 사용하는 경우, 리소스 게이트웨이는 자동으로 ENI당 /80 CIDR을 수신합니다. 이 값은 변경할 수 없습니다. 연결당 최대 전송 단위(MTU)는 8,500바이트입니다.

VPC Lattice에서 리소스 게이트웨이 생성

콘솔을 사용하여 리소스 게이트웨이를 생성합니다.

콘솔을 사용하여 리소스 게이트웨이 엔드포인트 생성하기

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 PrivateLink 및 Lattice에서 리소스 게이트웨이를 선택합니다.
3. 리소스 게이트웨이 생성을 선택합니다.
4. 리소스 게이트웨이 이름에 AWS 계정 내에서 고유한 이름을 입력합니다.
5. IP 주소 유형에서 리소스 게이트웨이트의 IP 주소 유형을 선택합니다.

    - IPv4 또는 듀얼 스택을 IP 주소 유형으로 선택한 경우, 리소스 게이트웨이의 각 ENI에 할당할 IPv4 주소 수를 입력할 수 있습니다.

      기본값은 ENI당 16개의 IPv4 주소입니다. 이는 백엔드 리소스와 연결을 형성하기에 적절한 IP 수입니다.
6. VPC에서 리소스 게이트웨이를 생성할 VPC 및 서브넷을 선택합니다.
7. 보안 그룹에서 최대 5개의 보안 그룹을 선택하여 VPC에서 서비스 네트워크로의 인바운드 트래픽을 제어합니다.
8. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
9. 리소스 게이트웨이 생성을 선택합니다.

를 사용하여 리소스 게이트웨이를 생성하려면 AWS CLI

create-resource-gateway 명령을 사용합니다.

VPC Lattice에서 리소스 게이트웨이 삭제

리소스 게이트웨이를 사용하여 리소스를 삭제합니다.

콘솔을 사용하여 리소스 게이트웨이 삭제하기

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 PrivateLink 및 Lattice에서 리소스 게이트웨이를 선택합니다.
3. 삭제하려는 리소스 게이트웨이의 확인란을 선택한 후 작업, 삭제를 차례로 선택합니다. 확인 메시지가 나타나면 **confirm**을 입력한 다음 삭제를 선택합니다.

를 사용하여 리소스 게이트웨이를 삭제하려면 AWS CLI

delete-resource-gateway 명령을 사용합니다.

VPC 리소스에 대한 리소스 구성

리소스 구성은 다른 VPC 및 계정의 클라이언트가 액세스할 수 있는 리소스 또는 리소스 그룹을 나타냅니다. 리소스 구성을 정의하면, 다른 VPC 및 계정의 클라이언트가 사용자 VPC의 리소스에 대한 안전한 프라이빗 단방향 네트워크 연결을 허용할 수 있습니다. 리소스 구성은 트래픽을 수신하는 리소스 게이트웨이와 연결됩니다. 다른 VPC에서 리소스에 액세스하려면 리소스 구성이 있어야 합니다.

내용

- 리소스 구성 유형
- 프로토콜
- 리소스 게이트웨이
- 리소스 공급자의 사용자 지정 도메인 이름
- 리소스 소비자의 사용자 지정 도메인 이름
- 서비스 네트워크 소유자의 사용자 지정 도메인 이름
- 리소스 정의
- 포트 범위
- 리소스 액세스
- 서비스 네트워크 유형과의 연결
- 서비스 네트워크 유형

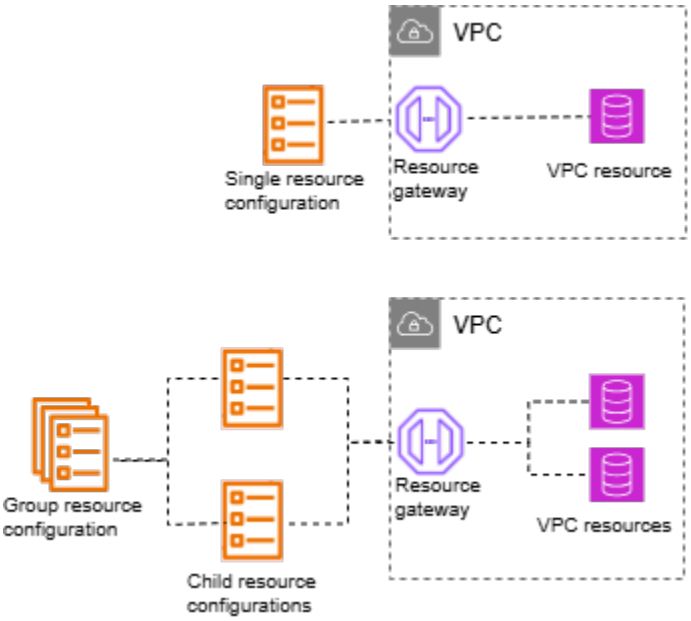
- 를 통해 리소스 구성 공유 AWS RAM
- 모니터링
- 도메인 생성 및 확인
- VPC Lattice에서 리소스 구성 생성
- VPC Lattice 리소스 구성을 위한 연결 관리

리소스 구성 유형

리소스 구성은 여러 유형이 있을 수 있습니다. 서로 다른 유형은 다양한 종류의 리소스를 표현하는 데 사용됩니다. 다음과 같은 유형이 있습니다.

- 단일 리소스 구성: IP 주소 또는 도메인 이름을 나타냅니다. 독립적으로 공유할 수 있습니다.
- 그룹 리소스 구성: 하위 리소스 구성의 모음입니다. DNS 및 IP 주소 엔드포인트 그룹을 나타내는 데 사용할 수 있습니다.
- 하위 리소스 구성: 그룹 리소스 구성의 멤버입니다. IP 주소 또는 도메인 이름을 나타냅니다. 독립적으로 공유할 수 없으며 그룹의 일부로만 공유할 수 있습니다. 그룹에서 추가 및 제거할 수 있습니다. 추가하면 해당 그룹에 액세스할 수 있는 모든 사람이 자동으로 액세스할 수 있게 됩니다.
- ARN 리소스 구성: AWS 서비스에 의해 프로비저닝되는 지원되는 리소스 유형을 나타냅니다. 모든 그룹-하위 관계는 자동으로 처리됩니다.

다음 이미지는 단일, 하위 및 그룹 리소스 구성을 보여줍니다.



프로토콜

리소스 구성을 생성할 때 리소스가 지원할 프로토콜을 정의할 수 있습니다. 현재는 TCP 프로토콜만 지원됩니다.

리소스 게이트웨이

리소스 구성은 리소스 게이트웨이와 연결됩니다. 리소스 게이트웨이는 해당 리소스가 위치한 VPC로 들어오는 진입 지점 역할을 하는 탄력적 네트워크 인터페이스(ENI)의 집합입니다. 여러 리소스 구성을 동일한 리소스 게이트웨이에 연결할 수 있습니다. 다른 VPCs 또는 계정의 클라이언트가 VPC의 리소스에 액세스하면 리소스는 해당 VPC의 리소스 게이트웨이 IP 주소에서 로컬로 들어오는 트래픽을 확인합니다.

리소스 공급자의 사용자 지정 도메인 이름

리소스 공급자는 리소스 소비자가 리소스 구성에 액세스하는 데 사용할 수 `example.com`있는와 같은 리소스 구성에 사용자 지정 도메인 이름을 연결할 수 있습니다. 사용자 지정 도메인 이름은 리소스 공급자가 소유 및 확인하거나 타사 또는 AWS 도메인일 수 있습니다. 리소스 공급자는 리소스 구성을 사용하여 캐시 클러스터 및 Kafka 클러스터, TLS 기반 애플리케이션 또는 기타 AWS 리소스를 공유할 수 있습니다.

리소스 구성 공급자에는 다음 고려 사항이 적용됩니다.

- 리소스 구성에는 사용자 지정 도메인이 하나만 있을 수 있습니다.
- 리소스 구성의 사용자 지정 도메인 이름은 변경할 수 없습니다.
- 사용자 지정 도메인 이름은 모든 리소스 구성 소비자에게 표시됩니다.
- VPC Lattice의 도메인 이름 확인 프로세스를 사용하여 사용자 지정 도메인 이름을 확인할 수 있습니다. 자세한 내용은 단원을 참조하십시오the section called “도메인 생성 및 확인”.
- 유형 그룹 및 하위 유형의 리소스 구성의 경우 먼저 그룹 리소스 구성에서 그룹 도메인을 지정해야 합니다. 이후 하위 리소스 구성에는 그룹 도메인의 하위 도메인인 사용자 지정 도메인이 있을 수 있습니다. 그룹에 그룹 도메인이 없는 경우 하위에 대한 사용자 지정 도메인 이름을 사용할 수 있지만 VPC Lattice는 리소스 소비자의 VPC에서 하위 도메인 이름에 대한 호스팅 영역을 프로비저닝하지 않습니다.

리소스 소비자의 사용자 지정 도메인 이름

리소스 소비자가 사용자 지정 도메인 이름이 있는 리소스 구성에 대한 연결을 활성화하면 VPC Lattice가 VPC에서 Route 53 프라이빗 호스팅 영역을 관리하도록 허용할 수 있습니다. 리소스 소비자는 VPC

Lattice가 프라이빗 호스팅 영역을 관리할 수 있도록 허용할 도메인에 대해 세분화된 옵션을 제공합니다.

리소스 소비자는 리소스 엔드포인트, 서비스 네트워크 엔드포인트 또는 서비스 네트워크 VPC 연결을 통해 리소스 구성에 대한 연결을 활성화할 때 `private-dns-enabled` 파라미터를 설정할 수 있습니다. `private-dns-enabled` 파라미터와 함께 소비자는 DNS 옵션을 사용하여 VPC Lattice가 프라이빗 호스팅 영역을 관리할 도메인을 지정할 수 있습니다. 소비자는 다음 프라이빗 DNS 기본 설정 중에서 선택할 수 있습니다.

`ALL_DOMAINS`

VPC Lattice는 모든 사용자 지정 도메인 이름에 프라이빗 호스팅 영역을 프로비저닝합니다.

`VERIFIED_DOMAINS_ONLY`

VPC Lattice는 공급자가 사용자 지정 도메인 이름을 확인한 경우에만 프라이빗 호스팅 영역을 프로비저닝합니다.

`VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS`

VPC Lattice는 확인된 모든 사용자 지정 도메인 이름 및 리소스 소비자가 지정하는 기타 도메인 이름에 대해 프라이빗 호스팅 영역을 프로비저닝합니다. 리소스 소비자는 `private DNS specified domains` 파라미터에 도메인 이름을 지정합니다.

`SPECIFIED_DOMAINS_ONLY`

VPC Lattice는 리소스 소비자가 지정한 도메인 이름에 대해 프라이빗 호스팅 영역을 프로비저닝합니다. 리소스 소비자는 `private DNS specified domains` 파라미터에 도메인 이름을 지정합니다.

프라이빗 DNS를 활성화하면 VPC Lattice는 리소스 구성과 연결된 사용자 지정 도메인 이름에 대한 프라이빗 호스팅 영역을 VPC에 생성합니다. 기본적으로 프라이빗 DNS 기본 설정은 로 설정됩니다`VERIFIED_DOMAINS_ONLY`. 즉, 리소스 공급자가 사용자 지정 도메인 이름을 확인한 경우에만 프라이빗 호스팅 영역이 생성됩니다. 프라이빗 DNS 기본 설정을 `ALL_DOMAINS` 또는 로 설정하면 `SPECIFIED_DOMAINS_ONLY` VPC Lattice는 사용자 지정 도메인 이름의 확인 상태에 관계없이 프라이빗 호스팅 영역을 생성합니다. 지정된 도메인에 대해 프라이빗 호스팅 영역이 생성되면 VPC에서 해당 도메인으로 가는 모든 트래픽이 VPC Lattice를 통해 라우팅됩니다. 이러한 사용자 지정 도메인 이름에 대한 트래픽이 VPC Lattice를 통과하도록 하려는 경우에만 `ALL_DOMAINS`VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS, 또는 `SPECIFIED_DOMAINS_ONLY` 기본 설정을 사용하는 것이 좋습니다.

리소스 소비자는 프라이빗 DNS 기본 설정을 로 설정하는 것이 좋습니다`VERIFIED_DOMAINS_ONLY`. 이를 통해 소비자는 VPC Lattice만 리소스 소비자 계정의 확인된 도메인에 프라이빗 호스팅 영역을 프로비저닝하도록 허용하여 보안 경계를 좁힐 수 있습니다.

프라이빗 DNS 지정 도메인에서 도메인을 선택하기 위해 리소스 소비자는와 같은 정규화된 도메인 이름을 입력`my.example.com`하거나와 같은 와일드카드를 사용할 수 있습니다`*.example.com`.

리소스 구성 소비자에게는 다음 고려 사항이 적용됩니다.

- 프라이빗 DNS 활성화 파라미터는 변경할 수 없습니다.
- VPC에서 프라이빗 호스팅을 생성하려면 서비스 네트워크 리소스 연결에서 프라이빗 DNS를 활성화해야 합니다. 리소스 구성의 경우 서비스 네트워크 리소스 연결의 프라이빗 DNS 활성화 상태는 서비스 네트워크 엔드포인트 또는 서비스 네트워크 VPC 연결의 프라이빗 DNS 활성화 상태를 재정의합니다.

서비스 네트워크 소유자의 사용자 지정 도메인 이름

서비스 네트워크 리소스 연결의 프라이빗 DNS 지원 속성은 서비스 네트워크 엔드포인트 및 서비스 네트워크 VPC 연결의 프라이빗 DNS 지원 속성을 재정의합니다.

서비스 네트워크 소유자가 서비스 네트워크 리소스 연결을 생성하고 프라이빗 DNS를 활성화하지 않는 경우 VPC Lattice는 서비스 네트워크 엔드포인트 또는 서비스 네트워크 VPCs 연결에서 프라이빗 DNS가 활성화되어 있더라도 서비스 네트워크가 연결된 VPC에서 해당 리소스 구성에 대한 프라이빗 호스팅 영역을 프로비저닝하지 않습니다.

ARN 유형의 리소스 구성의 경우 프라이빗 DNS 플래그는 true이고 변경할 수 없습니다.

리소스 정의

리소스 구성에서 리소스를 다음 방법 중 하나로 식별할 수 있습니다.

- Amazon 리소스 이름(ARN)으로: AWS 서비스에서 프로비저닝하는 지원되는 리소스 유형은 해당 ARN으로 식별할 수 있습니다. 지원되는 리소스는 Amazon RDS 데이터베이스뿐입니다. 공개적으로 액세스할 수 있는 클러스터에 대한 리소스 구성은 생성할 수 없습니다.
- 도메인 이름 대상별: 공개적으로 확인할 수 있는 모든 도메인 이름을 사용할 수 있습니다. 도메인 이름이 VPC 외부의 IP를 가리키는 경우, VPC 내에 NAT 게이트웨이가 있어야 합니다.
- IP 주소로 식별: IPv4의 경우 10.0.0.0/8, 100.64.0.0/10, 172.16.0.0/12, 192.168.0.0/16 범위 내의 프라이빗 IP를 지정합니다. IPv6의 경우 VPC 내의 IP를 지정합니다. 퍼블릭 IP는 지원되지 않습니다.

포트 범위

리소스 구성을 생성할 때 요청을 수락할 포트를 정의할 수 있습니다. 다른 포트에 대한 클라이언트 액세스는 허용되지 않습니다.

리소스 액세스

소비자는 VPC 엔드포인트를 사용하거나 서비스 네트워크를 통해 리소스 구성을 직접 액세스할 수 있습니다. 소비자로서, 자신의 VPC에서 동일 계정 내 리소스 구성이나 다른 계정에서 AWS RAM을 통해 공유된 리소스 구성에 대한 액세스를 활성화할 수 있습니다.

- 리소스 구성에 직접 액세스

  AWS PrivateLink VPC에서 리소스 유형의 VPC 엔드포인트(리소스 엔드포인트)를 생성하여 VPC에서 리소스 구성에 비공개로 액세스할 수 있습니다. 리소스 엔드포인트를 생성하는 방법에 대한 자세한 내용은 AWS PrivateLink사용 설명서의 VPC 리소스 액세스를 참조하세요.
- 서비스 네트워크를 통한 리소스 구성 액세스

  리소스 구성을 서비스 네트워크에 연결하고 VPC를 서비스 네트워크에 연결할 수 있습니다. 연결을 통해 또는 서비스 네트워크 VPC 엔드포인트를 사용하여 VPC를 AWS PrivateLink 서비스 네트워크에 연결할 수 있습니다.

  서비스 네트워크 연결에 대한 자세한 내용은 VPC Lattice 서비스 네트워크 연결 관리를 참조하세요.

  서비스 네트워크 VPC 엔드포인트에 대한 자세한 내용은 AWS PrivateLink 사용 설명서의 서비스 네트워크 액세스를 참조하세요.

VPC에서 프라이빗 DNS가 활성화된 경우, 동일한 리소스 구성에 대해 리소스 엔드포인트와 서비스 네트워크 엔드포인트를 동시에 생성할 수 없습니다.

서비스 네트워크 유형과의 연결

Account-B와 같은 소비자 계정과 리소스 구성을 공유하는 경우 Account AWS RAM-B는 리소스 VPC 엔드포인트 또는 서비스 네트워크를 통해 리소스 구성에 직접 액세스할 수 있습니다.

서비스 네트워크를 통해 리소스 구성에 액세스하려면, Account-B가 해당 리소스 구성을 서비스 네트워크에 연결해야 합니다. 서비스 네트워크는 계정 간에 공유할 수 있습니다. 따라서 Account-B는 (리소스 구성이 연결된) 자신의 서비스 네트워크를 Account-C와 공유하여 Account-C에서도 해당 리소스에 액세스할 수 있습니다.

이러한 전이적 공유를 방지하려면, 리소스 구성을 계정 간 공유가 가능한 서비스 네트워크에 추가할 수 없도록 지정할 수 있습니다. 이 설정을 하면 Account-B는 해당 리소스를 앞으로 다른 계정과 공유되거나 공유 가능한 서비스 네트워크에 추가할 수 없습니다.

서비스 네트워크 유형

Account-B와 같은 다른 계정과 리소스 구성을 공유할 때 Account AWS RAM-B는 다음 세 가지 방법 중 하나로 리소스 구성에 지정된 리소스에 액세스할 수 있습니다.

- 리소스 유형의 VPC 엔드포인트 사용(리소스 VPC 엔드포인트).
- 서비스 네트워크 유형의 VPC 엔드포인트 사용(서비스 네트워크 VPC 엔드포인트).
- 서비스 네트워크 VPC 연결 사용.

  서비스 네트워크 연결을 사용하는 경우 각 리소스에는 AWS 소유되고 라우팅할 수 없는 129.224.0.0/17 블록의 서브넷당 IP가 할당됩니다. 이는 VPC Lattice가 트래픽을 VPC Lattice 네트워크를 통해 서비스로 라우팅할 때 사용하는 관리형 접두사 목록과 별도로 적용됩니다. 이 두 IP 모두 VPC 라우팅 테이블에 업데이트됩니다.

서비스 네트워크 VPC 엔드포인트 및 서비스 네트워크 VPC 연결의 경우 리소스 구성을 Account-B의 서비스 네트워크와 연결해야 합니다. 서비스 네트워크는 계정 간에 공유할 수 있습니다. 따라서 Account-B는 (리소스 구성을 포함한) 자신의 서비스 네트워크를 Account-C와 공유하여 Account-C에서도 리소스에 액세스할 수 있습니다. 이러한 전이적 공유를 방지하려면, 리소스 구성이 계정 간 공유 가능한 서비스 네트워크에 추가되지 않도록 지정할 수 있습니다. 이 설정을 하면 Account-B는 해당 리소스를 공유되거나 공유 가능한 서비스 네트워크에 추가할 수 없습니다.

를 통해 리소스 구성 공유 AWS RAM

리소스 구성은와 통합됩니다 AWS Resource Access Manager. AWS RAM을 통해 리소스 구성을 다른 계정과 공유할 수 있습니다. 리소스 구성을 AWS 계정과 공유하면 해당 계정의 클라이언트가 리소스에 비공개로 액세스할 수 있습니다. 리소스 구성은 AWS RAM의 리소스 공유를 사용하여 공유할 수 있습니다.

AWS RAM 콘솔을 사용하여 추가된 리소스 공유, 액세스할 수 있는 공유 리소스, 리소스를 공유한 AWS 계정을 볼 수 있습니다. 자세한 내용은 AWS RAM 사용 설명서의 공유받은 리소스를 참조하세요.

리소스 구성과 동일한 계정의 다른 VPC에서 리소스에 액세스하려면 리소스 구성을 공유할 필요가 없습니다 AWS RAM.

모니터링

리소스 구성에서 모니터링 로그를 활성화할 수 있습니다. 로그를 전송할 대상을 선택할 수 있습니다.

도메인 생성 및 확인

도메인 이름 확인은 지정된 도메인의 소유권을 증명할 수 있는 엔터티입니다. 리소스 공급자는 도메인과 하위 도메인을 리소스 구성의 사용자 지정 도메인 이름으로 사용할 수 있습니다. 리소스 소비자는 리소스 구성을 설명할 때 사용자 지정 도메인 이름의 확인 상태를 볼 수 있습니다.

도메인 확인 시작

VPC Lattice를 사용하여 도메인 이름 확인을 시작한 다음 DNS 영역을 사용하여 프로세스를 완료합니다.

AWS Management Console

도메인 이름 확인을 시작하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 PrivateLink 및 Lattice에서 도메인 확인을 선택합니다.
3. 도메인 확인 시작을 선택합니다.
4. 도메인 이름에 소유한 도메인 이름을 입력합니다.
5. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
6. 도메인 이름 확인 시작을 선택합니다.

도메인 이름 확인이 성공적으로 시작되면 VPC Lattice는 `Id` 및를 반환합니다`txtMethodConfig`. `txtMethodConfig`를 사용하여 도메인 이름 확인을 완료합니다.

AWS CLI

다음 `start-domain-verification` 명령은 도메인 이름 확인을 시작합니다.

```
aws vpc-lattice start-domain-verification \
  --domain-name example.com
```

출력은 다음과 같습니다.

```
{
    "id": "dv-aaaa0000000111111",
    "arn": "arn:aws:vpc-lattice:us-west-2:111122223333:domainverification/dv-
aaaa0000000111111",
    "domainName": "example.com",
    "status": "PENDING",
    "txtMethodConfig": {
        "value": "vpc-lattice:1111aaaaaaa",
        "name": "_11111aaaaaaaaa"
    }
}
```

VPC Lattice는 `Id` 및를 반환합니다`txtMethodConfig`. `txtMethodConfig`를 사용하여 도메인 이름 확인을 완료합니다. 이 예제에서 `txtMethodConfig`는 다음과 같습니다.

```
txtMethodConfig": {
        "value": "vpc-lattice:1111aaaaaaa",
        "name": "_11111aaaaaaaaa"
    }
```

도메인 이름 확인 완료

도메인 이름 확인을 완료하려면 DNS 영역에 TXT 레코드를 추가합니다. Route 53을 사용하는 경우 도메인 이름의 호스팅 영역을 사용합니다. 도메인 이름을 확인하면 모든 하위 도메인도 확인됩니다. 예를 들어 `example.com`를 확인하는 경우 추가 확인을 수행하지 `beta.example.com` 않고 리소스 구성을 `alpha.example.com` 및와 연결할 수 있습니다.

를 사용하여 TXT 레코드를 생성하려면 Amazon Route 53 콘솔을 사용하여 레코드 생성을 AWS Management Console참조하세요.

Route 53 AWS CLI 용를 사용하여 TXT 레코드를 생성하려면

1. 다음 예제 `TXT-record.json` 파일과 함께 change-resource-record-sets 명령을 사용합니다.

```
{
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "_11111aaaaaaaaa",
```

```
        "Type": "TXT",
        "ResourceRecords": [
          {
           "value": "vpc-lattice:1111aaaaaaa"
          }
        ]
      }
    }
  ]
}
```

2. 다음 AWS CLI 명령을 사용하여 이전 단계의 TXT 레코드를 Route 53 호스팅 영역에 추가합니다.

```
aws route53 change-resource-record-sets \
  --hosted-zone-id ABCD123456 \
  --change-batch file://path/to/your/TXT-record.json
```

   를 계정에 있는 호스팅 영역의 Route 53 호스팅 영역 IDhosted-zone-id로 바꿉니다. change-batch 파라미터 값은 폴더(path/to/your)의 JSON 파일(TXT-record.json)을 가리킵니다.

도메인 이름의 확인 상태를 확인하려면 VPC Lattice 콘솔 또는 get-domain-verification 명령을 사용할 수 있습니다.

도메인 이름을 확인하면 삭제할 때까지 확인된 상태로 유지됩니다. DNS 영역에서 TXT 레코드를 삭제하면 VPC Lattice가를 삭제verification-id하므로 도메인 이름을 다시 확인해야 합니다. DNS 영역에서 TXT 레코드를 삭제하면 VPC Lattice는 도메인 이름 확인 상태를 로 설정합니다UNVERIFIED. 이는 기존 리소스 엔드포인트, 서비스 네트워크 엔드포인트 또는 리소스 구성에 대한 서비스 네트워크 VPC 연결에는 영향을 주지 않습니다. 도메인 이름을 다시 확인하려면 도메인 이름 확인 프로세스를 다시 시작합니다.

VPC Lattice에서 리소스 구성 생성

리소스 구성을 생성합니다.

AWS Management Console

콘솔을 사용하여 리소스 구성 생성하기

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 PrivateLink 및 Lattice에서 리소스 구성을 선택합니다.

3. 리소스 구성 생성을 선택합니다.
4. AWS 계정 내에서 고유한 이름을 입력합니다. 리소스 구성이 생성된 후에는 이 이름을 변경할 수 없습니다.
5. 구성 유형에서 단일 또는 하위 리소스용 리소스 또는 하위 리소스 그룹용 리소스 그룹을 선택합니다.
6. 이전에 생성한 리소스 게이트웨이를 선택하거나 새로 생성합니다.
7. (선택 사항) 사용자 지정 도메인 이름을 입력하려면 다음 중 하나를 수행합니다.
   - 단일 유형의 리소스 구성이 있는 경우 사용자 지정 도메인 이름을 입력할 수 있습니다. 리소스 소비자는이 도메인 이름을 사용하여 리소스 구성에 액세스할 수 있습니다.
   - 유형 그룹 및 하위의 리소스 구성이 있는 경우 먼저 그룹 리소스 구성에서 그룹 도메인을 지정해야 합니다. 다음으로 하위 리소스 구성에는 그룹 도메인의 하위 도메인인 사용자 지정 도메인이 있을 수 있습니다.
8. (선택 사항) 확인 ID를 입력합니다.

   도메인 이름을 확인하려는 경우 확인 ID를 제공합니다. 이를 통해 리소스 소비자는 사용자가 도메인 이름을 소유하고 있음을 알 수 있습니다.
9. 이 리소스 구성으로 나타낼 리소스의 식별자를 선택합니다.
10. 리소스를 공유할 포트 범위를 선택합니다.
11. 연결 설정에서 이 리소스 구성을 공유 가능한 서비스 네트워크와 연결할 수 있는지 여부를 지정합니다.
12. 리소스 구성 공유에서 이 리소스에 액세스할 수 있는 보안 주체를 식별하는 리소스 공유를 선택합니다.
13. (선택 사항) 모니터링에서 리소스 액세스 로그와 전송 대상을 활성화하여 리소스 구성을 오가는 요청과 응답을 모니터링할 수 있습니다.
14. (선택 사항) 태그를 추가하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
15. 리소스 구성 생성을 선택합니다.

AWS CLI

다음 create-resource-configuration 명령은 단일 리소스 구성을 생성하고 이를 사용자 지정 도메인 이름과 연결합니다`example.com`.

```
aws vpc-lattice create-resource-configuration \
```

```
    --name my-resource-config \
    --type SINGLE \
    --resource-gateway-identifier rgw-0bba03f3d56060135 \
    --resource-configuration-definition 'ipResource={ipAddress=10.0.14.85}' \
    --custom-domain-name example.com \
    --verification-id dv-aaaa0000000111111
```

다음 create-resource-configuration 명령은 그룹 리소스 구성을 생성하고 이를 사용자 지정 도메인 이름과 연결합니다`example.com`.

```
aws vpc-lattice-custom-dns create-resource-configuration \
  --name my-custom-dns-resource-config-group \
  --type GROUP \
  --resource-gateway-identifier rgw-0bba03f3d56060135 \
  --domain-verification-identifier dv-aaaa0000000111111
```

다음 create-resource-configuration 명령은 하위 리소스 구성을 생성하고 이를 사용자 지정 도메인 이름과 연결합니다`child.example.com`.

```
aws vpc-lattice-custom-dns create-resource-configuration \
  --name my-custom-dns-resource-config-child \
  --type CHILD \
  --resource-configuration-definition 'dnsResource={domainName=my-alb-123456789.us-
west-2.elb.amazonaws.com,ipAddressType=IPV4}' \
  --resource-configuration-group-identifier rcfg-07129f3acded87626 \
  --custom-domain-name child.example.com
```

VPC Lattice 리소스 구성을 위한 연결 관리

리소스 구성을 공유하는 소비자 계정 및 계정의 클라이언트는 리소스 유형의 VPC 엔드포인트를 직접 사용하거나 서비스-네트워크 유형의 VPC 엔드포인트를 통해 리소스 구성에 액세스할 수 있습니다. 따라서 리소스 구성에는 엔드포인트 연결과 서비스 네트워크 연결이 있습니다.

서비스 네트워크 리소스 연결 관리

서비스 네트워크 연결을 생성하거나 삭제할 수 있습니다.

> **Note**
>
> 서비스 네트워크와 리소스 구성 간의 연결을 생성하는 동안 액세스 거부 메시지가 표시되면 AWS RAM 정책 버전을 확인하고 버전 2인지 확인합니다. 자세한 내용은 AWS RAM 사용 설명서를 참조하세요.

콘솔을 사용하여 서비스-네트워크 연결 관리하기

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 PrivateLink 및 Lattice에서 리소스 구성을 선택합니다.
3. 리소스 구성의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 서비스 네트워크 연결 탭을 선택합니다.
5. 연결 생성을 선택합니다.
6. VPC Lattice 서비스 네트워크에서 서비스 네트워크를 선택합니다. 서비스 네트워크를 생성하려면 VPC Lattice 네트워크 생성을 선택합니다.
7. (선택 사항) 태그를 추가하려면 서비스 연결 태그를 확장하고 새 태그 추가를 선택하여 태그 키와 태그 값을 입력합니다.
8. (선택 사항)이 서비스 네트워크 리소스 연결에 프라이빗 DNS 이름을 활성화하려면 프라이빗 DNS 이름 활성화를 선택합니다. 자세한 내용은 the section called “서비스 네트워크 소유자의 사용자 지정 도메인 이름” 단원을 참조하십시오.
9. 변경 사항 저장을 선택합니다.
10. 연결을 삭제하려면 연결의 확인란을 선택한 다음 작업, 삭제를 선택합니다. 확인 메시지가 나타나면 **`confirm`**을 입력한 다음 삭제를 선택합니다.

를 사용하여 서비스 네트워크 연결을 생성하려면 AWS CLI

create-service-network-resource-association 명령을 사용합니다.

를 사용하여 서비스 네트워크 연결을 삭제하려면 AWS CLI

delete-service-network-resource-association 명령을 사용합니다.

리소스 VPC 엔드포인트 연결 관리

리소스 구성에 액세스할 수 있는 소비자 계정 또는 계정의 클라이언트는 리소스 VPC 엔드포인트를 사용하여 리소스 구성에 액세스할 수 있습니다. 리소스 구성에 사용자 지정 도메인 이름이 있는 경우 프

라이빗 DNS 활성화를 사용하여 VPC Lattice가 리소스 엔드포인트 또는 서비스 네트워크 엔드포인트에 프라이빗 호스팅 영역을 프로비저닝하도록 허용할 수 있습니다. 이를 통해 클라이언트는 도메인 이름을 직접 컬링하여 리소스 구성에 액세스할 수 있습니다. 자세한 내용은 the section called “리소스 소비자의 사용자 지정 도메인 이름” 단원을 참조하십시오.

AWS Management Console

1. 새 엔드포인트 연결을 생성하려면 왼쪽 탐색 창에서 PrivateLink 및 Lattice로 이동하여 엔드포인트를 선택합니다.
2. 엔드포인트 생성을 선택합니다.
3. VPC에 연결할 리소스 구성을 선택합니다.
4. VPC, 서브넷 및 보안 그룹을 선택합니다.
5. (선택 사항) 프라이빗 DNS를 켜고 DNS 옵션을 구성하려면 프라이빗 DNS 이름 활성화를 선택합니다.
6. (선택 사항) VPC 엔드포인트에 태그를 지정하려면 새 태그 추가를 선택하고 태그 키와 태그 값을 입력합니다.
7. 엔드포인트 생성을 선택합니다.

AWS CLI

다음 create-vpc-endpoint 명령은 프라이빗 DNS를 사용하는 VPC 엔드포인트를 생성합니다. 프라이빗 DNS 기본 설정은 로 설정VERIFIED_AND_SELECTED되고 선택한 도메인은 example.com 및 입니다example.org. VPC Lattice는 확인된 도메인 또는 example.com에 대해서만 프라이빗 호스팅 영역을 프로비저닝합니다example.org.

```
aws ec2 create-vpc-endpoint \
  --vpc-endpoint-type Resource \
  --vpc-id vpc-111122223333aabbc \
  --subnet-ids subnet-0011aabbcc2233445 \
  --resource-configuration-arn arn:aws:vpc-lattice:us-
west-2:111122223333:resourceconfiguration/rcfg-07129f3acded87625 \
  --private-dns-enabled \
  --private-dns-preferences VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS \
  --private-domains-set example.com, example.org
```

를 사용하여 VPC 엔드포인트 연결을 생성하려면 AWS CLI

create-vpc-endpoint 명령을 사용합니다.

를 사용하여 VPC 엔드포인트 연결을 삭제하려면 AWS CLI

delete-vpc-endpoint 명령을 사용합니다.

VPC Lattice 엔터티 공유

Amazon VPC Lattice는 AWS Resource Access Manager (AWS RAM)와 통합되어 서비스, 리소스 구성 및 서비스 네트워크를 공유할 수 있습니다. AWS RAM 는 일부 VPC Lattice 개체를 다른 AWS 계정 또는를 통해 공유할 수 있는 서비스입니다 AWS Organizations. 를 사용하면 리소스 AWS RAM공유를 생성하여 소유한 엔터티를 공유할 수 있습니다. 리소스 공유는 공유할 엔터티와 공유할 소비자를 지정합니다. 소비자에는 다음이 포함될 수 있습니다.

- 의 조직 AWS 계정 내부 또는 외부에 특정합니다 AWS Organizations.
- AWS Organizations내 조직 내부의 조직 단위
- AWS Organizations의 전체 조직.

에 대한 자세한 내용은 AWS RAM 사용 설명서를 AWS RAM참조하세요.

내용

VPC Lattice 엔터티 공유를 위한 사전 조건

- 개체를 공유하려면에서 해당 개체를 소유해야 합니다 AWS 계정. 즉, 계정에서 엔터티를 할당하거나 프로비저닝해야 합니다. 공유된 엔터티는 공유할 수 없습니다.
- 에서 조직 또는 조직 단위와 개체를 공유하려면 와의 공유를 활성화 AWS Organizations해야 합니다 AWS Organizations. 자세한 내용은 AWS RAM 사용 설명서의 AWS Organizations내 리소스 공유 활성화를 참조하세요.

VPC Lattice 엔터티 공유

개체를 공유하려면 먼저를 사용하여 리소스 공유를 생성합니다 AWS Resource Access Manager. 리소스 공유는 공유할 엔터티, 공유 대상 소비자, 보안 주체가 수행할 수 있는 작업을 지정합니다.

소유한 VPC Lattice 엔터티를 다른 엔터티와 공유 AWS 계정하면 해당 계정이 해당 엔터티를 계정의 엔터티와 연결할 수 있습니다. 공유 엔터티에 대한 연결을 생성하면 엔터티 소유자 계정과 연결을 생성한 계정에서 Amazon 리소스 이름(ARN)이 생성됩니다. 따라서 엔터티 소유자와 연결을 생성한 계정 모두 연결을 삭제할 수 있습니다.

의 조직에 속 AWS Organizations 해 있고 조직 내 공유가 활성화된 경우 조직의 소비자에게 공유 엔터티에 대한 액세스 권한이 자동으로 부여됩니다. 그렇지 않으면 소비자는 리소스 공유에 가입하라는 초대를 받고 초대를 수락한 후 공유 엔터티에 대한 액세스 권한이 부여됩니다.

고려 사항

- 서비스 네트워크, 서비스 및 리소스 구성이라는 세 가지 유형의 VPC Lattice 엔터티를 공유할 수 있습니다.
- VPC Lattice 엔터티를와 공유할 수 있습니다 AWS 계정.
- VPC Lattice 엔터티를 개별 IAM 사용자 및 역할과 공유할 수 없습니다.
- VPC Lattice는 서비스, 리소스 구성 및 서비스 네트워크에 대한 고객 관리형 권한을 지원합니다.

VPC Lattice 콘솔을 사용하여 소유한 개체를 공유하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스, 서비스 네트워크 또는 리소스 구성을 선택합니다.
3. 개체 이름을 선택하여 세부 정보 페이지를 연 다음 공유 탭에서 서비스 공유, 서비스 네트워크 공유 또는 리소스 구성 공유를 선택합니다.
4. AWS RAM 리소스 공유에서 리소스 공유를 선택합니다. 리소스 공유를 생성하려면 RAM 콘솔에서 리소스 공유 생성을 선택합니다.
5. 서비스 공유, 서비스 네트워크 공유 또는 리소스 구성 공유를 선택합니다.

AWS RAM 콘솔을 사용하여 소유한 엔터티를 공유하려면

AWS RAM 사용설명서의 리소스 공유 생성에 설명된 절차를 따릅니다.

를 사용하여 소유한 개체를 공유하려면 AWS CLI

associate-resource-share 명령을 사용합니다.

VPC Lattice 엔터티 공유 중지

소유한 VPC Lattice 엔터티 공유를 중지하려면 리소스 공유에서 제거해야 합니다. 기존 연결은 엔터티 공유를 중지한 후에도 유지됩니다. 이전에 공유된 엔터티에 대한 새 연결은 허용되지 않습니다. 엔터티 소유자 또는 연결 소유자가 연결을 삭제하면 두 계정에서 모두 삭제됩니다. 계정 소유자가 리소스 공유를 탈퇴하려는 경우 리소스 공유 소유자에게이 리소스가 공유된 계정 목록에서 계정을 제거하도록 요청해야 합니다.

VPC Lattice 콘솔을 사용하여 소유한 엔터티 공유를 중지하려면

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스, 서비스 네트워크 또는 리소스 구성을 선택합니다.
3. 개체의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. 공유 탭에서 리소스 공유의 확인란을 선택한 다음 제거를 선택합니다.

AWS RAM 콘솔을 사용하여 소유한 엔터티 공유를 중지하려면

AWS RAM 사용 설명서의 리소스 공유 업데이트를 참조하세요.

를 사용하여 소유한 엔터티 공유를 중지하려면 AWS CLI

disassociate-resource-share 명령을 사용합니다.

책임 및 권한

공유 VPC Lattice 엔터티를 사용할 때는 다음과 같은 책임과 권한이 적용됩니다.

개체 소유자

- 서비스 네트워크 소유자는 소비자가 만든 서비스를 수정할 수 없습니다.
- 서비스 네트워크 소유자는 소비자가 만든 서비스를 삭제할 수 없습니다.
- 서비스 네트워크 소유자는 서비스 네트워크에 대한 모든 서비스 연결을 설명할 수 있습니다.
- 서비스 네트워크 소유자는 누가 연결을 생성했는지에 상관없이 서비스 네트워크와 관련된 모든 서비스의 연결을 해제할 수 있습니다.
- 서비스 네트워크 소유자는 서비스 네트워크에 대한 모든 VPC 연결을 설명할 수 있습니다.
- 서비스 네트워크 소유자는 소비자가 서비스 네트워크에 연결한 모든 VPC의 연결을 해제할 수 있습니다.

- 서비스 네트워크 소유자는 서비스 네트워크에 대한 모든 리소스 구성 연결을 설명할 수 있습니다.
- 서비스 네트워크 소유자는 누가 연결을 생성했는지에 관계없이 서비스 네트워크와 연결된 모든 리소스 구성을 연결 해제할 수 있습니다.
- 서비스 네트워크 소유자는 서비스 네트워크에 대한 모든 엔드포인트 연결을 설명할 수 있습니다.
- 서비스 네트워크 소유자는 연결을 생성한 사람에 관계없이 서비스 네트워크와 연결된 엔드포인트의 연결을 해제할 수 있습니다.
- 서비스 소유자는 서비스와의 모든 서비스 네트워크 연결을 설명할 수 있습니다.
- 서비스 소유자는 서비스와 연결된 모든 서비스 네트워크에서 서비스의 연결을 해제할 수 있습니다.
- 리소스 구성 소유자는 리소스 구성과의 모든 네트워크 연결을 설명할 수 있습니다.
- 리소스 구성 소유자는 연결된 모든 서비스 네트워크에서 리소스 구성을 연결 해제할 수 있습니다.
- VPC 엔드포인트 소유자는 연결된 서비스 네트워크를 설명할 수 있습니다.
- VPC 엔드포인트 소유자는 서비스 네트워크에서 엔드포인트를 연결 해제할 수 있습니다.
- 연결을 생성한 계정만 서비스 네트워크와 VPC 간의 연결을 업데이트할 수 있습니다.

개체 소비자

- 소비자는 생성하지 않은 서비스 또는 리소스 구성을 삭제할 수 없습니다.
- 소비자는 서비스 네트워크와 연결된 서비스 또는 리소스 구성만 연결 해제할 수 있습니다.
- 소비자와 네트워크 소유자는 서비스 네트워크와 서비스 또는 리소스 구성 간의 모든 연결을 설명할 수 있습니다.
- 소비자는 자신이 소유하지 않은 리소스 구성의 서비스 정보 또는 리소스 구성 정보를 검색할 수 없습니다.
- 소비자는 공유 서비스 네트워크와의 모든 서비스 연결 및 리소스 구성 연결을 설명할 수 있습니다.
- 소비자는 서비스 또는 리소스 구성을 공유 서비스 네트워크에 연결할 수 있습니다.
- 소비자는 공유 서비스 네트워크와의 모든 VPC 연결을 볼 수 있습니다.
- 소비자는 VPC를 공유 서비스 네트워크와 연결할 수 있습니다.
- 소비자는 서비스 네트워크에 연결한 VPC만 연결 해제할 수 있습니다.
- 소비자는 서비스 네트워크 VPC 엔드포인트를 생성하여 VPC를 공유 서비스 네트워크에 연결할 수 있습니다.
- 소비자는 VPC를 공유 서비스 네트워크에 연결하기 위해 생성한 서비스 네트워크 VPC 엔드포인트만 삭제할 수 있습니다.

- 공유 서비스의 소비자는 서비스를 자신이 소유하지 않은 서비스 네트워크와 연결할 수 없습니다.
- 공유 서비스 네트워크의 소비자는 자신이 소유하지 않은 VPC 또는 서비스를 연결할 수 없습니다.
- 공유 리소스 구성의 소비자는 자신이 소유하지 않은 서비스 네트워크와 리소스 구성을 연결할 수 없습니다.
- 공유 서비스 네트워크의 소비자는 자신이 소유하지 않은 VPC, 서비스 또는 리소스 구성을 연결할 수 없습니다.
- 소비자는 자신과 공유되는 서비스, 서비스 네트워크 또는 리소스 구성을 설명할 수 있습니다.
- 두 개체가 모두 공유되는 경우 소비자는 두 개체를 연결할 수 없습니다.

교차 계정 이벤트

엔터티 소유자와 소비자가 공유 엔터티에 대한 작업을 수행할 때 해당 작업은에서 교차 계정 이벤트로 기록됩니다 AWS CloudTrail.

`CreateServiceNetworkResourceAssociationBySharee`

엔터티 소비자가 공유 엔터티와 CreateServiceNetworkResourceAssociation을 호출할 때 엔터티 소유자에게 전송됩니다. 호출자가 리소스 구성을 소유한 경우 이벤트는 서비스 네트워크의 소유자에게 전송됩니다. 호출자가 서비스 네트워크를 소유한 경우 이벤트는 리소스 구성의 소유자에게 전송됩니다.

`CreateServiceNetworkServiceAssociationBySharee`

엔터티 소비자가 공유 엔터티와 CreateServiceNetworkServiceAssociation을 호출할 때 엔터티 소유자에게 전송됩니다. 호출자가 서비스를 소유한 경우 이벤트는 서비스 네트워크 소유자에게 전송됩니다. 호출자가 서비스 네트워크를 소유한 경우 이벤트는 서비스 소유자에게 전송됩니다.

`CreateServiceNetworkVpcAssociationBySharee`

엔터티 소비자가 공유 서비스 네트워크를 사용하여 CreateServiceNetworkVpcAssociation을 호출할 때 엔터티 소유자에게 전송됩니다.

`DeleteServiceNetworkResourceAssociationByOwner`

엔터티 소유자가 공유 엔터티와 DeleteServiceNetworkResourceAssociation을 호출할 때 연결 소유자에게 전송됩니다. 호출자가 리소스 구성을 소유한 경우 이벤트는 서비스 네트워크 연결의 소유자에게 전송됩니다. 호출자가 서비스 네트워크를 소유한 경우 이벤트는 리소스 연결 소유자에게 전송됩니다.

`DeleteServiceNetworkResourceAssociationBySharee`

엔터티 소비자가 공유 엔터티와 DeleteServiceNetworkResourceAssociation을 호출할 때 엔터티 소유자에게 전송됩니다. 호출자가 리소스 구성을 소유한 경우 이벤트는 서비스 네트워크의 소유자에게 전송됩니다. 호출자가 서비스 네트워크를 소유한 경우 이벤트는 리소스 구성의 소유자에게 전송됩니다.

`DeleteServiceNetworkServiceAssociationByOwner`

엔터티 소유자가 공유 엔터티와 DeleteServiceNetworkServiceAssociation을 호출할 때 연결 소유자에게 전송됩니다. 호출자가 서비스를 소유한 경우 이벤트는 서비스 네트워크 연결 소유자에게 전송됩니다. 호출자가 서비스 네트워크를 소유한 경우 이벤트는 서비스 연결 소유자에게 전송됩니다.

`DeleteServiceNetworkServiceAssociationBySharee`

엔터티 소비자가 공유 엔터티와 DeleteServiceNetworkServiceAssociation을 호출할 때 엔터티 소유자에게 전송됩니다. 호출자가 서비스를 소유한 경우 이벤트는 서비스 네트워크 소유자에게 전송됩니다. 호출자가 서비스 네트워크를 소유한 경우 이벤트는 서비스 소유자에게 전송됩니다.

`DeleteServiceNetworkVpcAssociationByOwner`

엔터티 소유자가 공유 서비스 네트워크와의 DeleteServiceNetworkVpcAssociation을 호출할 때 연결 소유자에게 전송됩니다.

`DeleteServiceNetworkVpcAssociationBySharee`

엔터티 소비자가 공유 서비스 네트워크와 DeleteServiceNetworkVpcAssociation을 호출할 때 엔터티 소유자에게 전송됩니다.

`GetServiceBySharee`

엔터티 소비자가 공유 서비스로 GetService를 호출할 때 엔터티 소유자에게 전송됩니다.

`GetServiceNetworkBySharee`

엔터티 소비자가 공유 서비스 네트워크를 사용하여 GetServiceNetwork를 호출할 때 엔터티 소유자에게 전송됩니다.

`GetServiceNetworkResourceAssociationBySharee`

엔터티 소비자가 공유 엔터티와 GetServiceNetworkResourceAssociation을 호출할 때 엔터티 소유자에게 전송됩니다. 호출자가 리소스 구성을 소유한 경우 이벤트는 서비스 네트워크의 소유자에게 전송됩니다. 호출자가 서비스 네트워크를 소유한 경우 이벤트는 리소스 구성의 소유자에게 전송됩니다.

`GetServiceNetworkServiceAssociationBySharee`

엔터티 소비자가 공유 엔터티와 GetServiceNetworkServiceAssociation을 호출할 때 엔터티 소유자에게 전송됩니다. 호출자가 서비스를 소유한 경우 이벤트는 서비스 네트워크 소유자에게 전송됩니다. 호출자가 서비스 네트워크를 소유한 경우 이벤트는 서비스 소유자에게 전송됩니다.

`GetServiceNetworkVpcAssociationBySharee`

엔터티 소비자가 공유 서비스 네트워크를 사용하여 GetServiceNetworkVpcAssociation을 호출할 때 엔터티 소유자에게 전송됩니다.

다음은 `CreateServiceNetworkServiceAssociationBySharee` 이벤트의 예시 항목입니다.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown"
    },
    "eventTime": "2023-04-27T17:12:46Z",
    "eventSource": "vpc-lattice.amazonaws.com",
    "eventName": "CreateServiceNetworkServiceAssociationBySharee",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "vpc-lattice.amazonaws.com",
    "userAgent": "ec2.amazonaws.com",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "callerAccountId": "111122223333"
    },
    "requestID": "ddabb0a7-70c6-4f70-a6c9-00cbe8a6a18b",
    "eventID": "bd03cdca-7edd-4d50-b9c9-eaa89f4a47cd",
    "readOnly": false,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::VpcLattice::ServiceNetworkServiceAssociation",
            "ARN": "arn:aws:vpc-
lattice:region:123456789012:servicenetworkserviceassociation/snsa-0d5ea7bc72EXAMPLE"
        }
    ],
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
```

```
    "eventCategory": "Management"
}
```

용 VPC Lattice Oracle Database@AWS

VPC Lattice는 Oracle Database@AWS (ODB)에 대한 AWS 관리형 서비스 통합을 지원하고 ODB 네트워크, AWS VPCs. 이 연결을 지원하기 위해 VPC Lattice는 사용자를 대신하여 다음 엔터티를 프로비저닝합니다.

기본 서비스 네트워크

기본 서비스 네트워크는 명명 규칙을 사용합니다. `default-odb-network-randomHash`

기본 서비스 네트워크 엔드포인트

이 AWS 리소스에는 이름이 없습니다.

리소스 게이트웨이

리소스 게이트웨이는 명명 규칙을 사용합니다. `default-odb-network-randomHash`

VPC Lattice는 ODB 네트워크에 대한 AWS 관리형 통합이라고 하는 관리형 서비스 통합을 지원합니다. 기본적으로 Oracle Cloud Infrastructure(OCI) Managed Backup to Amazon S3가 활성화됩니다. Amazon S3 및 제로 ETL에 대한 자체 관리형 액세스를 활성화하도록 선택할 수 있습니다.

ODB 네트워크를 생성한 후에는 AWS Management Console 또는를 사용하여 프로비저닝된 리소스를 볼 수 있습니다 AWS CLI. 다음 예제 명령은 ODB 네트워크의 기본 관리형 통합과이 서비스 네트워크에 대해 보유할 수 있는 기타 리소스를 나열합니다.

```
aws vpc-lattice list-service-network-resource-associations \
        --service-network-identifier default-odb-network-randomHash
```

고려 사항

Oracle Database@AWS다음과 같은 고려 사항이 VPC Lattice에 적용됩니다.

- 기본 서비스 네트워크, 서비스 네트워크 엔드포인트, 리소스 게이트웨이 또는 VPC Lattice에서 프로비저닝한 ODB 관리형 통합은 삭제할 수 없습니다. 이러한 엔터티를 삭제하려면 ODB 네트워크를 삭제하거나 관리형 통합을 비활성화합니다.
- 클라이언트는 ODB 네트워크의 관리형 통합에만 액세스할 수 있습니다. VPCs와 같이 ODB 네트워크 외부의 클라이언트는 이러한 관리형 통합을 사용하여 S3 또는 제로 ETL에 액세스할 수 없습니다.
- VPC Lattice에서 프로비저닝한 ODB 네트워크 외부의 관리형 통합에는 연결할 수 없습니다.

- Amazon S3로의 모든 트래픽은 기본 서비스 네트워크 엔드포인트를 거치며 리소스 액세스에 대한 표준 처리 요금이 적용됩니다. 모든 제로 ETL 트래픽은 리소스 게이트웨이를 통과하며 공유하는 리소스에 대한 표준 데이터 처리 요금이 적용됩니다. 자세한 내용은 VPC Lattice 요금을 참조하세요.
- Oracle Database@AWS 관리형 통합에는 시간당 요금이 부과되지 않습니다.
- 다른 서비스 네트워크와 마찬가지로 VPC Lattice에서 프로비저닝한 리소스를 관리할 수 있습니다. 기본 서비스 네트워크를 다른 AWS 계정 또는 조직과 공유하고 새 엔드포인트, VPC 연결, VPC Lattice 서비스 및 리소스를 기본 네트워크에 추가할 수 있습니다.
- VPC Lattice가 Oracle Database@AWS 리소스를 프로비저닝하려면 다음 권한이 필요합니다.

  JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
     {
         "Sid": "AllowODBEC2andLatticeActions",
         "Action": [
             "ec2:DescribeVpcs",
             "ec2:CreateTags",
             "ec2:DescribeAvailabilityZones",
             "ec2:CreateOdbNetworkPeering",
             "ec2:DeleteOdbNetworkPeering",
             "ec2:ModifyOdbNetworkPeering",
             "ec2:DescribeVpcEndpointAssociations",
             "ec2:CreateVpcEndpoint",
             "ec2:DeleteVpcEndpoints",
             "ec2:DescribeVpcEndpoints",
             "vpc-lattice:CreateServiceNetwork",
             "vpc-lattice:DeleteServiceNetwork",
             "vpc-lattice:GetServiceNetwork",
             "vpc-lattice:CreateServiceNetworkResourceAssociation",
             "vpc-lattice:DeleteServiceNetworkResourceAssociation",
             "vpc-lattice:GetServiceNetworkResourceAssociation",
             "vpc-lattice:CreateResourceGateway",
             "vpc-lattice:DeleteResourceGateway",
             "vpc-lattice:GetResourceGateway",
             "vpc-lattice:CreateServiceNetworkVpcEndpointAssociation"
         ],
         "Effect": "Allow",
         "Resource": "*"
     },
     {
```

```
  "Sid": "AllowSLRActionsForLattice",
  "Effect": "Allow",
  "Action": [
   "iam:CreateServiceLinkedRole"
  ],
  "Resource": "*",
   "Condition": {
    "StringEquals": {
     "iam:AWSServiceName": [
      "vpc-lattice.amazonaws.com"
     ]
    }
   }
 }
  ]
}
```

에 VPC Lattice를 사용하려면 VPC Lattice의 서비스 네트워크, 서비스 네트워크 연결 및 리소스 게이트웨이를 숙지하는 Oracle Database@AWS것이 좋습니다.

주제

- the section called “Amazon S3에 대한 Oracle Cloud Infrastructure(OCI) 관리형 백업”
- the section called “Amazon S3 액세스”
- the section called “Amazon Redshift용 제로 ETL”
- the section called “VPC Lattice 엔터티 액세스 및 공유”

Amazon S3에 대한 Oracle Cloud Infrastructure(OCI) 관리형 백업

Oracle Database@AWS 데이터베이스를 생성할 때 VPC Lattice는 라는 리소스 구성을 생성합니다`odb-managed-s3-backup-access`. 이 리소스 구성은 Amazon S3에 대한 데이터베이스의 OCI 관리형 백업을 나타내며 OCI가 소유한 Amazon S3 버킷에만 연결할 수 있습니다. ODB 네트워크와 S3 간의 트래픽은 Amazon 네트워크를 벗어나지 않습니다.

Amazon S3 액세스

Amazon S3에 대한 OCI 관리형 백업 외에도 ODB 네트워크에서 Amazon S3에 액세스할 수 있는 관리형 통합을 생성할 수 있습니다. Amazon S3 Access 관리형 통합을 활성화하도록 Oracle

Database@AWS 네트워크를 수정하면 VPC Lattice는 기본 서비스 네트워크에서 odb-s3-access 라는 리소스 구성을 프로비저닝합니다. 이 통합을 사용하여 자체 관리형 백업 또는 복원을 포함하여 필요에 따라 Amazon S3에 액세스할 수 있습니다. 인증 정책을 제공하여 경계 제어를 설정할 수 있습니다.

고려 사항

다음은 Amazon S3 Access 관리형 통합에 대한 고려 사항입니다.

- ODB 네트워크에 대해 Amazon S3 Access 관리형 통합을 하나만 생성할 수 있습니다.
- 이 관리형 통합을 사용하면 ODB 네트워크에서만 Amazon S3에 액세스할 수 있으며 기본 서비스 네트워크의 다른 VPC 연결 또는 서비스 네트워크 엔드포인트에서는 액세스할 수 없습니다.
- 다른 AWS 리전의 S3 버킷에는 액세스할 수 없습니다.

Amazon S3 Access 관리형 통합 활성화

다음 명령을 사용하여 Amazon S3 Access 관리형 통합을 활성화합니다.

```
aws odb update-odb-network \
  --odb-network-id odb-network-id \
  --s3-access ENABLED
```

인증 정책을 사용한 보안 액세스

ODB API를 사용하여 인증 정책을 정의하여 S3 버킷에 대한 액세스를 보호할 수 있습니다. 다음 예제 정책은 특정 조직이 소유한 특정 S3 버킷에 대한 액세스 권한을 부여합니다.

JSON

```
{
  "Version":"2012-10-17",
  "Id": "Policy1515115909152",
  "Statement": [
    {
      "Sid": "GrantAccessToMyOrgS3",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": [
```

```
        "arn:aws:s3:::awsexamplebucket1",
        "arn:aws:s3:::awsexamplebucket1/*"
      ],
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceOrgID": "o-abcd1234"
        }
      }
    }
  ]
}
```

> **Note**
>
> ODB 관리형 통합을 사용할 때는 S3 버킷 정책에 대해 `aws:SourceVpcaws:SourceVpce`, 및 `aws:VpcSourceIp` 조건 키가 지원되지 않습니다.

Amazon Redshift용 제로 ETL

VPC Lattice에서 프로비저닝한 서비스 네트워크를 사용하여 제로 ETL을 활성화할 수 있습니다. 이 관리형 통합은 ODB 네트워크 데이터베이스를 Amazon Redshift에 연결하여 여러 데이터베이스의 데이터를 분석하는 데 도움이 됩니다. AWS Glue 통합 APIs를 사용하여 제로 ETL 설정을 시작하고 ODB APIs를 사용하여 관리형 통합을 켜고 네트워크 경로를 설정할 수 있습니다. 자세한 내용은 Amazon Redshift와의 제로 ETL 통합을 참조하세요.

고려 사항

다음은 관리형 제로 ETL 통합에 대한 고려 사항입니다.

- 관리형 제로 ETL 통합을 활성화하면 ODB 네트워크의 인스턴스에 액세스하는 데 제로 ETL만 사용할 수 있습니다. 서비스 네트워크와 연결된 다른 서비스 및 리소스는 제로 ETL에서 격리됩니다.

VPC Lattice 엔터티 액세스 및 공유

VPCs Lattice를 사용하여 VPC의 서비스, 리소스 및 기타 클라이언트에 ODB 네트워크를 연결할 수도 있습니다. 이러한 연결 옵션은 VPC Lattice에서 프로비저닝한 기본 서비스 네트워크, 리소스 게이트웨이 및 서비스 네트워크 엔드포인트를 통해 구동됩니다.

VPC Lattice 서비스 및 리소스에 액세스

다른 엔터티에 액세스하려면 소유하거나 공유된 서비스 또는 리소스를 기본 서비스 네트워크에 연결합니다. ODB 네트워크의 클라이언트는 기본 서비스 네트워크 엔드포인트를 통해 서비스 또는 리소스에 액세스할 수 있습니다.

고려 사항

다음은 다른 VPC Lattice 엔터티에 연결하기 위한 고려 사항입니다.

- 새 서비스 네트워크 엔드포인트, VPC 연결, VPC Lattice 리소스 및 서비스를 서비스 네트워크에 추가할 수 있지만 ODB 네트워크를 대신하여 VPC Lattice에서 프로비저닝한 리소스는 수정할 수 없습니다. 이는 Oracle Database@AWS APIs를 통해 관리해야 합니다.

VPC Lattice를 통해 ODB 네트워크 공유

ODB 네트워크 리소스를 다른 VPCs, 계정 또는 온프레미스의 클라이언트와 공유할 수 있습니다. 시작하려면 공유하려는 리소스에 대한 리소스 구성을 생성합니다. 리소스 구성은 ODB 네트워크의 기본 리소스 게이트웨이를 사용해야 합니다. 그런 다음 리소스를 기본 서비스 네트워크에 연결할 수 있습니다.

다른 VPCs의 클라이언트 또는 서비스 네트워크를 공유 AWS 계정 한 클라이언트는 자체 서비스 네트워크 엔드포인트 또는 VPC 연결을 통해 이러한 리소스에 액세스할 수 있습니다. 자세한 내용은 the section called “연결 관리” 단원을 참조하십시오.

고려 사항

다음은 ODB 네트워크 공유 시 고려 사항입니다.

- ODB 네트워크 인스턴스는 IP 기반 리소스로만 공유하는 것이 좋습니다.
- VPC Lattice는 OCI의 단일 클라이언트 액세스 이름(SCAN) 리스너 DNS를 지원하지 않습니다.

Amazon VPC Lattice의 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.

사용자는 이 인프라에 호스팅되는 콘텐츠에 대한 통제 권한을 유지할 책임이 있습니다. 공동 책임 모델은 이 사항을 클라우드의 보안 및 클라우드 내 보안으로 설명합니다.

- 클라우드 보안 - AWS 는에서 AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다 AWS 클라우드. AWS 또한는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. Amazon VPC Lattice에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 AWS 제공 범위 내 서비스규정 준수 프로그램 .
- 클라우드의 보안 – 이 인프라에서 호스팅되는 콘텐츠에 대한 제어를 유지하는 것은 사용자의 책임입니다. 또한 사용자는 데이터의 민감도, 회사 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 VPC Lattice 사용 시 책임 분담 모델을 적용하는 방법을 이해하는 데 도움이 됩니다. 다음 항목에서는 보안 및 규정 준수 목표를 충족하도록 VPC Lattice를 구성하는 방법을 보여줍니다. 또한 VPC Lattice AWS 서비스, 서비스 네트워크 및 리소스 구성을 모니터링하고 보호하는 데 도움이 되는 다른 서비스를 사용하는 방법을 알아봅니다.

내용

VPC Lattice 서비스에 대한 액세스 관리

VPC Lattice는 기본적으로 안전합니다. 액세스 권한을 제공할 서비스 및 리소스 구성과 VPCs에 대한 액세스를 명시적으로 지정해야 하기 때문입니다. VPC 연결 또는 서비스 네트워크 유형의 VPC 엔드

포인트를 통해 서비스에 액세스할 수 있습니다. 다중 계정 시나리오의 경우 AWS Resource Access Manager를 사용하여 계정 경계에서 서비스, 리소스 구성 및 서비스 네트워크를 공유할 수 있습니다.

VPC Lattice는 네트워크의 여러 계층에서 defense-in-depth 전략을 구현할 수 있는 프레임워크를 제공합니다.

- 첫 번째 계층 - 서비스 네트워크와의 서비스, 리소스, VPC 및 VPC 엔드포인트 연결입니다. VPC는 연결을 통해 또는 VPC 엔드포인트를 통해 서비스 네트워크에 연결될 수 있습니다. VPC가 서비스 네트워크에 연결되지 않은 경우 VPC의 클라이언트는 서비스 네트워크와 연결된 서비스 및 리소스 구성에 액세스할 수 없습니다.
- 두 번째 계층 – 서비스 네트워크를 위한 선택적인 네트워크 수준 보안 보호(예: 보안 그룹 및 네트워크 ACL). 이를 사용하면 VPC의 모든 클라이언트 대신 VPC의 특정 클라이언트 그룹에 대한 액세스를 허용할 수 있습니다.
- 세 번째 계층 – 선택적인 VPC Lattice 인증 정책. 인증 정책을 서비스 네트워크 및 개별 서비스에 적용할 수 있습니다. 일반적으로 서비스 네트워크의 인증 정책은 네트워크 또는 클라우드 관리자가 운영하며 세분화되지 않은 인증을 구현합니다. AWS Organizations에 있는 특정 조직의 인증된 요청만 허용하는 경우를 예로 들 수 있습니다. 서비스 수준의 인증 정책의 경우 일반적으로 서비스 소유자가 세분화된 제어를 설정하는데, 이는 서비스 네트워크 수준에서 적용되는 세분화되지 않은 권한 부여보다 더 제한적일 수 있습니다.

> **Note**
>
> 서비스 네트워크의 인증 정책은 서비스 네트워크의 리소스 구성에는 적용되지 않습니다.

액세스 제어 방법

- 인증 정책
- 보안 그룹
- 네트워크 ACL

인증 정책을 사용하여 VPC Lattice 서비스에 대한 액세스 제어

VPC Lattice 인증 정책은 서비스 네트워크 또는 서비스에 연결하여 지정된 보안 주체가 서비스 그룹 또는 특정 서비스에 액세스할 수 있는지 여부를 제어하는 IAM 정책 문서입니다. 액세스를 제어하려는 각 서비스 네트워크 또는 서비스에 하나의 인증 정책을 연결할 수 있습니다.

> **Note**
>
> 서비스 네트워크의 인증 정책은 서비스 네트워크의 리소스 구성에는 적용되지 않습니다.

인증 정책은 IAM 자격 증명 기반 정책과 다릅니다. IAM 자격 증명 기반 정책은 IAM 엔터티(사용자, 그룹 또는 역할)에 연결되어 이들 엔터티가 어떤 리소스에 대해 어떤 조치를 취할 수 있는지 정의합니다. 인증 정책은 서비스 및 서비스 네트워크에 연결됩니다. 승인이 성공하려면 인증 정책과 ID 기반 정책 모두에 명시적 허용 문이 있어야 합니다. 자세한 내용은 권한 부여의 작동 방식 단원을 참조하십시오.

AWS CLI 및 콘솔을 사용하여 서비스 및 서비스 네트워크에 대한 인증 정책을 확인, 추가, 업데이트 또는 제거할 수 있습니다. 인증 정책을 추가, 업데이트 또는 제거할 때 준비하는 데 몇 분 정도 걸릴 수 있습니다. 를 사용할 때 올바른 리전에 있는지 AWS CLI확인합니다. 프로필의 기본 리전을 변경하거나 명령과 함께 `--region` 파라미터를 사용할 수 있습니다.

내용

- 인증 정책의 공통 요소
- 인증 정책의 리소스 형식
- 인증 정책에 사용할 수 있는 조건 키
- 리소스 태그
- 보안 주체 태그
- 익명의(인증되지 않은) 보안 주체
- 인증 정책 예시
- 권한 부여의 작동 방식

인증 정책을 시작하려면 절차에 따라 서비스 네트워크에 적용되는 인증 정책을 만드세요. 다른 서비스에 적용하지 않으려는 보다 제한적인 권한을 원하는 경우 개별 서비스에 인증 정책을 설정할 수 있습니다.

인증 정책으로 서비스 네트워크에 대한 액세스 관리

다음 AWS CLI 작업은 인증 정책을 사용하여 서비스 네트워크에 대한 액세스를 관리하는 방법을 보여줍니다. 콘솔 사용 지침은 VPC Lattice의 서비스 네트워크 섹션을 참조하세요.

작업

- 서비스 네트워크에 인증 정책 추가

- 서비스 네트워크의 인증 유형 변경
- 서비스 네트워크에서 인증 정책 제거

서비스 네트워크에 인증 정책 추가

이 섹션의 단계에 따라를 사용하여 다음을 AWS CLI 수행합니다.

- IAM을 사용하여 서비스 네트워크에 대한 액세스 제어를 활성화합니다.
- 서비스 네트워크에 인증 정책을 추가합니다. 인증 정책을 추가하지 않으면 모든 트래픽에 액세스 거부 오류가 발생합니다.

액세스 제어를 활성화하고 새 서비스 네트워크에 인증 정책을 추가하는 방법

1. 인증 정책을 사용할 수 있도록 서비스 네트워크에 대한 액세스 제어를 활성화하려면 `--auth-type` 옵션과 값이 `AWS_IAM`인 create-service-network 명령을 사용합니다.

   ```
   aws vpc-lattice create-service-network --name Name --auth-type AWS_IAM [--
   tags TagSpecification]
   ```

   이 작업이 성공하면 다음과 비슷한 출력이 반환됩니다.

   ```
   {
      "arn": "arn",
      "authType": "AWS_IAM",
      "id": "sn-0123456789abcdef0",
      "name": "Name"
   }
   ```

2. put-auth-policy 명령을 사용하여 인증 정책을 추가할 서비스 네트워크의 ID와 추가할 인증 정책을 지정합니다.

   예를 들어, 다음 명령을 사용하여 ID가 *sn-0123456789abcdef0*인 서비스 네트워크에 대한 인증 정책을 생성합니다.

   ```
   aws vpc-lattice put-auth-policy --resource-identifier sn-0123456789abcdef0 --
   policy file://policy.json
   ```

   JSON을 사용하여 정책 정의를 생성합니다. 자세한 내용은 인증 정책의 공통 요소 단원을 참조하십시오.

이 작업이 성공하면 다음과 비슷한 출력이 반환됩니다.

```
{
   "policy": "policy",
   "state": "Active"
}
```

액세스 제어를 활성화하고 기존 서비스 네트워크에 인증 정책을 추가하는 방법

1. 인증 정책을 사용할 수 있도록 서비스 네트워크에 대한 액세스 제어를 활성화하려면 `--auth-type` 옵션과 값이 `AWS_IAM`인 update-service-network 명령을 사용합니다.

```
aws vpc-lattice update-service-network --service-network-
identifier sn-0123456789abcdef0 --auth-type AWS_IAM
```

이 작업이 성공하면 다음과 비슷한 출력이 반환됩니다.

```
{
   "arn": "arn",
   "authType": "AWS_IAM",
   "id": "sn-0123456789abcdef0",
   "name": "Name"
}
```

2. put-auth-policy 명령을 사용하여 인증 정책을 추가할 서비스 네트워크의 ID와 추가할 인증 정책을 지정합니다.

```
aws vpc-lattice put-auth-policy --resource-identifier sn-0123456789abcdef0 --
policy file://policy.json
```

JSON을 사용하여 정책 정의를 생성합니다. 자세한 내용은 인증 정책의 공통 요소 단원을 참조하십시오.

이 작업이 성공하면 다음과 비슷한 출력이 반환됩니다.

```
{
   "policy": "policy",
   "state": "Active"
```

```
}
```

서비스 네트워크의 인증 유형 변경

서비스 네트워크에 대한 인증 정책을 비활성화하는 방법

--auth-type 옵션과 값이 NONE인 update-service-network 명령을 사용합니다.

```
aws vpc-lattice update-service-network --service-network-
identifier sn-0123456789abcdef0 --auth-type NONE
```

나중에 인증 정책을 다시 활성화해야 하는 경우 --auth-type 옵션에 AWS_IAM이 지정된 상태로 이 명령을 실행하세요.

서비스 네트워크에서 인증 정책 제거

서비스 네트워크에서 인증 정책을 제거하는 방법

delete-auth-policy 명령을 사용합니다.

```
aws vpc-lattice delete-auth-policy --resource-identifier sn-0123456789abcdef0
```

서비스 네트워크의 인증 유형을 NONE으로 변경하기 전에 인증 정책을 제거하면 요청이 실패합니다.

인증 정책으로 서비스에 대한 액세스 관리

다음 AWS CLI 작업은 인증 정책을 사용하여 서비스에 대한 액세스를 관리하는 방법을 보여줍니다. 콘솔 사용 지침은 VPC Lattice 내 서비스 섹션을 참조하세요.

작업

- 서비스에 인증 정책 추가
- 서비스의 인증 유형 변경
- 서비스에서 인증 정책 제거

서비스에 인증 정책 추가

다음 단계에 따라를 사용하여 다음을 AWS CLI 수행합니다.

- IAM을 사용하여 서비스에 대한 액세스 제어를 활성화합니다.

- 서비스에 인증 정책을 추가합니다. 인증 정책을 추가하지 않으면 모든 트래픽에 액세스 거부 오류가 발생합니다.

액세스 제어를 활성화하고 새 서비스에 인증 정책을 추가하는 방법

1. 인증 정책을 사용할 수 있도록 서비스에 대한 액세스 제어를 활성화하려면 `--auth-type` 옵션과 값이 `AWS_IAM`인 create-service 명령을 사용합니다.

```
aws vpc-lattice create-service --name Name --auth-type AWS_IAM [--
tags TagSpecification]
```

이 작업이 성공하면 다음과 비슷한 출력이 반환됩니다.

```
{
   "arn": "arn",
   "authType": "AWS_IAM",
   "dnsEntry": {
      ...
   },
   "id": "svc-0123456789abcdef0",
   "name": "Name",
   "status": "CREATE_IN_PROGRESS"
}
```

2. put-auth-policy 명령을 사용하여 인증 정책을 추가할 서비스의 ID와 추가할 인증 정책을 지정합니다.

예를 들어, 다음 명령을 사용하여 ID가 *svc-0123456789abcdef0*인 서비스에 대한 인증 정책을 생성합니다.

```
aws vpc-lattice put-auth-policy --resource-identifier svc-0123456789abcdef0 --
policy file://policy.json
```

JSON을 사용하여 정책 정의를 생성합니다. 자세한 내용은 인증 정책의 공통 요소 단원을 참조하십시오.

이 작업이 성공하면 다음과 비슷한 출력이 반환됩니다.

```
{
   "policy": "policy",
```

```
    "state": "Active"
}
```

액세스 제어를 활성화하고 기존 서비스에 인증 정책을 추가하는 방법

1. 인증 정책을 사용할 수 있도록 서비스에 대한 액세스 제어를 활성화하려면 `--auth-type` 옵션과 값이 `AWS_IAM`인 update-service 명령을 사용합니다.

```
aws vpc-lattice update-service --service-identifier svc-0123456789abcdef0 --auth-type AWS_IAM
```

이 작업이 성공하면 다음과 비슷한 출력이 반환됩니다.

```
{
    "arn": "arn",
    "authType": "AWS_IAM",
    "id": "svc-0123456789abcdef0",
    "name": "Name"
}
```

2. put-auth-policy 명령을 사용하여 인증 정책을 추가할 서비스의 ID와 추가할 인증 정책을 지정합니다.

```
aws vpc-lattice put-auth-policy --resource-identifier svc-0123456789abcdef0 --policy file://policy.json
```

JSON을 사용하여 정책 정의를 생성합니다. 자세한 내용은 인증 정책의 공통 요소 단원을 참조하십시오.

이 작업이 성공하면 다음과 비슷한 출력이 반환됩니다.

```
{
    "policy": "policy",
    "state": "Active"
}
```

서비스의 인증 유형 변경

서비스에 대한 인증 정책을 비활성화하는 방법

--auth-type 옵션과 값이 NONE인 update-service 명령을 사용합니다.

```
aws vpc-lattice update-service --service-identifier svc-0123456789abcdef0 --auth-type
 NONE
```

나중에 인증 정책을 다시 활성화해야 하는 경우 --auth-type 옵션에 AWS_IAM이 지정된 상태로 이 명령을 실행하세요.

서비스에서 인증 정책 제거

서비스에서 인증 정책을 제거하는 방법

delete-auth-policy 명령을 사용합니다.

```
aws vpc-lattice delete-auth-policy --resource-identifier svc-0123456789abcdef0
```

서비스의 인증 유형을 NONE으로 변경하기 전에 인증 정책을 제거하면 요청이 실패합니다.

서비스에 대한 인증된 요청을 요구하는 인증 정책을 활성화하는 경우 해당 서비스에 대한 모든 요청에는 Signature Version 4(SigV4)를 사용하여 계산된 유효한 요청 서명이 포함되어야 합니다. 자세한 내용은 Amazon VPC Lattice에 대한 SIGv4 인증 요청 단원을 참조하십시오.

인증 정책의 공통 요소

IAM 정책과 동일한 구문을 사용하여 VPC Lattice 인증 정책이 지정됩니다. 자세한 내용은 IAM 사용 설명서의 자격 증명 기반 정책 및 리소스 기반 정책을 참조하세요.

인증 정책에는 다음 요소가 포함됩니다.

- 보안 주체 - 문에서의 작업 및 리소스에 액세스할 수 있는 사람 또는 애플리케이션입니다. 인증 정책에서 보안 주체는 이 권한의 수신자인 IAM 엔터티입니다. 보안 주체는 IAM 엔티티로 인증되어 서비스 네트워크의 서비스 경우와 같이 특정 리소스 또는 리소스 그룹에 요청할 수 있습니다.

  리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 AWS 서비스가 포함될 수 있습니다. 자세한 내용은 IAM 사용 설명서의 AWS JSON 정책 요소: 보안 주체를 참조하세요.
- 결과 – 지정된 보안 주체가 특정 작업을 요청할 때의 결과입니다. 이는 Allow 또는 Deny일 수 있습니다. 기본적으로 IAM을 사용하여 서비스 또는 서비스 네트워크에 대한 액세스 제어를 활성화하면 보안 주체는 서비스 또는 서비스 네트워크에 요청할 권한이 없습니다.

- 작업 - 권한을 부여하거나 거부하는 특정 API 작업입니다. VPC Lattice는 `vpc-lattice-svcs` 접두사를 사용하는 작업을 지원합니다. 자세한 내용은 서비스 승인 참조의 Amazon VPC Lattice Services에서 정의한 작업을 참조하세요.
- 리소스 – 작업의 영향을 받는 리소스입니다.
- 조건: 조건은 선택 사항으로서 이를 사용하여 정책이 적용되는 시기를 제어할 수 있습니다. 자세한 내용은 서비스 권한 부여 참조의 Amazon VPC Lattice 서비스의 조건 키를 참조하세요.

인증 정책을 생성하고 관리할 때 IAM 정책 생성기를 사용하려고 할 수 있습니다.

요구 사항

JSON의 정책에는 줄 바꿈이나 빈 줄이 포함되어서는 안 됩니다.

인증 정책의 리소스 형식

다음 예와 같이 `<serviceARN>/<path>` 패턴이 있는 매칭 스키마를 사용하고 `Resource` 요소를 코딩하는 인증 정책을 생성하여 특정 리소스에 대한 액세스를 제한할 수 있습니다.

프로토콜	예제
HTTP	• `"Resource": "arn:aws:vpc-lattice:us-west-2:1234567890:service/svc-0123456789abcdef0/rates"` • `"Resource": "*/rates"` • `"Resource": "*/*"`
gRPC	• `"Resource": "arn:aws:vpc-lattice:us-west-2:1234567890:service/svc-0123456789abcdef0/api.parking/GetRates"` • `"Resource": "arn:aws:vpc-lattice:us-west-2:1234567890:service/svc-0123456789abcdef0/api.parking/*"`

프로토콜	예제
	• "Resource": "arn:aws:vpc-lattice:us-west-2:1234567890:service/svc-0123456789abcdef0/*"

<serviceARN>에는 다음의 Amazon 리소스 이름(ARN) 리소스 형식을 사용합니다.

```
arn:aws:vpc-lattice:region:account-id:service/service-id
```

예제:

```
"Resource": "arn:aws:vpc-lattice:us-west-2:123456789012:service/svc-0123456789abcdef0"
```

인증 정책에 사용할 수 있는 조건 키

인증 정책의 조건 요소에 있는 조건 키를 통해 액세스를 추가로 제어할 수 있습니다. 이러한 조건 키는 프로토콜과 요청이 Signature Version 4(SigV4)로 서명되었는지 아니면 익명으로 서명되었는지에 따라 평가용으로 제공됩니다. 조건 키는 대소문자를 구분합니다.

AWS 는 aws:PrincipalOrgID 및와 같이 액세스를 제어하는 데 사용할 수 있는 전역 조건 키를 제공합니다aws:SourceIp. AWS 전역 조건 키 목록을 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

다음 스토리에서는 VPC Lattice 조건 키를 나열합니다. 자세한 내용은 서비스 권한 부여 참조의 Amazon VPC Lattice 서비스의 조건 키를 참조하세요.

조건 키	설명	예제	익명의(인증되지 않은) 호출자가 사용할 수 있나요?	gRPC 사용 가능
vpc-lattice-svcs:Port	요청이 있는 서비스 포트를 기준으로 액세스를 필터링합니다.	80	예	예

조건 키	설명	예제	익명의(인증되지 않은) 호출자가 사용할 수 있나요?	gRPC 사용 가능
vpc-lattice-svcs:RequestMethod	요청의 방법을 기준으로 액세스를 필터링합니다.	GET	예	항상 게시
vpc-lattice-svcs:RequestPath	요청 URL의 경로 부분을 기준으로 액세스를 필터링합니다.	/path	예	예
vpc-lattice-svcs:RequestHeader/ *header-name* : *value*	요청 헤더의 헤더 이름-값 쌍을 기준으로 액세스를 필터링합니다.	content-type: application/json	예	예
vpc-lattice-svcs:RequestQueryString/ *key-name*: *value*	요청 URL에 있는 쿼리 문자열 키-값 페어를 기준으로 액세스를 필터링합니다.	quux: [corge, grault]	예	아니요
vpc-lattice-svcs:ServiceNetworkArn	요청을 수신하는 서비스의 서비스 네트워크에 대한 ARN을 기준으로 액세스를 필터링합니다.	arn:aws:vpc-lattice:us-west-2:123456789012:servicenetwork/sn-0123456789abcdef0	예	예

조건 키	설명	예제	익명의(인증되지 않은) 호출자가 사용할 수 있나요?	gRPC 사용 가능
vpc-lattice-svcs:ServiceArn	요청을 수신하는 서비스의 ARN을 기준으로 액세스를 필터링합니다.	arn:aws:vpc-lattice:us-west-2:123456789012:service/svc-0123456789abcdef0	예	예
vpc-lattice-svcs:SourceVpc	요청이 있는 VPC를 기준으로 액세스를 필터링합니다.	vpc-1a2b3c4d	예	예
vpc-lattice-svcs:SourceVpcOwnerAccount	요청이 있는 VPC의 소유 계정을 기준으로 액세스를 필터링합니다.	123456789012	예	예

리소스 태그

태그는 사용자가 할당하거나가 AWS 리소스에 AWS 할당하는 메타데이터 레이블입니다. 각 태그는 두 부분으로 구성됩니다.

- 태그 키(예: `CostCenter`, `Environment` 또는 `Project`) 태그 키는 대/소문자를 구별합니다.
- 태그 값(예: `111122223333` 또는 `Production`)으로 알려진 선택적 필드 태그 값을 생략하는 것은 빈 문자열을 사용하는 것과 같습니다. 태그 키처럼 태그 값은 대/소문자를 구분합니다.

태그 지정에 대한 자세한 내용은 태그를 사용하여 AWS 리소스에 대한 액세스 제어를 참조하세요.

`aws:ResourceTag/key` AWS 글로벌 조건 컨텍스트 키를 사용하여 인증 정책에서 태그를 사용할 수 있습니다.

다음 예제 정책은 태그가 인 서비스에 대한 액세스 권한을 부여합니다`Environment=Gamma`. 이 정책을 사용하면 하드 코딩 서비스 ARNs 또는 IDs.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGammaAccess",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "vpc-lattice-svcs:Invoke",
      "Resource": "arn:aws:vpc-lattice:us-west-2:123456789012:service/
svc-0124446789abcdef0/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "Gamma",
        }
      }
    }
  ]
}
```

보안 주체 태그

호출자의 자격 증명에 연결된 태그를 기반으로 서비스 및 리소스에 대한 액세스를 제어할 수 있습니다. VPC Lattice는 `aws:PrincipalTag/context` 변수를 사용하여 사용자, 역할 또는 세션 태그의 모든 보안 주체 태그를 기반으로 액세스 제어를 지원합니다. 자세한 내용은 IAM 보안 주체에 대한 액세스 제어를 참조하세요.

다음 예제 정책은 태그가 인 자격 증명에만 액세스 권한을 부여합니다`Team=Payments`. 이 정책을 사용하면 계정 IDs 또는 역할 ARNs.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPaymentsTeam",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "vpc-lattice-svcs:Invoke",
      "Resource": "arn:aws:vpc-lattice:us-west-2:123456789012:service/
svc-0123456789abcdef0/*",
```

```
      "Condition": {
        "StringEquals": {
          "aws:PrincipalTag/Team": "Payments",
        }
      }
    }
  ]
}
```

익명의(인증되지 않은) 보안 주체

익명 보안 주체는 서명 버전 4(SigV4)로 AWS 요청에 서명하지 않고 서비스 네트워크에 연결된 VPC 내에 있는 호출자입니다. 인증 정책에서 허용하는 경우 익명의 보안 주체는 서비스 네트워크의 서비스에 인증되지 않은 요청을 할 수 있습니다.

인증 정책 예시

다음은 인증된 보안 주체가 요청을 해야 하는 인증 정책의 예시입니다.

모든 예시는 us-west-2 리전을 사용하며 가상의 계정 ID를 포함합니다.

예제 1: 특정 AWS 조직의 서비스에 대한 액세스 제한

다음 인증 정책 예시는 정책이 적용되는 서비스 네트워크의 모든 서비스에 액세스할 수 있는 권한을 인증된 모든 요청에 부여합니다. 그러나 요청은 조건에 지정된 AWS 조직에 속한 보안 주체에서 시작되어야 합니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": "vpc-lattice-svcs:Invoke",
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "aws:PrincipalOrgID": [
                  "o-123456example"
```

```
                ]
            }
        }
    }
]
}
```

예 2: 특정 IAM 역할에 의한 서비스 액세스 제한

다음 인증 정책 예시는 Resource 요소에 지정된 서비스에 대해 HTTP GET 요청을 보내기 위해 IAM 역할 rates-client를 사용하는 모든 인증된 요청에 권한을 부여합니다. Resource 요소의 리소스는 정책이 연결된 서비스와 동일합니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect": "Allow",
         "Principal": {
            "AWS": [
               "arn:aws:iam::123456789012:role/rates-client"
            ]
         },
         "Action": "vpc-lattice-svcs:Invoke",
         "Resource": [
            "arn:aws:vpc-lattice:us-
west-2:123456789012:service/svc-0123456789abcdef0/*"
         ],
         "Condition": {
            "StringEquals": {
               "vpc-lattice-svcs:RequestMethod": "GET"
            }
         }
      }
   ]
}
```

예 3: 특정 VPC의 인증된 보안 주체에 의한 서비스 액세스 제한

다음 인증 정책 예시는 VPC ID가 *vpc-1a2b3c4d*인 VPC의 보안 주체가 인증된 요청을 하는 경우만 허용합니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Principal": "*",
         "Action": "vpc-lattice-svcs:Invoke",
         "Resource": "*",
         "Condition": {
            "StringNotEquals": {
               "aws:PrincipalType": "Anonymous"
            },
            "StringEquals": {
               "vpc-lattice-svcs:SourceVpc": "vpc-1a2b3c4d"
            }
         }
      }
   ]
}
```

권한 부여의 작동 방식

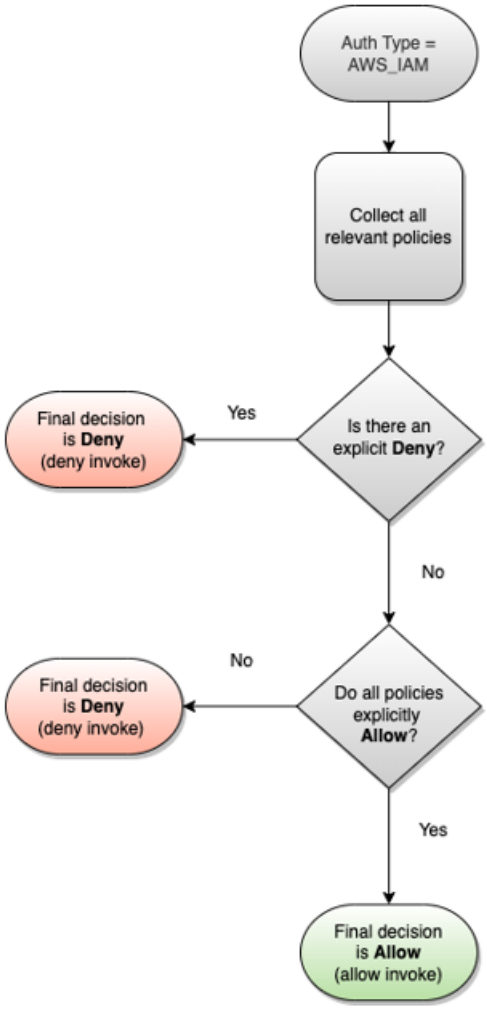
VPC Lattice 서비스가 요청을 수신하면 AWS 적용 코드는 모든 관련 권한 정책을 함께 평가하여 요청을 승인할지 거부할지를 결정합니다. 권한 부여 중에 요청 컨텍스트에 적용할 수 있는 모든 IAM 자격 증명 기반 정책 및 인증 정책을 평가합니다. 기본적으로 인증 유형이 `AWS_IAM`인 경우 모든 요청이 묵시적으로 거부됩니다. 모든 관련 정책의 명시적 허용은 기본 설정을 무시합니다.

승인에는 다음이 포함됩니다.

- 모든 관련 IAM 자격 증명 기반 정책 및 인증 정책 수집
- 정책 세트 결과 평가:
  - 요청자(예: IAM 사용자 또는 역할)에게 요청자가 속한 계정에서 작업을 수행할 수 있는 권한이 있는지 확인합니다. 명시적 허용 문이 없는 경우 요청을 승인하지 AWS 않습니다.

- 서비스 네트워크의 인증 정책에서 해당 요청을 허용하는지 확인합니다. 인증 정책이 활성화되었지만 명시적 허용 문이 없는 경우는 요청을 승인하지 AWS 않습니다. 명시적 허용 문이 있거나 인증 유형이 NONE인 경우 코드는 계속 실행됩니다.
- 서비스에 대한 인증 정책에서 해당 요청을 허용하는지 확인합니다. 인증 정책이 활성화되었지만 명시적 허용 문이 없는 경우는 요청을 승인하지 AWS 않습니다. 명시적 허용 문이 있거나 인증 유형이 NONE인 경우 적용 코드가 최종 Allow 결정을 반환합니다.
- 어떠한 정책의 명시적 거부도 허용을 무시합니다.

다이어그램은 권한 부여 워크플로를 보여줍니다. 요청을 하면 관련 정책은 해당 서비스에 대한 요청 액세스를 허용하거나 거부합니다.



보안 그룹을 사용하여 VPC Lattice의 트래픽 제어

AWS 보안 그룹은 가상 방화벽 역할을 하여 연결된 엔터티와의 네트워크 트래픽을 제어합니다. VPC Lattice를 사용하면 보안 그룹을 생성하고 VPC를 서비스 네트워크에 연결하는 VPC 연결에 할당하여

서비스 네트워크에 추가 네트워크 수준 보안 보호를 적용할 수 있습니다. VPC 엔드포인트를 사용하여 VPC를 서비스 네트워크에 연결하는 경우 VPC 엔드포인트에도 보안 그룹을 할당할 수 있습니다. 마찬가지로 생성한 리소스 게이트웨이에 보안 그룹을 할당하여 VPC의 리소스에 액세스할 수 있습니다.

내용

관리형 접두사 목록

VPC Lattice는 서비스 네트워크 연결을 사용하여 VPC를 서비스 네트워크에 연결할 때 VPC Lattice 네트워크를 통해 트래픽을 라우팅하는 데 사용되는 IP 주소를 포함하는 관리형 접두사 목록을 제공합니다. 이러한 IPs는 프라이빗 링크 로컬 IPs 또는 라우팅할 수 없는 퍼블릭 IPs.

보안 그룹 규칙에서 VPC Lattice 관리형 접두사 목록을 참조할 수 있습니다. 이렇게 하면 트래픽이 클라이언트에서 VPC Lattice 서비스 네트워크를 통해 VPC Lattice 서비스 대상으로 흐를 수 있습니다.

예를 들어 미국 서부(오레곤) 리전(`us-west-2`)에 대상으로 등록된 EC2 인스턴스가 있다고 가정합시다. VPC Lattice 관리형 접두사 목록에서 인바운드 HTTPS 액세스를 허용하는 규칙을 인스턴스 보안 그룹에 추가하면 이 리전의 VPC Lattice 트래픽이 인스턴스에 도달할 수 있습니다. 보안 그룹에서 다른 모든 인바운드 규칙을 제거하면 VPC Lattice 트래픽 이외의 모든 트래픽이 인스턴스에 도달하는 것을 방지할 수 있습니다.

VPC Lattice의 관리형 접두사 목록 이름은 다음과 같습니다.

- com.amazonaws.*region*.vpc-lattice
- com.amazonaws.*region*.ipv6.vpc-lattice

자세한 내용은 Amazon VPC 사용 설명서의 AWS관리형 접두사 목록을 참조하세요.

Windows 및 macOS 클라이언트

VPC Lattice 접두사 목록의 주소는 링크-로컬 주소와 라우팅할 수 없는 퍼블릭 주소입니다. 이러한 클라이언트에서 VPC Lattice에 연결하는 경우 관리형 접두사 목록의 IP 주소를 클라이언트의 기본 IP 주소로 전달하도록 구성을 업데이트해야 합니다. 다음은 Windows 클라이언트의 구성을 업데이트하는 예제 명령입니다. 여기서 169.254.171.0는 관리형 접두사 목록의 주소 중 하나입니다.

```
C:\> route add 169.254.171.0 mask 255.255.255.0 primary-ip-address
```

다음은 macOS 클라이언트의 구성을 업데이트하는 예제 명령입니다. 여기서 169.254.171.0는 관리형 접두사 목록의 주소 중 하나입니다.

```
sudo route -n add -net 169.254.171.0 primary-ip-address 255.255.255.0
```

정적 경로를 생성하지 않으려면 VPC의 서비스 네트워크 엔드포인트를 사용하여 연결을 설정하는 것이 좋습니다. 자세한 내용은 the section called “서비스 네트워크 VPC 엔드포인트 연결 관리” 단원을 참조하십시오.

보안 그룹 규칙

보안 그룹이 있든 없든 VPC Lattice를 사용해도 기존 VPC 보안 그룹 구성에 영향을 주지 않습니다. 그러나 언제든지 자체 보안 그룹을 추가할 수 있습니다.

주요 고려 사항

- 클라이언트에 대한 보안 그룹 규칙은 VPC Lattice로의 아웃바운드 트래픽을 제어합니다.
- 대상에 대한 보안 그룹 규칙은 상태 확인 트래픽을 포함하여 VPC Lattice에서 대상으로 가는 인바운드 트래픽을 제어합니다.
- 서비스 네트워크와 VPC 간의 연결을 위한 보안 그룹 규칙은 VPC Lattice 서비스 네트워크에 액세스할 수 있는 클라이언트를 제어합니다.
- 리소스 게이트웨이에 대한 보안 그룹 규칙은 리소스 게이트웨이에서 리소스로의 아웃바운드 트래픽을 제어합니다.

리소스 게이트웨이에서 데이터베이스 리소스로 흐르는 트래픽에 대한 권장 아웃바운드 규칙

트래픽이 리소스 게이트웨이에서 리소스로 흐르려면 열린 포트에 대한 아웃바운드 규칙과 리소스에 대해 허용되는 리스너 프로토콜을 생성해야 합니다.

Destination	프로토콜	포트 범위	설명
#### CIDR ##	*TCP*	*3306*	리소스 게이트웨이에서 데이터베이스로의 트래픽 허용

서비스 네트워크 및 VPC 연결을 위한 권장 인바운드 규칙

트래픽VPCs에서 서비스 네트워크와 연결된 서비스로 흐르려면 서비스에 대한 리스너 포트 및 리스너 프로토콜에 대한 인바운드 규칙을 생성해야 합니다.

소스	프로토콜	포트 범위	Comment
VPC CIDR	*listener*	*listener*	클라이언트에서 VPC Lattice로의 트래픽 허용

클라이언트 인스턴스에서 VPC Lattice로의 트래픽 흐름에 대한 권장 아웃바운드 규칙

기본적으로 보안 그룹은 모든 아웃바운드 트래픽을 허용합니다. 그러나 사용자 지정 아웃바운드 규칙이 있는 경우 클라이언트 인스턴스가 VPC Lattice 서비스 네트워크와 연결된 모든 서비스에 연결할 수 있도록 리스너 포트 및 프로토콜에 대해 VPC Lattice 접두사로의 아웃바운드 트래픽을 허용해야 합니다. VPC Lattice에 대한 접두사 목록의 ID를 참조하여이 트래픽을 허용할 수 있습니다.

Destination	프로토콜	포트 범위	설명
VPC Lattice ### ### ID	*listener*	*listener*	클라이언트에서 VPC Lattice로의 트래픽 허용

VPC Lattice에서 대상 인스턴스로의 트래픽 흐름에 대한 권장 인바운드 규칙

트래픽이 VPC Lattice에서 흐르기 때문에 클라이언트 보안 그룹을 대상 보안 그룹의 소스로 사용할 수 없습니다. VPC Lattice에 대한 접두사 목록의 ID를 참조할 수 있습니다.

소스	프로토콜	포트 범위	설명
VPC Lattice ### ### ID	*target*	*target*	VPC Lattice에서 대상으로의 트래픽 허용
VPC Lattice ### ### ID	*health check*	*health check*	VPC Lattice에서 대상으로 가는 상태 확인 트래픽 허용

VPC 연결을 위한 보안 그룹 관리

AWS CLI 를 사용하여 VPC에서 서비스 네트워크 연결에 대한 보안 그룹을 보거나 추가하거나 업데이트할 수 있습니다. 를 사용할 때 명령은 프로파일에 대해 AWS 리전 구성된에서 실행 AWS CLI됩니다. 다른 리전에서 명령을 실행하려는 경우 프로필의 기본 리전을 변경하거나 명령에 `--region` 파라미터를 사용합니다.

시작하기 전에 서비스 네트워크에 추가하려는 VPC와 동일한 VPC에 보안 그룹을 생성했는지 확인합니다. 자세한 내용은 Amazon VPC 사용 설명서의 보안 그룹을 사용하여 리소스에 대한 트래픽 제어를 참조하세요.

콘솔을 사용하여 VPC 연결을 생성할 때 보안 그룹을 추가하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스 네트워크를 선택합니다.
3. 서비스 네트워크의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. VPC 연결 탭에서 VPC 연결 생성을 선택한 다음 VPC 연결 추가를 선택합니다.
5. VPC와 최대 5개의 보안 그룹을 선택합니다.
6. 변경 사항 저장을 선택합니다.

콘솔을 사용하여 기존 VPC 연결에 대한 보안 그룹을 추가 또는 업데이트하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 탐색 창의 VPC Lattice에서 서비스 네트워크를 선택합니다.
3. 서비스 네트워크의 이름을 선택하여 세부 정보 페이지를 엽니다.
4. VPC 연결 탭에서 연결의 확인란을 선택한 다음 작업, 보안 그룹 편집을 선택합니다.
5. 필요에 따라 보안 그룹을 추가하고 제거합니다.
6. 변경 사항 저장을 선택합니다.

를 사용하여 VPC 연결을 생성할 때 보안 그룹을 추가하려면 AWS CLI

create-service-network-vpc-association 명령을 사용하여 VPC 연결을 위한 VPC의 ID와 추가할 보안 그룹의 ID를 지정합니다.

```
aws vpc-lattice create-service-network-vpc-association \
    --service-network-identifier sn-0123456789abcdef0 \
    --vpc-identifier vpc-1a2b3c4d \
```

```
    --security-group-ids sg-7c2270198example
```

이 작업이 성공하면 다음과 비슷한 출력이 반환됩니다.

```
{
  "arn": "arn",
  "createdBy": "464296918874",
  "id": "snva-0123456789abcdef0",
  "status": "CREATE_IN_PROGRESS",
  "securityGroupIds": ["sg-7c2270198example"]
}
```

를 사용하여 기존 VPC 연결에 대한 보안 그룹을 추가하거나 업데이트하려면 AWS CLI

update-service-network-vpc-association 명령을 사용하여 서비스 네트워크의 ID와 보안 그룹의 ID를 지정합니다. 이 보안 그룹은 이전에 연결된 보안 그룹을 재정의합니다. 목록을 업데이트할 때 보안 그룹을 하나 이상 정의합니다.

```
aws vpc-lattice update-service-network-vpc-association
    --service-network-vpc-association-identifier sn-903004f88example \
    --security-group-ids sg-7c2270198example sg-903004f88example
```

> **Warning**
>
> 모든 보안 그룹을 제거할 수 없습니다. 대신 먼저 VPC 연결을 삭제한 다음 보안 그룹 없이 VPC 연결을 다시 생성해야 합니다. VPC 연결을 삭제할 때는 주의해야 합니다. 삭제하면 트래픽이 해당 서비스 네트워크에 있는 서비스에 도달하지 못하게 됩니다.

네트워크 ACL을 사용하여 VPC Lattice에 대한 트래픽 제어

네트워크 액세스 제어 목록(ACL)은 서브넷 수준에서 특정 인바운드 또는 아웃바운드 트래픽을 허용하거나 거부합니다. 기본 네트워크 ACL은 인바운드와 아웃바운드 트래픽을 모두 허용합니다. 서브넷에 대한 사용자 지정 네트워크 ACLs을 생성하여 추가 보안 계층을 제공할 수 있습니다. 자세한 내용을 알아보려면 Amazon VPC 사용 설명서의 네트워크 ACL을 참조하세요.

내용

- 클라이언트 서브넷ACLs
- 대상 서브넷ACLs

클라이언트 서브넷ACLs

클라이언트 서브넷ACLs은 클라이언트와 VPC Lattice 간의 트래픽을 허용해야 합니다. VPC Lattice의 관리형 접두사 목록에서 허용할 IP 주소 범위를 가져올 수 있습니다.

다음은 인바운드 규칙의 예입니다.

소스	프로토콜	포트 범위	설명
vpc_latti ce_cidr_block	TCP	1025-65535	VPC Lattice에서 클라이언트로의 트래픽 허용

다음은 아웃바운드 예제입니다.

Destination	프로토콜	포트 범위	설명
vpc_latti ce_cidr_block	*listener*	*listener*	클라이언트에서 VPC Lattice로의 트래픽 허용

대상 서브넷ACLs

대상 서브넷의 네트워크 ACLs은 대상 포트와 상태 확인 포트 모두에서 대상과 VPC Lattice 간의 트래픽을 허용해야 합니다. VPC Lattice의 관리형 접두사 목록에서 허용할 IP 주소 범위를 가져올 수 있습니다.

다음은 인바운드 규칙의 예입니다.

소스	프로토콜	포트 범위	설명
vpc_latti ce_cidr_block	*target*	*target*	VPC Lattice에서 대상으로의 트래픽 허용
vpc_latti ce_cidr_block	*health check*	*health check*	VPC Lattice에서 대상으로 가는 상태 확인 트래픽 허용

다음은 아웃바운드 예제입니다.

Destination	프로토콜	포트 범위	설명
vpc_lattice_cidr_block	*target*	1024-65535	대상에서 VPC Lattice로의 트래픽 허용
vpc_lattice_cidr_block	*health check*	1024-65535	대상에서 VPC Lattice로의 상태 확인 트래픽 허용

Amazon VPC Lattice에 대한 SIGv4 인증 요청

VPC Lattice는 클라이언트 인증에 서명 버전 4(SIGv4) 또는 서명 버전 4A(SIGv4A)를 사용합니다. 자세한 내용은 IAM 사용 설명서의 API 요청용AWS Signature Version 4 섹션을 참조하세요.

고려 사항

- VPC Lattice는 SIGv4 또는 SIGv4A로 서명된 모든 요청을 인증하려고 시도합니다. 인증하지 않으면 요청이 실패합니다.
- VPC Lattice는 페이로드 서명을 지원하지 않습니다. 값이 `"UNSIGNED-PAYLOAD"`로 설정된 `x-amz-content-sha256` 헤더를 보내야 합니다.

예제

- Python
- Java
- Node.js
- Golang
- 골랑 - GRPC

Python

이 예제에서는 네트워크에 등록된 서비스에 대한 보안 연결을 통해 서명된 요청을 보냅니다. 요청을 사용하려는 경우 botocore 패키지는 인증 프로세스를 간소화하지만 엄격히 요구되는 것은 아닙니다. 자세한 내용은 Boto3 설명서의 자격 증명을 참조하세요.

botocore 및 awscrt 패키지를 설치하려면 다음 명령을 사용합니다. 자세한 내용은 AWS CRT Python을 참조하세요.

```
pip install botocore awscrt
```

Lambda에서 클라이언트 애플리케이션을 실행하는 경우 Lambda 계층을 사용하여 필요한 모듈을 설치하거나 배포 패키지에 포함합니다.

다음 예제에서는 자리 표시자 값을 고유한 값으로 바꿉니다.

SIGv4

```
from botocore import crt
import requests
from botocore.awsrequest import AWSRequest
import botocore.session

if __name__ == '__main__':
    session = botocore.session.Session()
    signer = crt.auth.CrtSigV4Auth(session.get_credentials(), 'vpc-lattice-svcs',
 'us-west-2')
    endpoint = 'https://data-svc-022f67d3a42.1234abc.vpc-lattice-svcs.us-
west-2.on.aws'
    data = "some-data-here"
    headers = {'Content-Type': 'application/json', 'x-amz-content-sha256':
 'UNSIGNED-PAYLOAD'}
    request = AWSRequest(method='POST', url=endpoint, data=data, headers=headers)
    request.context["payload_signing_enabled"] = False
    signer.add_auth(request)

    prepped = request.prepare()

    response = requests.post(prepped.url, headers=prepped.headers, data=data)
    print(response.text)
```

SIGv4A

```
from botocore import crt
import requests
from botocore.awsrequest import AWSRequest
import botocore.session
```

```
if __name__ == '__main__':
    session = botocore.session.Session()
    signer = crt.auth.CrtSigV4AsymAuth(session.get_credentials(), 'vpc-lattice-
svcs', '*')
    endpoint = 'https://data-svc-022f67d3a42.1234abc.vpc-lattice-svcs.us-
west-2.on.aws'
    data = "some-data-here"
    headers = {'Content-Type': 'application/json', 'x-amz-content-sha256':
 'UNSIGNED-PAYLOAD'}
    request = AWSRequest(method='POST', url=endpoint, data=data, headers=headers)
    request.context["payload_signing_enabled"] = False
    signer.add_auth(request)

    prepped = request.prepare()

    response = requests.post(prepped.url, headers=prepped.headers, data=data)
    print(response.text)
```

Java

이 예시에서는 사용자 지정 인터셉터를 사용하여 요청 서명을 수행하는 방법을 보여줍니다. 올바른 보안 인증을 가져오는 AWS SDK for Java 2.x의 기본 자격 증명 공급자 클래스를 사용합니다. 특정 보안 인증 공급자를 사용하려면 AWS SDK for Java 2.x에서 하나를 선택할 수 있습니다. 는 HTTPS를 통한 서명되지 않은 페이로드만 AWS SDK for Java 허용합니다. 하지만 HTTP를 통한 서명되지 않은 페이로드를 지원하도록 서명자를 확장할 수 있습니다.

SIGv4

```
package com.example;

import software.amazon.awssdk.http.auth.aws.signer.AwsV4HttpSigner;
import software.amazon.awssdk.http.auth.spi.signer.SignedRequest;

import software.amazon.awssdk.http.SdkHttpMethod;
import software.amazon.awssdk.http.SdkHttpClient;
import software.amazon.awssdk.identity.spi.AwsCredentialsIdentity;
import software.amazon.awssdk.http.SdkHttpRequest;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.http.HttpExecuteRequest;
import software.amazon.awssdk.http.HttpExecuteResponse;
import java.io.IOException;
import java.net.URI;
```

```
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;

public class sigv4 {

    public static void main(String[] args) {
        AwsV4HttpSigner signer = AwsV4HttpSigner.create();

        AwsCredentialsIdentity credentials =
 DefaultCredentialsProvider.create().resolveCredentials();

        if (args.length < 2) {
            System.out.println("Usage: sample <url> <region>");
            System.exit(1);
        }
        // Create the HTTP request to be signed
        var url = args[0];
        SdkHttpRequest httpRequest = SdkHttpRequest.builder()
                .uri(URI.create(url))
                .method(SdkHttpMethod.GET)
                .build();

        SignedRequest signedRequest = signer.sign(r -> r.identity(credentials)
                .request(httpRequest)
                .putProperty(AwsV4HttpSigner.SERVICE_SIGNING_NAME, "vpc-lattice-
svcs")
                .putProperty(AwsV4HttpSigner.PAYLOAD_SIGNING_ENABLED, false)
                .putProperty(AwsV4HttpSigner.REGION_NAME, args[1]));

        System.out.println("[*] Raw request headers:");
        signedRequest.request().headers().forEach((key, values) -> {
            values.forEach(value -> System.out.println("  " + key + ": " + value));
        });

        try (SdkHttpClient httpClient = ApacheHttpClient.create()) {
            HttpExecuteRequest httpExecuteRequest = HttpExecuteRequest.builder()
                    .request(signedRequest.request())
                    .contentStreamProvider(signedRequest.payload().orElse(null))
                    .build();

            System.out.println("[*] Sending request to: " + url);

            HttpExecuteResponse httpResponse =
 httpClient.prepareRequest(httpExecuteRequest).call();
```

```
            System.out.println("[*] Request sent");

            System.out.println("[*] Response status code: " +
 httpResponse.httpResponse().statusCode());
            // Read and print the response body
            httpResponse.responseBody().ifPresent(inputStream -> {
                try {
                    String responseBody = new String(inputStream.readAllBytes());
                    System.out.println("[*] Response body: " + responseBody);
                } catch (IOException e) {
                    System.err.println("[*] Failed to read response body");
                    e.printStackTrace();
                } finally {
                    try {
                        inputStream.close();
                    } catch (IOException e) {
                        System.err.println("[*] Failed to close input stream");
                        e.printStackTrace();
                    }
                }
            });
        } catch (IOException e) {
            System.err.println("[*] HTTP Request Failed.");
            e.printStackTrace();
        }

    }
}
```

SIGv4A

이 예제에서는에 대한 추가 종속성이 필요합니다`software.amazon.awssdk:http-auth-aws-crt`.

```
package com.example;


import software.amazon.awssdk.http.auth.aws.signer.AwsV4aHttpSigner;
import software.amazon.awssdk.http.auth.aws.signer.RegionSet;
import software.amazon.awssdk.http.auth.spi.signer.SignedRequest;

import software.amazon.awssdk.http.SdkHttpMethod;
```

```
import software.amazon.awssdk.http.SdkHttpClient;
import software.amazon.awssdk.identity.spi.AwsCredentialsIdentity;
import software.amazon.awssdk.http.SdkHttpRequest;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.http.HttpExecuteRequest;
import software.amazon.awssdk.http.HttpExecuteResponse;
import java.io.IOException;
import java.net.URI;
import java.util.Arrays;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;

public class sigv4a {

    public static void main(String[] args) {
        AwsV4aHttpSigner signer = AwsV4aHttpSigner.create();

        AwsCredentialsIdentity credentials =
 DefaultCredentialsProvider.create().resolveCredentials();

        if (args.length < 2) {
            System.out.println("Usage: sample <url> <regionset>");
            System.exit(1);
        }
        // Create the HTTP request to be signed
        var url = args[0];
        SdkHttpRequest httpRequest = SdkHttpRequest.builder()
                .uri(URI.create(url))
                .method(SdkHttpMethod.GET)
                .build();

        SignedRequest signedRequest = signer.sign(r -> r.identity(credentials)
                .request(httpRequest)
                .putProperty(AwsV4aHttpSigner.SERVICE_SIGNING_NAME, "vpc-lattice-
svcs")
                .putProperty(AwsV4aHttpSigner.PAYLOAD_SIGNING_ENABLED, false)
                .putProperty(AwsV4aHttpSigner.REGION_SET,
 RegionSet.create(String.join(" ",Arrays.copyOfRange(args, 1, args.length)))));

        System.out.println("[*] Raw request headers:");
        signedRequest.request().headers().forEach((key, values) -> {
            values.forEach(value -> System.out.println("  " + key + ": " + value));
        });
```

```
        try (SdkHttpClient httpClient = ApacheHttpClient.create()) {
            HttpExecuteRequest httpExecuteRequest = HttpExecuteRequest.builder()
                    .request(signedRequest.request())
                    .contentStreamProvider(signedRequest.payload().orElse(null))
                    .build();

            System.out.println("[*] Sending request to: " + url);

            HttpExecuteResponse httpResponse =
 httpClient.prepareRequest(httpExecuteRequest).call();

            System.out.println("[*] Request sent");

            System.out.println("[*] Response status code: " +
 httpResponse.httpResponse().statusCode());
            // Read and print the response body
            httpResponse.responseBody().ifPresent(inputStream -> {
                try {
                    String responseBody = new String(inputStream.readAllBytes());
                    System.out.println("[*] Response body: " + responseBody);
                } catch (IOException e) {
                    System.err.println("[*] Failed to read response body");
                    e.printStackTrace();
                } finally {
                    try {
                        inputStream.close();
                    } catch (IOException e) {
                        System.err.println("[*] Failed to close input stream");
                        e.printStackTrace();
                    }
                }
            });
        } catch (IOException e) {
            System.err.println("[*] HTTP Request Failed.");
            e.printStackTrace();
        }
    }
}
```

Node.js

이 예시에서는 aws-crt NodeJS 바인딩을 사용하여 HTTPS를 사용하여 서명된 요청을 보냅니다.

aws-crt 패키지를 설치하려면 다음 명령을 실행합니다.

```
npm -i aws-crt
```

AWS_REGION 환경 변수가 있는 경우 예시에서는 AWS_REGION에서 지정한 리전을 사용합니다. 기본 리전은 us-east-1입니다.

SIGv4

```
const https = require('https')
const crt = require('aws-crt')
const { HttpRequest } = require('aws-crt/dist/native/http')

function sigV4Sign(method, endpoint, service, algorithm) {
    const host = new URL(endpoint).host
    const request = new HttpRequest(method, endpoint)
    request.headers.add('host', host)
    // crt.io.enable_logging(crt.io.LogLevel.INFO)
    const config = {
        service: service,
        region: process.env.AWS_REGION ? process.env.AWS_REGION : 'us-east-1',
        algorithm: algorithm,
        signature_type: crt.auth.AwsSignatureType.HttpRequestViaHeaders,
        signed_body_header: crt.auth.AwsSignedBodyHeaderType.XAmzContentSha256,
        signed_body_value: crt.auth.AwsSignedBodyValue.UnsignedPayload,
        provider: crt.auth.AwsCredentialsProvider.newDefault()
    }

    return crt.auth.aws_sign_request(request, config)
}

if (process.argv.length === 2) {
  console.error(process.argv[1] + ' <url>')
  process.exit(1)
}

const algorithm = crt.auth.AwsSigningAlgorithm.SigV4;

sigV4Sign('GET', process.argv[2], 'vpc-lattice-svcs', algorithm).then(
  httpResponse => {
    var headers = {}

    for (const sigv4header of httpResponse.headers) {
```

```
      headers[sigv4header[0]] = sigv4header[1]
    }

    const options = {
      hostname: new URL(process.argv[2]).host,
      path: new URL(process.argv[2]).pathname,
      method: 'GET',
      headers: headers
    }

    req = https.request(options, res => {
      console.log('statusCode:', res.statusCode)
      console.log('headers:', res.headers)
      res.on('data', d => {
        process.stdout.write(d)
      })
    })
    req.on('error', err => {
      console.log('Error: ' + err)
    })
    req.end()
  }
)
```

SIGv4A

```
const https = require('https')
const crt = require('aws-crt')
const { HttpRequest } = require('aws-crt/dist/native/http')

function sigV4Sign(method, endpoint, service, algorithm) {
    const host = new URL(endpoint).host
    const request = new HttpRequest(method, endpoint)
    request.headers.add('host', host)
    // crt.io.enable_logging(crt.io.LogLevel.INFO)
    const config = {
        service: service,
        region: process.env.AWS_REGION ? process.env.AWS_REGION : 'us-east-1',
        algorithm: algorithm,
        signature_type: crt.auth.AwsSignatureType.HttpRequestViaHeaders,
        signed_body_header: crt.auth.AwsSignedBodyHeaderType.XAmzContentSha256,
        signed_body_value: crt.auth.AwsSignedBodyValue.UnsignedPayload,
        provider: crt.auth.AwsCredentialsProvider.newDefault()
```

```
    }

    return crt.auth.aws_sign_request(request, config)
}

if (process.argv.length === 2) {
  console.error(process.argv[1] + ' <url>')
  process.exit(1)
}

const algorithm = crt.auth.AwsSigningAlgorithm.SigV4Asymmetric;

sigV4Sign('GET', process.argv[2], 'vpc-lattice-svcs', algorithm).then(
  httpResponse => {
    var headers = {}

    for (const sigv4header of httpResponse.headers) {
      headers[sigv4header[0]] = sigv4header[1]
    }

    const options = {
      hostname: new URL(process.argv[2]).host,
      path: new URL(process.argv[2]).pathname,
      method: 'GET',
      headers: headers
    }

    req = https.request(options, res => {
      console.log('statusCode:', res.statusCode)
      console.log('headers:', res.headers)
      res.on('data', d => {
        process.stdout.write(d)
      })
    })
    req.on('error', err => {
      console.log('Error: ' + err)
    })
    req.end()
  }
)
```

Golang

이 예제에서는 Go용 Smithy 코드 생성기와 AWS Go 프로그래밍 언어용 SDK를 사용하여 요청 서명 요청을 처리합니다. 이 예제에서는 Go 버전 1.21 이상이 필요합니다.

SIGv4

```
package main

import (
        "context"
        "flag"
        "fmt"
        "io"
        "log"
        "net/http"
        "net/http/httputil"
        "os"
        "strings"

        "github.com/aws/aws-sdk-go-v2/aws"
        "github.com/aws/aws-sdk-go-v2/config"
        "github.com/aws/smithy-go/aws-http-auth/credentials"
        "github.com/aws/smithy-go/aws-http-auth/sigv4"
        v4 "github.com/aws/smithy-go/aws-http-auth/v4"
)

type nopCloser struct {
        io.ReadSeeker
}

func (nopCloser) Close() error {
        return nil
}

type stringFlag struct {
        set   bool
        value string
}


        flag.PrintDefaults()
        os.Exit(1)
```

```
}

func main() {
        flag.Parse()
        if !url.set || !region.set {
                Usage()
        }

        cfg, err := config.LoadDefaultConfig(context.TODO(),
 config.WithClientLogMode(aws.LogSigning))
        if err != nil {
                log.Fatalf("failed to load SDK configuration, %v", err)
        }

        if len(os.Args) < 2 {
                log.Fatalf("Usage: go run main.go  <url>")
        }

        // Retrieve credentials from an SDK source, such as the instance profile
        sdkCreds, err := cfg.Credentials.Retrieve(context.TODO())
        if err != nil {
                log.Fatalf("Unable to retrieve credentials from SDK, %v", err)
        }

        creds := credentials.Credentials{
                AccessKeyID:     sdkCreds.AccessKeyID,
                SecretAccessKey: sdkCreds.SecretAccessKey,
                SessionToken:    sdkCreds.SessionToken,
        }

        // Add a payload body, which will not be part of the signature calculation
        body := nopCloser{strings.NewReader(`Example payload body`)}

        req, _ := http.NewRequest(http.MethodPost, url.value, body)

        // Create a sigv4a signer with specific options
        signer := sigv4.New(func(o *v4.SignerOptions) {
                o.DisableDoublePathEscape = true
                // This will add the UNSIGNED-PAYLOAD sha256 header
                o.AddPayloadHashHeader = true
                o.DisableImplicitPayloadHashing = true
        })
```

```
        // Perform the signing on req, using the credentials we retrieved from the
 SDK
        err = signer.SignRequest(&sigv4.SignRequestInput{
                Request:     req,
                Credentials: creds,
                Service:     "vpc-lattice-svcs",
                Region: region.String(),
        })

        if err != nil {
                log.Fatalf("%s", err)
        }

        res, err := httputil.DumpRequest(req, true)

        if err != nil {
                log.Fatalf("%s", err)
        }

        log.Printf("[*] Raw request\n%s\n", string(res))

        log.Printf("[*] Sending request to %s\n", url.value)

        resp, err := http.DefaultClient.Do(req)
        if err != nil {
                log.Fatalf("%s", err)
        }

        log.Printf("[*] Request sent\n")

        log.Printf("[*] Response status code: %d\n", resp.StatusCode)

        respBody, err := io.ReadAll(resp.Body)
        if err != nil {
                log.Fatalf("%s", err)
        }

        log.Printf("[*] Response body: \n%s\n", respBody)
}
```

SIGv4A

```
package main
```

```
import (
        "context"
        "flag"
        "fmt"
        "io"
        "log"
        "net/http"
        "net/http/httputil"
        "os"
        "strings"

        "github.com/aws/aws-sdk-go-v2/aws"
        "github.com/aws/aws-sdk-go-v2/config"
        "github.com/aws/smithy-go/aws-http-auth/credentials"
        "github.com/aws/smithy-go/aws-http-auth/sigv4a"
        v4 "github.com/aws/smithy-go/aws-http-auth/v4"
)

type nopCloser struct {
        io.ReadSeeker
}

func (nopCloser) Close() error {
        return nil
}

type stringFlag struct {

func main() {
        flag.Parse()
        if !url.set || !regionSet.set {
                Usage()
        }

        cfg, err := config.LoadDefaultConfig(context.TODO(),
 config.WithClientLogMode(aws.LogSigning))
        if err != nil {
                log.Fatalf("failed to load SDK configuration, %v", err)
        }

        if len(os.Args) < 2 {
                log.Fatalf("Usage: go run main.go <url>")
        }
```

```
	// Retrieve credentials from an SDK source, such as the instance profile
	sdkCreds, err := cfg.Credentials.Retrieve(context.TODO())
	if err != nil {
		log.Fatalf("Unable to retrieve credentials from SDK, %v", err)
	}

	creds := credentials.Credentials{
		AccessKeyID:     sdkCreds.AccessKeyID,
		SecretAccessKey: sdkCreds.SecretAccessKey,
		SessionToken:    sdkCreds.SessionToken,
	}

	// Add a payload body, which will not be part of the signature calculation
	body := nopCloser{strings.NewReader(`Example payload body`)}

	req, _ := http.NewRequest(http.MethodPost, url.value, body)

	// Create a sigv4a signer with specific options
	signer := sigv4a.New(func(o *v4.SignerOptions) {
		o.DisableDoublePathEscape = true
		// This will add the UNSIGNED-PAYLOAD sha256 header
		o.AddPayloadHashHeader = true
		o.DisableImplicitPayloadHashing = true
	})

	// Create a slice out of the provided regionset
	rs := strings.Split(regionSet.value, ",")

	// Perform the signing on req, using the credentials we retrieved from the
SDK
	err = signer.SignRequest(&sigv4a.SignRequestInput{
		Request:     req,
		Credentials: creds,
		Service:     "vpc-lattice-svcs",
		RegionSet: rs,
	})

	if err != nil {
		log.Fatalf("%s", err)
	}

	res, err := httputil.DumpRequest(req, true)
```

```
        if err != nil {
                log.Fatalf("%s", err)
        }

        log.Printf("[*] Raw request\n%s\n", string(res))

        log.Printf("[*] Sending request to %s\n", url.value)

        resp, err := http.DefaultClient.Do(req)
        if err != nil {
                log.Fatalf("%s", err)
        }

        log.Printf("[*] Request sent\n")

        log.Printf("[*] Response status code: %d\n", resp.StatusCode)

        respBody, err := io.ReadAll(resp.Body)
        if err != nil {
                log.Fatalf("%s", err)
        }

        log.Printf("[*] Response body: \n%s\n", respBody)
}
```

골랑 - GRPC

이 예제에서는 AWS Go 프로그래밍 언어용 SDK를 사용하여 GRPC 요청에 대한 요청 서명을 처리합니다. 이는 GRPC 샘플 코드 리포지토리의 에코 서버에서 사용할 수 있습니다.

```
package main

import (
    "context"
    "crypto/tls"
    "crypto/x509"

    "flag"
    "fmt"
    "log"
    "net/http"
    "net/url"
```

```
    "strings"
    "time"

    "google.golang.org/grpc"
    "google.golang.org/grpc/credentials"

    "github.com/aws/aws-sdk-go-v2/aws"
    v4 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/config"

    ecpb "google.golang.org/grpc/examples/features/proto/echo"
)

const (
    headerContentSha    = "x-amz-content-sha256"
    headerSecurityToken = "x-amz-security-token"
    headerDate          = "x-amz-date"
    headerAuthorization = "authorization"
    unsignedPayload     = "UNSIGNED-PAYLOAD"
)

type SigV4GrpcSigner struct {
    service      string
    region       string
    credProvider aws.CredentialsProvider
    signer       *v4.Signer
}

func NewSigV4GrpcSigner(service string, region string, credProvider
 aws.CredentialsProvider) *SigV4GrpcSigner {
    signer := v4.NewSigner()
    return &SigV4GrpcSigner{
        service:      service,
        region:       region,
        credProvider: credProvider,
        signer:       signer,
    }
}

func (s *SigV4GrpcSigner) GetRequestMetadata(ctx context.Context, uri ...string)
 (map[string]string, error) {
    ri, _ := credentials.RequestInfoFromContext(ctx)
    creds, err := s.credProvider.Retrieve(ctx)
    if err != nil {
```

```
        return nil, fmt.Errorf("failed to load credentials: %w", err)
    }

    // The URI we get here is scheme://authority/service/ - for siging we want to
include the RPC name
    // But RequestInfoFromContext only has the combined /service/rpc-name - so read the
URI, and
    // replace the Path with what we get from RequestInfo.
    parsed, err := url.Parse(uri[0])
    if err != nil {
        return nil, err
    }
    parsed.Path = ri.Method

    // Build a request for the signer.
    bodyReader := strings.NewReader("")
    req, err := http.NewRequest("POST", uri[0], bodyReader)
    if err != nil {
        return nil, err
    }
    date := time.Now()
    req.Header.Set(headerContentSha, unsignedPayload)
    req.Header.Set(headerDate, date.String())
    if creds.SessionToken != "" {
        req.Header.Set(headerSecurityToken, creds.SessionToken)
    }
    // The signer wants this as //authority/path
    // So get this by triming off the scheme and the colon before the first slash.
    req.URL.Opaque = strings.TrimPrefix(parsed.String(), parsed.Scheme+":")

    err = s.signer.SignHTTP(context.Background(), creds, req, unsignedPayload,
s.service, s.region, date)
    if err != nil {
        return nil, fmt.Errorf("failed to sign request: %w", err)
    }

    // Pull the relevant headers out of the signer, and return them to get
    // included in the request we make.
    reqHeaders := map[string]string{
        headerContentSha:    req.Header.Get(headerContentSha),
        headerDate:          req.Header.Get(headerDate),
        headerAuthorization: req.Header.Get(headerAuthorization),
    }
    if req.Header.Get(headerSecurityToken) != "" {
```

```
        reqHeaders[headerSecurityToken] = req.Header.Get(headerSecurityToken)
    }

    return reqHeaders, nil
}

func (c *SigV4GrpcSigner) RequireTransportSecurity() bool {
    return true
}

var addr = flag.String("addr", "some-lattice-service:443", "the address to connect to")
var region = flag.String("region", "us-west-2", "region")

func callUnaryEcho(client ecpb.EchoClient, message string) {
    ctx, cancel := context.WithTimeout(context.Background(), 10*time.Second)
    defer cancel()
    resp, err := client.UnaryEcho(ctx, &ecpb.EchoRequest{Message: message})
    if err != nil {
        log.Fatalf("client.UnaryEcho(_) = _, %v: ", err)
    }
    fmt.Println("UnaryEcho: ", resp.Message)
}

func main() {
    flag.Parse()
    cfg, err := config.LoadDefaultConfig(context.TODO(),
 config.WithClientLogMode(aws.LogSigning))
    if err != nil {
        log.Fatalf("failed to load SDK configuration, %v", err)
    }

    pool, _ := x509.SystemCertPool()
    tlsConfig := &tls.Config{
        RootCAs: pool,
    }

    authority, _, _ := strings.Cut(*addr, ":") // Remove the port from the addr
    opts := []grpc.DialOption{
        grpc.WithTransportCredentials(credentials.NewTLS(tlsConfig)),

        // Lattice needs both the Authority to be set (without a port), and the SigV4
 signer
        grpc.WithAuthority(authority),
```

```
        grpc.WithPerRPCCredentials(NewSigV4GrpcSigner("vpc-lattice-svcs", *region,
 cfg.Credentials)),
    }

    conn, err := grpc.Dial(*addr, opts...)

    if err != nil {
        log.Fatalf("did not connect: %v", err)
    }
    defer conn.Close()
    rgc := ecpb.NewEchoClient(conn)

    callUnaryEcho(rgc, "hello world")
}
```

Amazon VPC Lattice의 데이터 보호

AWS 공동 책임 모델 Amazon VPC Lattice의 데이터 보호에 적용됩니다. 이 모델에 설명된 대로 AWS는 모든를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 이 콘텐츠에는 사용하는 AWS 서비스 서비스의 보안 구성과 관리 작업이 포함되어 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.

전송 중 암호화

VPC Lattice는 컨트롤 플레인과 데이터 영역으로 구성된 완전 관리형 서비스입니다. 각 영역은 서비스에서 서로 다른 용도로 사용됩니다. 컨트롤 플레인은 (CRUDL) 리소스(예: `CreateService` 및)를 생성, 읽기/설명, 업데이트, 삭제 및 나열하는 데 사용되는 관리 APIs를 제공합니다`UpdateService`. VPC Lattice 컨트롤 플레인과의 통신은 TLS에 의해 전송 중으로 보호됩니다. 데이터 영역은 서비스 간 상호 연결을 제공하는 VPC Lattice Invoke API입니다. TLS는 HTTPS 또는 TLS를 사용할 때 VPC Lattice 데이터 영역과의 통신을 암호화합니다. 암호 그룹 및 프로토콜 버전은 VPC Lattice에서 제공하는 기본값을 사용하며, 구성할 수 없습니다. 자세한 내용은 VPC Lattice 서비스를 위한 HTTPS 리스너 단원을 참조하십시오.

저장 중 암호화

기본적으로 저장 데이터를 암호화하면 민감한 데이터 보호와 관련된 운영 오버헤드와 복잡성을 줄이는 데 도움이 됩니다. 동시에 엄격한 암호화 규정 준수 및 규제 요구 사항을 충족하는 안전한 애플리케이션을 구축할 수 있습니다.

내용

Amazon S3 관리형 키를 사용한 서버 측 암호화(SSE-S3)

Amazon S3 관리형 키를 사용한 서버 측 암호화(SSE-S3)를 사용하면 각 객체는 고유한 키로 암호화됩니다. 추가 보호 조치로 정기적으로 교체하는 루트 키로 키 자체를 암호화합니다. Amazon S3 서버 측 암호화는 가장 강력한 블록 암호 중 하나인 256비트 Advanced Encryption Standard(AES-256) GCM을 사용하여 데이터를 암호화합니다. AES-GCM 이전에 암호화된 객체의 경우 AES-CBC는 여전히 해당 객체의 암호를 해독하도록 지원됩니다. 자세한 내용은 Amazon S3-관리형 암호화 키(SSE-S3)와 함께 서버 측 암호화 사용을 참조하세요.

VPC Lattice 액세스 로그용 S3-managed 암호화 키(SSE-S3)를 사용하여 서버 측 암호화를 활성화하면 각 액세스 로그 파일이 S3 버킷에 저장되기 전에 자동으로 암호화됩니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon S3으로 전송된 로그를 참조하세요.

에 저장된 AWS KMS 키를 사용한 서버 측 암호화 AWS KMS (SSE-KMS)

AWS KMS 키를 사용한 서버 측 암호화(SSE-KMS)는 SSE-S3와 유사하지만이 서비스 사용에 따른 추가 이점과 요금이 있습니다. Amazon S3에서 객체에 대한 무단 액세스에 대한 추가 보호를 제공하는 AWS KMS 키에 대한 별도의 권한이 있습니다. 또한 SSE-KMS는 AWS KMS 키를 언제 누가 사용했는지 보여주는 감사 추적을 제공합니다. 자세한 내용은 AWS Key Management Service (SSE-KMS)로 서버 측 암호화 사용을 참조하세요.

내용

인증서 프라이빗 키의 암호화 및 암호 해독

ACM 인증서와 프라이빗 키는 별칭이 aws/acm인 AWS 관리형 KMS 키를 사용하여 암호화됩니다. AWS KMS 콘솔의 AWS 관리형 키에서이 별칭을 사용하여 키 ID를 볼 수 있습니다.

VPC Lattice는 ACM 리소스에 직접 액세스하지 않습니다. AWS TLS Connection Manager를 사용하여 인증서의 프라이빗 키를 보호하고 액세스합니다. ACM 인증서를 사용하여 VPC Lattice 서비스를 생성

할 때 VPC Lattice는 인증서를 AWS TLS 연결 관리자와 연결합니다. 이는 접두사 aws/acm을 사용하여 관리형 키에 AWS KMS 대한 AWS 권한을에 생성하여 수행됩니다. 권한 부여는 TLS 연결 관리자가 암호화 작업에서 KMS 키를 사용할 수 있도록 하는 정책 도구입니다. 권한 부여를 통해 피부여자 보안 주체(TLS 연결 관리자)는 KMS 키에 대해 지정된 권한 부여 작업을 호출해 인증서의 프라이빗 키를 해독할 수 있습니다. 그런 다음 TLS 연결 관리자는 인증서와 해독된(일반 텍스트) 프라이빗 키를 사용하여 VPC Lattice 서비스의 클라이언트와 보안 연결(SSL/TLS 세션)을 설정합니다. 인증서가 통합 서비스에서 연결 해제되면 권한 부여가 사용 중지됩니다.

KMS 키에 대한 액세스를 제거하려면 또는의 `update-service` 명령을 사용하여 서비스에서 인증서를 교체 AWS Management Console 하거나 삭제하는 것이 좋습니다 AWS CLI.

VPC Lattice의 암호화 컨텍스트

암호화 컨텍스트는 프라이빗 키가 사용될 수 있는 대상에 대한 컨텍스트 정보를 포함하는 선택적 키-값 페어 세트입니다.는 암호화 컨텍스트를 암호화된 데이터에 AWS KMS 바인딩하고 이를 인증된 암호화를 지원하기 위한 추가 인증 데이터로 사용합니다.

VPC Lattice와 TLS 연결 관리자에서 TLS 키를 사용하는 경우, 저장된 키를 암호화하는 데 사용되는 암호화 컨텍스트에 VPC Lattice 서비스의 이름이 포함됩니다. 다음 섹션과 같이 CloudTrail 로그의 암호화 컨텍스트를 보거나 ACM 콘솔의 연결된 리소스 탭을 확인하여 인증서와 프라이빗 키가 사용되는 VPC Lattice 서비스를 확인할 수 있습니다.

데이터를 해독하기 위해 동일한 암호화 컨텍스트를 요청에 포함시킵니다. VPC Lattice는 모든 AWS KMS 암호화 작업에서 동일한 암호화 컨텍스트를 사용합니다. 여기서 키는 `aws:vpc-lattice:arn`이고 값은 VPC Lattice 서비스의 Amazon 리소스 이름(ARN)입니다.

다음 예시에서는 작업(예: `CreateGrant`) 출력의 암호화 컨텍스트를 보여줍니다.

```
"encryptionContextEquals": {
    "aws:acm:arn": "arn:aws:acm:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "aws:vpc-lattice:arn": "arn:aws:vpc-lattice:us-west-2:111122223333:service/
svc-0b23c1234567890ab"
}
```

VPC Lattice의 암호화 키 모니터링

VPC Lattice 서비스와 함께 AWS 관리형 키를 사용하는 경우 AWS CloudTrail를 사용하여 VPC Lattice가 보내는 요청을 추적할 수 있습니다 AWS KMS.

CreateGrant

VPC Lattice 서비스에 ACM 인증서를 추가하면 사용자를 대신하여 TLS 연결 관리자가 ACM 인증서와 연결된 프라이빗 키를 해독할 수 있도록 해달라는 `CreateGrant` 요청이 전송됩니다.

CloudTrail, 이벤트 기록, CreateGrant에서 `CreateGrant` 작업을 이벤트로 볼 수 있습니다.

다음은 `CreateGrant` 작업에 대한 CloudTrail 이벤트 기록의 이벤트 레코드 예제입니다.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "sessionContext": {
            "sessionIssuer": {
                "type": "IAMUser",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111122223333:user/Alice",
                "accountId": "111122223333",
                "userName": "Alice"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-06T23:30:50Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "acm.amazonaws.com"
    },
    "eventTime": "2023-02-07T00:07:18Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "acm.amazonaws.com",
    "userAgent": "acm.amazonaws.com",
    "requestParameters": {
        "granteePrincipal": "tlsconnectionmanager.amazonaws.com",
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "operations": [
            "Decrypt"
        ],
        "constraints": {
```

```
            "encryptionContextEquals": {
                "aws:acm:arn": "arn:aws:acm:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                "aws:vpc-lattice:arn": "arn:aws:vpc-lattice:us-
west-2:111122223333:service/svc-0b23c1234567890ab"
            }
        },
        "retiringPrincipal": "acm.us-west-2.amazonaws.com"
    },
    "responseElements": {
        "grantId": "f020fe75197b93991dc8491d6f19dd3cebb24ee62277a05914386724f3d48758",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "ba178361-8ab6-4bdd-9aa2-0d1a44b2974a",
    "eventID": "8d449963-1120-4d0c-9479-f76de11ce609",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


위 `CreateGrant` 예제에서 피부여자 보안 주체는 TLS Connection Manager이고 암호화 컨텍스트에는 VPC Lattice 서비스 ARN이 있습니다.

ListGrants

KMS 키 ID와 계정 ID를 사용하여 `ListGrants` API를 호출할 수 있습니다. 그러면 지정된 KMS 키에 대한 모든 권한 부여 목록이 표시됩니다. 자세한 내용은 ListGrants를 참조하세요.

에서 다음 `ListGrants` 명령을 사용하여 모든 권한 부여의 세부 정보를 AWS CLI 확인합니다.

```
aws kms list-grants –key-id your-kms-key-id
```

다음은 예제 출력입니다.

```
{
    "Grants": [
        {
            "Operations": [
                "Decrypt"
            ],
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "Name": "IssuedThroughACM",
            "RetiringPrincipal": "acm.us-west-2.amazonaws.com",
            "GranteePrincipal": "tlsconnectionmanager.amazonaws.com",
            "GrantId":
 "f020fe75197b93991dc8491d6f19dd3cebb24ee62277a05914386724f3d48758",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "CreationDate": "2023-02-06T23:30:50Z",
            "Constraints": {
                "encryptionContextEquals": {
                  "aws:acm:arn": "arn:aws:acm:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                  "aws:vpc-lattice:arn": "arn:aws:vpc-lattice:us-
west-2:111122223333:service/svc-0b23c1234567890ab"
                }
            }
        }
    ]
}
```

위 `ListGrants` 예제에서 피부여자 보안 주체는 TLS Connection Manager이고 암호화 컨텍스트에는 VPC Lattice 서비스 ARN이 있습니다.

Decrypt

VPC Lattice는 VPC Lattice 서비스에서 TLS 연결을 제공하기 위해 TLS 연결 관리자를 사용하여 프라이빗 키를 해독하는 `Decrypt` 작업을 호출합니다. CloudTrail 이벤트 기록, 암호 해독에서 `Decrypt` 작업을 이벤트로 볼 수 있습니다.

다음은 `Decrypt` 작업에 대한 CloudTrail 이벤트 기록의 예제 이벤트 레코드입니다.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "type": "AWSService",
        "invokedBy": "tlsconnectionmanager.amazonaws.com"
    },
    "eventTime": "2023-02-07T00:07:23Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "tlsconnectionmanager.amazonaws.com",
    "userAgent": "tlsconnectionmanager.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:acm:arn": "arn:aws:acm:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "aws:vpc-lattice:arn": "arn:aws:vpc-lattice:us-west-2:111122223333:service/
svc-0b23c1234567890ab"
        },
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "sharedEventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "eventCategory": "Management"
}
```


Amazon VPC Lattice용 Identity and Access Management

다음 섹션에서는 VPC Lattice API 작업을 수행할 수 있는 사용자를 제어하여 AWS Identity and Access Management (IAM)를 사용하여 VPC Lattice 리소스를 보호하는 방법을 설명합니다.

주제

- Amazon VPC Lattice가 IAM과 작동하는 방식
- Amazon VPC Lattice API 권한
- Amazon VPC Lattice의 자격 증명 기반 정책
- Amazon VPC Lattice에 서비스 연결 역할 사용
- AWS Amazon VPC Lattice에 대한 관리형 정책

Amazon VPC Lattice가 IAM과 작동하는 방식

IAM을 사용하여 VPC Lattice에 대한 액세스를 관리하기 전에 VPC Lattice에서 사용할 수 있는 IAM 기능을 알아봅니다.

IAM 특성	VPC Lattice 지원
자격 증명 기반 정책	예
리소스 기반 정책	예
정책 작업	예
정책 리소스	예
정책 조건 키	예
ACL	아니요
ABAC(정책의 태그)	예
임시 보안 인증	예
서비스 역할	아니요
서비스 연결 역할	예

VPC Lattice 및 기타 AWS 서비스가 대부분의 IAM 기능과 작동하는 방식을 전체적으로 알아보려면 IAM 사용 설명서의 AWS IAM으로 작업하는 서비스를 참조하세요.

VPC Lattice에 대한 자격 증명 기반 IAM 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. JSON 정책에서 사용할 수 있는 모든 요소에 대해 알아보려면 IAM 사용 설명서의 IAM JSON 정책 요소 참조를 참조하세요.

VPC Lattice 내 리소스 기반 정책

리소스 기반 정책 지원: 예

리소스 기반 정책은의 리소스에 연결하는 JSON 정책 문서입니다 AWS. 리소스 기반 정책을 지원하는 AWS 서비스에서 서비스 관리자는 이를 사용하여 해당 AWS 서비스의 특정 리소스에 대한 액세스를 제어할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다.

VPC Lattice는 서비스 네트워크의 서비스에 대한 액세스를 제어할 수 있는 리소스 기반 정책인 인증 정책을 지원합니다. 자세한 내용은 인증 정책을 사용하여 VPC Lattice 서비스에 대한 액세스 제어 단원을 참조하십시오.

VPC Lattice는 AWS Resource Access Manager에 통합하기 위한 리소스 기반 권한 정책도 지원합니다. 이러한 리소스 기반 정책을 사용하여 서비스, 리소스 구성 및 서비스 네트워크에 대해 다른 AWS 계정 또는 조직에 대한 연결을 관리할 수 있는 권한을 부여할 수 있습니다. 자세한 내용은 VPC Lattice 엔터티 공유 단원을 참조하십시오.

VPC Lattice에 대한 정책 조치

정책 작업 지원: 예

IAM 정책 설명에는 IAM을 지원하는 모든 서비스의 모든 API 작업을 지정할 수 있습니다. VPC Lattice의 경우 API 작업인 `vpc-lattice:`에 다음 접두사를 사용합니다. 예를 들어, `vpc-lattice:CreateService`, `vpc-lattice:CreateTargetGroup` 및 `vpc-lattice:PutAuthPolicy`입니다.

명령문 하나에 여러 태스크를 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [ "vpc-lattice:action1", "vpc-lattice:action2" ]
```

와일드카드를 사용하여 여러 작업을 지정할 수도 있습니다. 예를 들어 다음과 같이 이름이 Get으로 시작되는 모든 작업을 지정할 수 있습니다.

```
"Action": "vpc-lattice:Get*"
```

VPC Lattice API 작업 목록을 보려면 서비스 권한 부여 참조에서 Amazon VPC Lattice에서 정의한 작업을 참조하세요.

VPC Lattice를 위한 정책 리소스

정책 리소스 지원: 예

IAM 정책 구문에서 Resource 요소는 명령문이 다루는 하나 이상의 객체를 지정합니다. VPC Lattice의 경우 각 IAM 정책문은 ARN을 사용하여 지정한 리소스에 적용됩니다.

특정 Amazon 리소스 이름(ARN) 형식은 리소스에 따라 다릅니다. ARN을 제공할 때 #### 텍스트를 리소스별 정보로 바꿉니다.

- 액세스 로그 구독:

  ```
  "Resource": "arn:aws:vpc-lattice:region:account-id:accesslogsubscription/access-log-subscription-id"
  ```

- 리스너:

  ```
  "Resource": "arn:aws:vpc-lattice:region:account-id:service/service-id/
  listener/listener-id"
  ```

- 리소스 게이트웨이

  ```
  "Resource": "arn:aws:vpc-lattice:region:account-id:resourcegateway/resource-gateway-id"
  ```

- 리소스 구성

  ```
  "Resource": "arn:aws:vpc-lattice:region:account-id:resourceconfiguration/resource-configuration-id"
  ```

- 규칙:

```
"Resource": "arn:aws:vpc-lattice:region:account-id:service/service-id/
listener/listener-id/rule/rule-id"
```

- 서비스:

```
"Resource": "arn:aws:vpc-lattice:region:account-id:service/service-id"
```

- 서비스 네트워크:

```
"Resource": "arn:aws:vpc-lattice:region:account-id:servicenetwork/service-network-id"
```

- 서비스 네트워크 서비스 연결:

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkserviceassociation/service-network-service-association-id"
```

- 서비스 네트워크 리소스 구성 연결

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkresourceassociation/service-network-resource-association-id"
```

- 서비스 네트워크 VPC 연결:

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkvpcassociation/service-network-vpc-association-id"
```

- 대상 그룹:

```
"Resource": "arn:aws:vpc-lattice:region:account-id:targetgroup/target-group-id"
```

VPC Lattice에 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` 요소는 정의된 기준에 따라 문이 실행되는 시기를 지정합니다. 같음(equals) 또는 미만(less than)과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수

있습니다. 모든 AWS 전역 조건 키를 보려면 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

VPC Lattice 조건 키 목록을 보려면 서비스 승인 참조의 Amazon VPC Lattice에 사용되는 조건 키를 참조하세요.

AWS 는 전역 조건 키와 서비스별 조건 키를 지원합니다. AWS 전역 조건 키에 대한 자세한 내용은 IAM 사용 설명서의 AWS 전역 조건 컨텍스트 키를 참조하세요.

VPC Lattice의 액세스 제어 목록(ACL)

ACL 지원: 아니요

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.

VPC Lattice에서 ABAC(속성 기반 액세스 제어)

ABAC 지원(정책의 태그): 예

속성 기반 액세스 제어(ABAC)는 태그라고 불리는 속성을 기반으로 권한을 정의하는 권한 부여 전략입니다. IAM 엔터티 및 AWS 리소스에 태그를 연결한 다음 보안 주체의 태그가 리소스의 태그와 일치할 때 작업을 허용하는 ABAC 정책을 설계할 수 있습니다.

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.

VPC Lattice에서 임시 보안 인증 사용

임시 자격 증명 지원: 예

임시 자격 증명은 AWS 리소스에 대한 단기 액세스를 제공하며 페더레이션 또는 전환 역할을 사용할 때 자동으로 생성됩니다. 장기 액세스 키를 사용하는 대신 임시 자격 증명을 동적으로 생성하는 것이

AWS 좋습니다. 자세한 내용은 IAM 사용 설명서의 IAM의 임시 보안 자격 증명 및 IAM으로 작업하는 AWS 서비스 섹션을 참조하세요.

VPC Lattice의 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 내용은 IAM 사용 설명서의 AWS 서비스 AWS에 권한을 위임할 역할 생성을 참조하세요.

> **Warning**
>
> 서비스 역할에 대한 권한을 변경하면 VPC Lattice 기능이 중단될 수 있습니다. VPC Lattice에서 관련 지침을 제공하는 경우에만 서비스 역할을 편집합니다.

VPC Lattice의 서비스 연결 역할

서비스 연결 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 연결 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.

VPC Lattice 서비스 연결 역할을 생성하거나 관리하는 방법에 대한 자세한 내용은 Amazon VPC Lattice에 서비스 연결 역할 사용 단원을 참조하세요.

Amazon VPC Lattice API 권한

VPC Lattice에 대한 정책 조치에 설명된 대로 필요한 VPC Lattice API 작업을 호출할 수 있는 IAM 자격 증명(예: 사용자 또는 역할)을 부여해야 합니다. 또한 일부 VPC Lattice 작업의 경우 다른 AWS APIs에서 특정 작업을 호출할 수 있는 IAM 자격 증명 권한을 부여해야 합니다.

API에 필요한 권한

API에서 다음 작업을 호출할 때 지정된 작업을 호출할 수 있는 IAM 사용자 권한을 부여해야 합니다.

`CreateResourceConfiguration`

- `vpc-lattice:CreateResourceConfiguration`

- ec2:DescribeSubnets
- rds:DescribeDBInstances
- rds:DescribeDBClusters

CreateResourceGateway

- vpc-lattice:CreateResourceGateway
- ec2:AssignPrivateIpAddresses
- ec2:AssignIpv6Addresses
- ec2:CreateNetworkInterface
- ec2:CreateNetworkInterfacePermission
- ec2:DeleteNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets

DeleteResourceGateway

- vpc-lattice:DeleteResourceGateway
- ec2:DeleteNetworkInterface

UpdateResourceGateway

- vpc-lattice:UpdateResourceGateway
- ec2:AssignPrivateIpAddresses
- ec2:AssignIpv6Addresses
- ec2:UnassignPrivateIpAddresses
- ec2:CreateNetworkInterface
- ec2:CreateNetworkInterfacePermission
- ec2:DeleteNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:ModifyNetworkInterfaceAttribute

CreateServiceNetworkResourceAssociation

- vpc-lattice:CreateServiceNetworkResourceAssociation

- ec2:AssignIpv6Addresses
- ec2:CreateNetworkInterface
- ec2:CreateNetworkInterfacePermission
- ec2:DescribeNetworkInterfaces

CreateServiceNetworkVpcAssociation

- vpc-lattice:CreateServiceNetworkVpcAssociation
- ec2:DescribeVpcs
- ec2:DescribeSecurityGroups(보안 그룹이 제공된 경우에만 필요)

UpdateServiceNetworkVpcAssociation

- vpc-lattice:UpdateServiceNetworkVpcAssociation
- ec2:DescribeSecurityGroups(보안 그룹이 제공된 경우에만 필요)

CreateTargetGroup

- vpc-lattice:CreateTargetGroup
- ec2:DescribeVpcs

RegisterTargets

- vpc-lattice:RegisterTargets
- ec2:DescribeInstances(대상 그룹 유형이 INSTANCE인 경우에만 필요)
- ec2:DescribeVpcs(대상 그룹 유형이 INSTANCE 또는 IP인 경우에만 필요)
- ec2:DescribeSubnets(대상 그룹 유형이 INSTANCE 또는 IP인 경우에만 필요)
- lambda:GetFunction(대상 그룹 유형이 LAMBDA인 경우에만 필요)
- lambda:AddPermission(대상 그룹에 지정된 Lambda 함수를 호출할 권한이 아직 없는 경우에만 필요)

DeregisterTargets

- vpc-lattice:DeregisterTargets

CreateAccessLogSubscription

- vpc-lattice:CreateAccessLogSubscription
- logs:GetLogDelivery
- logs:CreateLogDelivery

DeleteAccessLogSubscription

- vpc-lattice:DeleteAccessLogSubscription

- `logs:DeleteLogDelivery`

`UpdateAccessLogSubscription`

- `vpc-lattice:UpdateAccessLogSubscription`
- `logs:UpdateLogDelivery`

Amazon VPC Lattice의 자격 증명 기반 정책

기본적으로 사용자 및 역할은 Amazon VPC Lattice 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다.

이러한 예제 JSON 정책 문서를 사용하여 IAM ID 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성(콘솔)을 참조하세요.

각 리소스 유형에 대한 ARN 형식을 포함하여 VPC Lattice에서 정의한 작업 및 리소스 유형에 대한 자세한 내용은 서비스 인증 참조에서 Amazon VPC Lattice에 대한 작업, 리소스 및 조건 키를 참조하세요.

내용

- 정책 모범 사례
- 전체 액세스에 필요한 추가 권한
- VPC Lattice에 대한 자격 증명 기반 정책 예시

정책 모범 사례

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

- AWS 관리형 정책을 시작하고 최소 권한으로 전환 - 사용자 및 워크로드에 권한 부여를 시작하려면 많은 일반적인 사용 사례에 대한 권한을 부여하는 AWS 관리형 정책을 사용합니다. 에서 사용할 수 있습니다 AWS 계정. 사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 추가로 줄이는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우, 작업을 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있

는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.

- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어, SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. AWS 서비스와 같은 특정를 통해 사용되는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 있습니다 CloudFormation. 자세한 내용은 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 확인하여 안전하고 기능적인 권한 보장 - IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 새로운 및 기존 정책을 확인합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 내용은 IAM 사용 설명서의 IAM Access Analyzer에서 정책 검증을 참조하세요.
- 다중 인증(MFA) 필요 -에서 IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 MFA를 AWS 계정켭니다. API 작업을 직접적으로 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 내용은 IAM 사용 설명서의 MFA를 통한 보안 API 액세스를 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

전체 액세스에 필요한 추가 권한

VPC Lattice가 통합된 다른 AWS 서비스와 전체 VPC Lattice 기능 제품군을 사용하려면 특정 추가 권한이 있어야 합니다. 혼동된 대리자 권한 에스컬레이션 위험 때문에 이러한 권한은 `VPCLatticeFullAccess` 관리형 정책에 포함되지 않습니다.

다음 정책을 역할에 연결하고 `VPCLatticeFullAccess` 관리형 정책과 함께 사용해야 합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream",
                "lambda:AddPermission",
                "s3:PutBucketPolicy"
            ],
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutResourcePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "vpc-lattice.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/vpc-
lattice.amazonaws.com/AWSServiceRoleForVpcLattice"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery*"
        }
    ]
}
```

이 정책은 다음과 같은 추가 권한을 제공합니다.

- `iam:AttachRolePolicy`: 지정된 관리형 정책을 지정된 IAM 역할에 연결하도록 허용합니다.
- `iam:PutRolePolicy`: 지정된 IAM 역할에 포함된 인라인 정책 문서를 추가 또는 업데이트하도록 허용합니다.

- `s3:PutBucketPolicy`: Amazon S3 버킷에 버킷 정책을 적용하도록 허용합니다.
- `firehose:TagDeliveryStream`: Firehose 전송 스트림에 태그를 추가 또는 업데이트하도록 허용합니다.

VPC Lattice에 대한 자격 증명 기반 정책 예시

주제

- 예제 정책: 서비스 네트워크에 대한 VPC 연결 관리
- 예제 정책: 서비스 네트워크에 대한 서비스 연결 생성
- 예제 정책: 리소스에 태그 추가
- 예제 정책: 서비스 연결 역할 생성

예제 정책: 서비스 네트워크에 대한 VPC 연결 관리

다음 예시는 이 정책의 사용자에게 서비스 네트워크에 대한 VPC 연결을 생성, 업데이트, 삭제할 수 있는 권한을 부여하는 정책을 보여줍니다. 단, 조건에 지정된 VPC와 서비스 네트워크에 한합니다. 조건 키 지정에 대한 자세한 내용은 VPC Lattice에 사용되는 정책 조건 키 섹션을 참조하세요.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "vpc-lattice:CreateServiceNetworkVpcAssociation",
            "vpc-lattice:UpdateServiceNetworkVpcAssociation",
            "vpc-lattice:DeleteServiceNetworkVpcAssociation"
         ],
         "Resource": [
            "*"
         ],
         "Condition": {
            "StringEquals": {
               "vpc-lattice:ServiceNetworkArn": "arn:aws:vpc-lattice:us-
west-2:123456789012:servicenetwork/sn-903004f88example",
               "vpc-lattice:VpcId": "vpc-1a2b3c4d"
```

```
            }
         }
      }
   ]
}
```

예제 정책: 서비스 네트워크에 대한 서비스 연결 생성

조건 키를 사용하여 VPC Lattice 리소스에 대한 액세스를 제어하지 않는 경우 `Resource` 요소에 리소스의 ARN을 지정하여 대신 액세스를 제어할 수 있습니다.

다음 예시는 이 정책의 사용자가 `CreateServiceNetworkServiceAssociation` API 작업에 사용할 수 있는 서비스 및 서비스 네트워크의 ARN을 지정하여 서비스 연결을 서비스 네트워크로 제한하는 정책을 보여줍니다. ARN 값을 지정하는 방법에 대한 자세한 내용은 VPC Lattice를 위한 정책 리소스 섹션을 참조하세요.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "vpc-lattice:CreateServiceNetworkServiceAssociation"
         ],
         "Resource": [
            "arn:aws:vpc-lattice:us-
west-2:123456789012:servicenetworkserviceassociation/*",
            "arn:aws:vpc-lattice:us-west-2:123456789012:service/
svc-04d5cc9b88example",
            "arn:aws:vpc-lattice:us-west-2:123456789012:servicenetwork/
sn-903004f88example"
         ]
      }
   ]
}
```

예제 정책: 리소스에 태그 추가

다음 예시는 이 정책의 사용자에게 VPC Lattice 리소스에 태그를 생성할 수 있는 권한을 부여하는 정책을 보여줍니다.

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "vpc-lattice:TagResource"
         ],
         "Resource": "arn:aws:vpc-lattice:us-west-2:123456789012:*/*"
      }
   ]
}
```

예제 정책: 서비스 연결 역할 생성

VPC Lattice에는의 사용자가 VPC Lattice 리소스를 처음 생성할 때 서비스 연결 역할을 AWS 계정 생성할 수 있는 권한이 필요합니다. 서비스 연결 역할이 아직 존재하지 않으면 VPC Lattice에서 해당 역할을 계정에 생성합니다. 서비스 연결 역할은 사용자를 대신하여 다른를 호출할 수 있도록 VPC Lattice AWS 서비스 에 권한을 부여합니다. 자세한 내용은 the section called “서비스 연결 역할 사용” 단원을 참조하십시오.

역할 자동 생성이 성공하려면 사용자가 `iam:CreateServiceLinkedRole` 작업에 대한 권한을 보유해야 합니다.

```
"Action": "iam:CreateServiceLinkedRole"
```

다음 예시는 이 정책의 사용자에게 VPC Lattice에 대한 서비스 연결 역할을 생성할 수 있는 권한을 부여하는 정책을 보여줍니다.

JSON

```
{
```

```
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "iam:CreateServiceLinkedRole",
         "Resource": "arn:aws:iam::*:role/aws-service-role/vpc-
lattice.amazonaws.com/AWSServiceRoleForVpcLattice",
         "Condition": {
            "StringLike": {
               "iam:AWSServiceName":"vpc-lattice.amazonaws.com"
            }
         }
      }
   ]
}
```

자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 권한을 참조하세요.

Amazon VPC Lattice에 서비스 연결 역할 사용

Amazon VPC Lattice는 사용자를 대신하여 다른를 호출하는 데 필요한 권한 AWS 서비스 에 서비스 연결 역할을 사용합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할을 참조하세요.

VPC Lattice는 라는 서비스 연결 역할을 사용합니다AWSServiceRoleForVpcLattice.

VPC Lattice의 서비스 연결 역할 권한

AWSServiceRoleForVpcLattice 서비스 연결 역할은 다음 서비스를 신뢰하여 역할을 위임합니다.

- `vpc-lattice.amazonaws.com`

AWSVpcLatticeServiceRolePolicy라는 이름이 지정된 역할 권한 정책은 VPC Lattice가 `AWS/VpcLattice` 네임스페이스에 CloudWatch 지표를 게시할 수 있도록 허용합니다. 자세한 내용은 관리형 정책 참조AWSVpcLatticeServiceRolePolicy의 섹션을 참조하세요. AWS

IAM 엔터티(사용자, 그룹, 역할 등)가 서비스 연결 역할을 생성하고 편집하거나 삭제할 수 있도록 권한을 구성할 수 있습니다. 자세한 내용은 the section called “예제 정책: 서비스 연결 역할 생성” 단원을 참조하십시오.

VPC Lattice에 대한 서비스 연결 역할 생성

서비스 연결 역할은 수동으로 생성할 필요가 없습니다. AWS Management Console AWS CLI, 또는 AWS API에서 VPC Lattice 리소스를 생성하면 VPC Lattice가 서비스 연결 역할을 생성합니다.

이 서비스 연결 역할을 삭제했다가 다시 생성해야 하는 경우 동일한 프로세스를 사용하여 계정에서 역할을 다시 생성할 수 있습니다. VPC Lattice 리소스를 생성하면 VPC Lattice에서 서비스 연결 역할이 다시 생성됩니다.

VPC Lattice에 대한 서비스 연결 역할 편집

IAM을 사용하여 AWSServiceRoleForVpcLattice의 설명을 편집할 수 있습니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 설명 편집을 참조하세요.

VPC Lattice에 대한 서비스 연결 역할 삭제

Amazon VPC Lattice를 더 이상 사용할 필요가 없다면 AWSServiceRoleForVpcLattice를 삭제하는 것이 좋습니다.

이 서비스 연결 역할은 AWS 계정에서 모든 VPC Lattice 연결을 삭제한 후에만 삭제할 수 있습니다.

IAM 콘솔 AWS CLI, 또는 AWS API를 사용하여 AWSServiceRoleForVpcLattice 서비스 연결 역할을 삭제합니다. 자세한 내용은 IAM 사용 설명서의 서비스 연결 역할 삭제를 참조하세요.

서비스 연결 역할을 삭제한 후 AWS 계정에서 VPC Lattice 리소스를 생성하면 VPC Lattice에서 역할이 다시 생성됩니다.

VPC Lattice 서비스 연결 역할에 대해 지원되는 리전

VPC Lattice에서는 서비스를 사용할 수 있는 모든 리전에서 서비스 연결 역할 사용을 지원합니다.

AWS Amazon VPC Lattice에 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 줍니다. AWS AWS 서비스 는 새가 시작되거나 기존 서비스에 새 API 작업을 사용할 수 있게 될 때 AWS 관리형 정책을 업데이트할 가능성이 높습니다.

자세한 내용은 IAM 사용자 가이드의 AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: VPCLatticeFullAccess

이 정책은 Amazon VPC Lattice에 대한 전체 액세스 권한과 다른 종속 서비스에 대한 제한된 액세스 권한을 제공합니다. 이 정책에는 다음을 수행할 권한이 포함되어 있습니다.

- ACM – 사용자 지정 도메인 이름에 대한 SSL/TLS 인증서 ARN을 검색합니다.
- CloudWatch – 액세스 로그와 모니터링 데이터를 확인합니다.
- CloudWatch Logs – CloudWatch Logs에 액세스 로그를 설정하고 전송합니다.
- Amazon EC2 - 네트워크 인터페이스를 구성하고 EC2 인스턴스 및 VPCs 대한 정보를 검색합니다. 이는 리소스 구성, 리소스 게이트웨이 및 대상 그룹을 생성하고, VPC Lattice 엔터티 연결을 구성하고, 대상을 등록하는 데 사용됩니다.
- Elastic Load Balancing – Application Load Balancer에 대한 정보를 검색하여 대상으로 등록합니다.
- Firehose - 액세스 로그를 저장하는 데 사용되는 전송 스트림에 대한 정보를 검색합니다.
- Lambda – Lambda 함수에 대한 정보를 검색하여 대상으로 등록합니다.
- Amazon RDS - RDS 클러스터 및 인스턴스에 대한 정보를 검색합니다.
- Amazon S3 – 액세스 로그를 저장하는 데 사용되는 S3 버킷에 대한 정보를 검색합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조에서 VPCLatticeFullAccess를 참조하세요.

VPC Lattice가 통합된 다른 AWS 서비스와 전체 VPC Lattice 기능 제품군을 사용하려면 특정 추가 권한이 있어야 합니다. 혼동된 대리자 권한 에스컬레이션 위험 때문에 이러한 권한은 `VPCLatticeFullAccess` 관리형 정책에 포함되지 않습니다. 자세한 내용은 전체 액세스에 필요한 추가 권한 단원을 참조하십시오.

AWS 관리형 정책: VPCLatticeReadOnlyAccess

이 정책은 Amazon VPC Lattice에 대한 읽기 전용 액세스 권한과 기타 종속 서비스에 대한 제한된 액세스 권한을 제공합니다. 이 정책에는 다음을 수행할 권한이 포함되어 있습니다.

- ACM – 사용자 지정 도메인 이름에 대한 SSL/TLS 인증서 ARN을 검색합니다.
- CloudWatch – 액세스 로그와 모니터링 데이터를 확인합니다.

- CloudWatch Logs — 액세스 로그 구독에 대한 로그 전송 정보를 확인합니다.
- Amazon EC2 – EC2 인스턴스 및 VPC에 대한 정보를 검색하여 대상 그룹을 생성하고 대상을 등록합니다.
- Elastic Load Balancing – Application Load Balancer에 대한 정보를 검색합니다.
- Firehose - 액세스 로그 전송을 위한 전송 스트림에 대한 정보를 검색합니다.
- Lambda – Lambda 함수에 대한 정보를 확인합니다.
- Amazon RDS - RDS 클러스터 및 인스턴스에 대한 정보를 검색합니다.
- Amazon S3 – 액세스 로그 전송을 위해 S3 버킷에 대한 정보를 검색합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조에서 AmazonVPCFullAccess를 참조하세요.

AWS 관리형 정책: VPCLatticeServicesInvokeAccess

이 정책은 Amazon VPC Lattice 서비스를 호출할 수 있는 액세스 권한을 제공합니다.

이 정책의 권한을 보려면 AWS 관리형 정책 참조에서 VPCLatticeServicesInvokeAccess를 참조하세요.

AWS 관리형 정책: AWSVpcLatticeServiceRolePolicy

이 정책은 AWSServiceRoleForVpcLattice이라는 서비스 연결 역할에 연결되어 VPC Lattice가 사용자를 대신하여 작업을 수행할 수 있도록 합니다. IAM 엔터티에 이 정책을 연결할 수 없습니다. 자세한 내용은 Amazon VPC Lattice에 서비스 연결 역할 사용 단원을 참조하십시오.

이 정책의 권한을 보려면 AWS 관리형 정책 참조의 AWSVpcLatticeServiceRolePolicy를 참조하세요.

AWS 관리형 정책에 대한 VPC Lattice 업데이트

이 서비스가 이러한 변경 사항을 추적하기 시작한 이후부터 VPC Lattice의 AWS 관리형 정책 업데이트에 대한 세부 정보를 봅니다. 이 페이지의 변경 사항에 대한 자동 알림을 받아보려면 VPC Lattice 사용 설명서에 대한 RSS 피드를 구독하세요.

변경	설명	Date
VPCLatticeFullAccess	VPC Lattice는 Amazon RDS 클러스터 및 인스턴스를 설명하는 읽기 전용 권한을 추가합니다.	2024년 12월 1일

변경	설명	Date
VPCLatticeReadOnlyAccess	VPC Lattice는 Amazon RDS 클러스터 및 인스턴스를 설명하는 읽기 전용 권한을 추가합니다.	2024년 12월 1일
AWSVpcLatticeServiceRolePolicy	VPC Lattice는 VPC Lattice가 요청자 관리형 네트워크 인터페이스를 생성할 수 있는 권한을 추가합니다.	2024년 12월 1일
VPCLatticeFullAccess	VPC Lattice는 Amazon VPC Lattice에 대한 전체 액세스 권한과 다른 종속 서비스에 대한 제한된 액세스를 허용하는 새 정책을 추가합니다.	2023년 3월 31일
VPCLatticeReadOnlyAccess	VPC Lattice는 Amazon VPC Lattice에 대한 읽기 전용 액세스 권한과 다른 종속 서비스에 대한 제한된 액세스 권한을 부여하는 새 정책을 추가합니다.	2023년 3월 31일
VPCLatticeServicesInvokeAccess	VPC Lattice는 Amazon VPC Lattice 서비스를 호출할 수 있는 액세스 권한을 부여하는 새 정책을 추가합니다.	2023년 3월 31일
AWSVpcLatticeServiceRolePolicy	VPC Lattice는 서비스 연결 역할에 권한을 추가하여 VPC Lattice가 AWS/VpcLattice 네임스페이스에 CloudWatch 지표를 게시할 수 있도록 합니다. 이 AWSVpcLatticeServiceRolePolicy 정책에 CloudWatch PutMetricData API 작업을 호출할 수 있는 권한이 포함됩니다. 자세한 내용은 Amazon VPC Lattice에 서비스 연결 역할 사용 단원을 참조하십시오.	2022년 12월 5일
VPC Lattice에서 변경 내용 추적 시작	VPC Lattice는 AWS 관리형 정책에 대한 변경 사항 추적을 시작했습니다.	2022년 12월 5일

Amazon VPC Lattice의 규정 준수 검증

타사 감사자는 여러 규정 준수 프로그램의 일환으로 Amazon VPC Lattice의 보안 및 AWS 규정 준수를 평가합니다.

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 AWS 서비스 규정 준수 프로그램 범위 내를 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 Artifact을 참조하세요.

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

인터페이스 엔드포인트를 사용하여 Amazon VPC Lattice에 액세스 (AWS PrivateLink)

인터페이스 VPC 엔드포인트를 생성하여 VPC와 Amazon VPC Lattice 간에 프라이빗 연결을 설정할 수 있습니다. 인터페이스 엔드포인트는 인터넷 게이트웨이AWS PrivateLink, NAT 디바이스, VPN 연결 또는 Direct Connect 연결 없이 VPC Lattice APIs에 비공개로 액세스할 수 있는 기술로 구동됩니다. VPC의 인스턴스는 VPC Lattice API와 통신하는 데 퍼블릭 IP 주소가 필요하지 않습니다.

각 인터페이스 엔드포인트는 서브넷에서 하나 이상의 네트워크 인터페이스로 표현됩니다.

인터페이스 VPC 엔드포인트에 대한 고려 사항

VPC Lattice에 대한 인터페이스 VPC 엔드포인트를 설정하기 전에 AWS PrivateLink 가이드의를 AWS 서비스 통한 액세스를 AWS PrivateLink 검토해야 합니다.

VPC Lattice는 VPC에서 모든 API 작업에 대한 호출 수행을 지원합니다.

VPC Lattice에 대한 인터페이스 VPC 엔드포인트 생성

Amazon VPC 콘솔 또는 AWS Command Line Interface ()를 사용하여 VPC Lattice 서비스에 대한 VPC 엔드포인트를 생성할 수 있습니다AWS CLI. 자세한 내용은 AWS PrivateLink 가이드의 인터페이스 VPC 엔드포인트 생성을 참조하세요.

다음 서비스 이름을 사용하여 VPC Lattice에 대한 VPC 엔드포인트를 생성합니다.

`com.amazonaws.region.vpc-lattice`

엔드포인트에 프라이빗 DNS를 사용하도록 설정하는 경우, 리전에 대한 기본 DNS 이름(예: `vpc-lattice.us-east-1.amazonaws.com`)을 사용하여 VPC Lattice에 API 요청을 할 수 있습니다.

Amazon VPC Lattice의 복원성

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 중심으로 구축됩니다.

AWS 리전 는 물리적으로 분리되고 격리된 여러 가용 영역을 제공하며,이 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹과 연결됩니다.

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.

Amazon VPC Lattice의 인프라 보안

관리형 서비스인 Amazon VPC Lattice는 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 환경을 설계하려면 보안 원칙 AWS Well-Architected Framework의 인프라 보호를 참조하세요 AWS .

AWS 에서 게시한 API 호출을 사용하여 네트워크를 통해 VPC Lattice에 액세스합니다. 클라이언트는 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.

Amazon VPC Lattice 모니터링

이 섹션의 기능을 사용하여 Amazon VPC Lattice 서비스 네트워크, 서비스, 대상 그룹 및 VPC 연결을 모니터링할 수 있습니다.

내용

- Amazon VPC Lattice에 대한 CloudWatch 지표
- Amazon VPC Lattice에 대한 액세스 로그
- Amazon VPC Lattice에 대한 CloudTrail 로그

Amazon VPC Lattice에 대한 CloudWatch 지표

Amazon VPC Lattice는 대상 그룹 및 서비스와 관련된 데이터를 Amazon CloudWatch로 보내고 실시간에 가까운 읽기 가능한 지표로 처리합니다. 이러한 지표는 15개월간 보관되므로 기록 정보에 액세스하고 웹 애플리케이션 또는 서비스가 어떻게 실행되고 있는지 전체적으로 더 잘 파악할 수 있습니다. 특정 임계값을 주시하다가 임계값이 충족될 때 알림을 전송하거나 조치를 취하도록 경보를 설정할 수도 있습니다. 자세한 정보는 Amazon CloudWatch 사용 설명서를 참조하십시오.

Amazon VPC Lattice는 AWS 계정의 서비스 연결 역할을 사용하여 지표를 Amazon CloudWatch로 전송합니다. 자세한 내용은 Amazon VPC Lattice에 서비스 연결 역할 사용 단원을 참조하십시오.

내용

- Amazon CloudWatch 지표 보기
- 대상 그룹 지표
- 서비스 지표

Amazon CloudWatch 지표 보기

Amazon CloudWatch 콘솔 또는 AWS CLI를 사용하여 대상 그룹에 대한 CloudWatch 지표를 볼 수 있습니다.

CloudWatch 콘솔을 사용하여 지표를 보는 방법

1. https://console.aws.amazon.com/cloudwatch/에서 Amazon CloudWatch 콘솔을 엽니다.
2. 탐색 창에서 지표를 선택합니다.

3. AWS/VpcLattice 네임스페이스를 선택합니다.
4. (선택 사항) 모든 측정기준의 지표를 보려면 검색 필드에 이름을 입력합니다.
5. (선택 사항) 측정기준을 기준으로 필터링하려면 다음 중 하나를 선택하세요.

   - 대상 그룹에 보고된 지표만 표시하려면 대상 그룹을 선택합니다. 단일 대상 그룹에 대한 지표를 보려면 검색 필드에 해당되는 이름을 입력합니다.
   - 서비스에 보고된 지표만 표시하려면 서비스를 선택합니다. 단일 서비스에 대한 지표를 보려면 검색 필드에 해당되는 이름을 입력합니다.

를 사용하여 지표를 보려면 AWS CLI

다음 CloudWatch list-metrics AWS CLI 명령을 사용하여 사용 가능한 지표를 나열합니다.

```
aws cloudwatch list-metrics --namespace AWS/VpcLattice
```

지표와 그 측정기준 각각에 대한 자세한 내용은 대상 그룹 지표 및 서비스 지표 단원을 참조하세요.

대상 그룹 지표

VPC Lattice는 대상 그룹과 관련된 지표를 AWS/VpcLattice Amazon CloudWatch 네임스페이스에 자동으로 저장합니다. 대상 그룹에 대한 자세한 내용은 VPC Lattice의 대상 그룹 섹션을 참조하세요.

측정 기준

대상 그룹에 대한 지표를 필터링하려면 다음 차원을 사용합니다.

- AvailabilityZone
- TargetGroup

지표	설명	TargetGroup 프로토콜
TotalConnectionCount	총 연결 수. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든).	HTTP, HTTPS, TCP

지표	설명	TargetGroup 프로토콜
	보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	
ActiveConnectionCount	활성 연결 수. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS, TCP

지표	설명	TargetGroup 프로토콜
ConnectionErrorCount	총 연결 실패 수. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS, TCP
HTTP1_ConnectionCount	총 HTTP/1.1 연결 수. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS

지표	설명	TargetGroup 프로토콜
HTTP2_ConnectionCount	총 HTTP/2 연결 수. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS
ConnectionTimeoutCount	총 연결 제한 시간. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS, TCP

지표	설명	TargetGroup 프로토콜
TotalReceivedConnectionBytes	수신된 총 연결 바이트. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS, TCP
TotalSentConnectionBytes	전송된 총 연결 바이트. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS, TCP

지표	설명	TargetGroup 프로토콜
TotalRequestCount	총 요청 수. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS
ActiveRequestCount	총 활성 요청 수. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS

지표	설명	TargetGroup 프로토콜
RequestTime	요청 시간은 밀리초 단위의 마지막 바이트입니다. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Average 및 pNN.NN(백분위수)입니다.	HTTP, HTTPS
HTTPCode_2XX_Count, HTTPCode_3XX_Count, HTTPCode_4XX_Count, HTTPCode_5XX_Count	HTTP 응답 코드를 집계합니다. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS

지표	설명	TargetGroup 프로토콜
TLSConnectionErrorCount	실패한 인증서 확인을 제외한 총 TLS 연결 오류 수. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS, TCP
TotalTLSConnectionHandshakeCount	성공한 총 TLS 연결 핸드셰이크 수. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.	HTTP, HTTPS, TCP

서비스 지표

VPC Lattice는 서비스와 관련된 지표를 AWS/VpcLattice Amazon CloudWatch 네임스페이스에 자동으로 저장합니다. 서비스에 대한 자세한 내용은 VPC Lattice 내 서비스 섹션을 참조하세요.

측정 기준

대상 그룹에 대한 지표를 필터링하려면 다음 차원을 사용합니다.

- AvailabilityZone
- Service

지표	설명
RequestTimeoutCount	응답 대기 시간이 초과된 총 요청 수. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고됩니다(0 또는 0이 아닌 값). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.
TotalRequestCount	총 요청 수. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번.

지표	설명
	통계 • 가장 유용한 통계는 Sum입니다.
RequestTime	요청 시간(밀리초). 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Average 및 pNN.NN(백분위수)입니다.
HTTPCode_2XX_Count, HTTPCode_3XX_Count, HTTPCode_4XX_Count, HTTPCode_5XX_Count	HTTP 응답 코드를 집계합니다. 보고 기준 • 리소스가 트래픽을 수신한 시점부터 항상 보고합니다(값이 0이든 아니든). 보고 빈도 • 1분에 한 번. 통계 • 가장 유용한 통계는 Sum입니다.

Amazon VPC Lattice에 대한 액세스 로그

액세스 로그는 VPC Lattice 서비스 및 리소스 구성에 대한 자세한 정보를 캡처합니다. 이 액세스 로그를 사용하여 트래픽 패턴을 분석하고 네트워크의 모든 서비스를 감사할 수 있습니다. VPC Lattice 서비스의 경우를 게시VpcLatticeAccessLogs하고 리소스 구성의 경우 별도로 구성해야 VpcLatticeResourceAccessLogs 하는를 게시합니다.

액세스 로그는 선택 사항이며 기본적으로 비활성화됩니다. 액세스 로그를 활성화한 후에는 언제든지 비활성화할 수 있습니다.

가격 책정

액세스 로그가 게시되면 요금이 부과됩니다. 사용자를 대신하여 AWS 기본적으로 게시하는 로그를 벤딩 로그라고 합니다. 벤딩 로그 요금에 대한 자세한 내용을 알려면 Amazon CloudWatch 요금을 참조하고 로그를 선택한 다음 벤딩 로그에서 요금을 확인하세요.

내용

액세스를 활성화하는 데 필요한 IAM 권한

액세스 로그를 활성화하고 대상으로 로그를 전송하려면 사용 중인 IAM 사용자, 그룹 또는 역할에 연결된 정책에 다음 작업이 있어야 합니다.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Sid": "ManageVPCLatticeAccessLogSetup",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "vpc-lattice:CreateAccessLogSubscription",
                "vpc-lattice:GetAccessLogSubscription",
                "vpc-lattice:UpdateAccessLogSubscription",
                "vpc-lattice:DeleteAccessLogSubscription",
                "vpc-lattice:ListAccessLogSubscriptions"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

자세한 내용은 AWS Identity and Access Management 사용 설명서의 IAM 자격 증명 권한 추가 및 제거 섹션을 참조하세요.

사용 중인 IAM 사용자, 그룹 또는 역할에 연결된 정책을 업데이트한 후 액세스 로그 활성화로 이동하세요.

액세스 로그 대상

액세스 로그를 다음과 같은 대상으로 보낼 수 있습니다.

Amazon CloudWatch Logs

- VPC Lattice는 일반적으로 2분 이내에 로그를 CloudWatch Logs에 전송합니다. 하지만 실제 로그 전송 시간은 최선의 노력을 바탕으로 하며 추가 지연 시간이 발생할 수 있습니다.
- 로그 그룹에 특정 권한이 없는 경우 리소스 정책이 자동으로 생성되어 CloudWatch 로그 그룹에 추가됩니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 CloudWatch Logs로 전송된 로그를 참조하세요.

- CloudWatch 콘솔의 로그 그룹에서 CloudWatch로 전송된 액세스 로그를 확인할 수 있습니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 CloudWatch Logs로 전송된 데이터 보기를 참조하세요.

Amazon S3

- VPC Lattice는 일반적으로 6분 이내에 Amazon S3에 로그를 전송합니다. 하지만 실제 로그 전송 시간은 최선의 노력을 바탕으로 하며 추가 지연 시간이 발생할 수 있습니다.
- 버킷에 특정 권한이 없는 경우 버킷 정책이 자동으로 생성되어 Amazon S3 버킷에 추가됩니다. 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 Amazon S3으로 전송된 로그를 참조하세요.
- Amazon S3으로 전송되는 액세스 로그는 다음과 같은 명명 규칙을 사용합니다.

```
[bucket]/[prefix]/AWSLogs/[accountId]/VpcLattice/AccessLogs/[region]/[YYYY/
MM/DD]/[resource-id]/[accountId]_VpcLatticeAccessLogs_[region]_[resource-
id]_YYYYMMDDTHHmmZ_[hash].json.gz
```

- Amazon S3로 전송되는 VpcLatticeResourceAccessLogs는 다음 명명 규칙을 사용합니다.

```
[bucket]/[prefix]/AWSLogs/[accountId]/VpcLattice/ResourceAccessLogs/[region]/[YYYY/
MM/DD]/[resource-id]/[accountId]_VpcLatticeResourceAccessLogs_[region]_[resource-
id]_YYYYMMDDTHHmmZ_[hash].json.gz
```

Amazon Data Firehose

- VPC Lattice는 일반적으로 2분 이내에 Firehose에 로그를 전송합니다. 하지만 실제 로그 전송 시간은 최선의 노력을 바탕으로 하며 추가 지연 시간이 발생할 수 있습니다.
- Amazon Data Firehose에 액세스 로그를 전송할 권한을 VPC Lattice에 부여하는 서비스 연결 역할이 자동으로 생성됩니다. 역할 자동 생성이 성공하려면 사용자가 `iam:CreateServiceLinkedRole` 작업에 대한 권한을 보유해야 합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서의 Amazon Data Firehose으로 전송된 로그를 참조하세요.
- Amazon Data Firehose로 전송된 로그 보기에 대한 자세한 내용은 Amazon Data Firehose 개발자 안내서의 Amazon Kinesis Data Streams를 참조하세요.

액세스 로그 활성화

액세스 로그를 캡처하고 선택한 대상으로 액세스 로그를 전송하도록 다음과 같은 절차에 따라 액세스 로그를 구성하세요.

내용

콘솔을 사용하여 액세스 로그 활성화

생성 중에 서비스 네트워크, 서비스 또는 리소스 구성에 대한 액세스 로그를 활성화할 수 있습니다. 다음 절차에 설명된 대로 서비스 네트워크, 서비스 또는 리소스 구성을 생성한 후 액세스 로그를 활성화할 수도 있습니다.

콘솔을 사용하여 기본 서비스를 생성하는 방법

1. https://console.aws.amazon.com/vpc/에서 Amazon VPC 콘솔을 엽니다.
2. 서비스 네트워크, 서비스 또는 리소스 구성을 선택합니다.
3. 작업, VPC 설정 편집을 선택합니다.
4. 액세스 로그 토글 스위치를 켭니다.
5. 다음과 같이 액세스 로그의 전송 대상을 추가합니다.
    - CloudWatch 로그 그룹을 선택하고 로그 그룹을 선택합니다. 로그 그룹을 생성하려면 CloudWatch에서 로그 그룹 생성을 선택합니다.
    - S3 버킷을 선택하고 접두사를 포함한 S3 버킷 경로를 입력합니다. S3 버킷을 검색하려면 S3 찾아보기를 선택합니다.
    - Kinesis Data Firehose 전송 스트림을 클릭하고 전송 스트림을 선택합니다. 전송 스트림을 생성하려면 Kinesis에서 전송 스트림 생성을 선택합니다.
6. 변경 사항 저장을 선택합니다.

를 사용하여 액세스 로그 활성화 AWS CLI

CLI 명령인 create-access-log-subscription을 사용하여 서비스 네트워크 또는 서비스에 대한 액세스 로그를 활성화합니다.

요청 추적

VPC Lattice는 x-amzn-requestid 헤더를 사용한 관찰성 및 디버깅을 위해 클라이언트, 대상 및 로그 간의 요청 추적 및 상관 관계를 지원합니다. 이 헤더는 클라이언트에서 설정 및 전송하거나 VPC Lattice에서 생성할 수 있으며 대상으로 전송되고 액세스 로그에서도 사용할 수 있습니다.

기본 동작

- VPC Lattice는 모든 요청에 대해이 헤더를 자동으로 생성합니다.
- 값은 임의로 생성된 식별자(기본 UUID 스타일)입니다.
- 생성된 식별자는 다음과 같습니다.
  - 다운스트림 대상으로 전파됩니다.
  - 클라이언트에 대한 응답 헤더에 반환됩니다.
  - 액세스 로그에 로그인됨

예제(기본 응답)

다음은 valu eof x-amzn-requestid 헤더에 대한 임의 값을 생성하는 VPC Lattice의 기본 동작으로 클라이언트에 전송된 응답의 예입니다.

```
{
    "HTTP/1.1 200 OK
    x-amzn-requestid: a9f2c7a1-6b4f-4c79-9e87-ff5a1234a001"
}
```

클라이언트에서 값 설정

- 클라이언트는 수신 요청에이 헤더를 선택적으로 설정하여 자동으로 생성된 값을 재정의할 수 있습니다.
- 고려 사항
  - 헤더 값은 UUID 형식을 따를 필요가 없습니다.
  - 헤더 값이 512바이트를 초과하면 VPC Lattice는 헤더 값을 512로 자릅니다.

- 성공적으로 재정의되면 제공된 헤더 값은 다음을 수행합니다.
  - 응답 헤더에 표시
  - 대상에 전파

- 액세스 로그 및 지표에 표시

예제(클라이언트 요청 재정의)

다음은 헤더 값을 사용하여 클라이언트가 보낸 요청의 예입니다.

```
{
    "GET /my-service/endpoint HTTP/1.1
    Host: my-api.example.com
    x-amzn-requestid: trace-request-foobar"
}
```

예제(기본 재정의 응답)

다음은 재정의된 값을 사용하여 클라이언트에 전송된 응답의 예입니다.

```
{
    "HTTP/1.1 200 OK
    x-amzn-requestid: trace-request-foobar"
}
```

액세스 로그 내용

다음 표에서는 액세스 로그 항목의 필드를 설명합니다.

필드	설명	형식
`callerPrincipalTags`	요청의 PrincipalTags입니다.	JSON
`hostHeader`	요청의 권한 헤더.	문자열
`sslCipher`	클라이언트 TLS 연결을 설정하는 데 사용되는 암호 집합의 OpenSSL 이름.	문자열
`serviceNetworkArn`	서비스 네트워크 ARN.	arn:aws:vpc-lattice:*##*:*##*:servicenetwork/*id*
`resolvedUser`	인증이 활성화되고 완료되었을 때의 사용자 ARN.	null | ARN | "익명" | "알 수 없음"

필드	설명	형식
authDeniedReason	인증이 활성화된 경우 액세스가 거부되는 이유.	null | "서비스" | "네트워크" | "자격 증명"
requestMethod	요청의 메서드 헤더.	문자열
targetGroupArn	대상 호스트가 속하는 대상 호스트 그룹.	문자열
tlsVersion	TLS 버전.	TLSv*x*
userAgent	사용자-에이전트 헤더.	문자열
serverNameIndication	[HTTPS만 해당] SSL 연결 소켓에 SNI(서버 이름 표시)에 대해 설정된 값.	문자열
destinationVpcId	대상 VPC ID.	vpc-*xxxxxxxx*
sourceIpPort	클라이언트의 IP 주소 및 소스 포트.	*ip*:##
targetIpPort	클라이언트의 IP 주소 및 대상 포트.	*ip*:##
serviceArn	서비스 ARN.	arn:aws:vpc-lattice:##:## :service/*id*
sourceVpcId	소스 VPC.	vpc-*xxxxxxxx*
requestPath	요청의 경로.	LatticePath?:##
startTime	요청 시작 시간.	*YYYY*-*MM*-*DD*T*HH*:*MM*:*SS*Z
protocol	프로토콜. 현재 HTTP/1.1 또는 HTTP/2.	문자열

필드	설명	형식
responseCode	HTTP 응답 코드입니다. 최종 헤더의 응답 코드만 기록됩니다. 자세한 내용은 액세스 로그 문제 해결 단원을 참조하십시오.	정수
bytesReceived	받은 본문 및 헤더 바이트.	정수
bytesSent	보낸 본문 및 헤더 바이트.	정수
duration	시작 시간부터 마지막 바이트 출력까지의 총 요청 시간(밀리초).	정수
requestToTargetDuration	시작 시간부터 대상으로 전송된 마지막 바이트까지의 총 요청 시간(밀리초).	정수
responseFromTargetDuration	대상 호스트에서 읽은 첫 번째 바이트부터 클라이언트로 전송된 마지막 바이트까지의 총 요청 시간(밀리초).	정수
grpcResponseCode	gRPC 응답 코드. 자세한 내용은 gRPC에서의 상태 코드 및 사용을 참조하세요. 서비스에서 gRPC를 지원하는 경우에만 이 필드를 기록합니다.	정수
requestId	이 고유 식별자는 응답에 x-amzn-requestid 헤더의 값으로 자동으로 포함됩니다. 관찰성 및 디버깅을 위해 클라이언트, 대상 및 로그 간의 요청 상관관계를 활성화합니다.	문자열

필드	설명	형식
callerPrincipal	인증된 보안 주체.	문자열
callerX509SubjectCN	주체 이름(CN).	문자열
callerX509IssuerOU	발급자(OU).	문자열
callerX509SANNameCN	발급자 대체(이름/CN).	문자열
callerX509SANDNS	주체 대체 이름(DNS).	문자열
callerX509SANURI	주체 대체 이름(URI).	문자열
sourceVpcArn	요청이 시작된 VPC의 ARN.	arn:aws:ec2:##:##:vpc/*id*

필드	설명	형식
failureReason	요청이 실패한 이유를 나타냅니다. 가능한 값은 다음과 같습니다. • TargetConnectionError - 요청이 대상 그룹의 대상에 연결하지 못했습니다. • TargetProtocolError - 대상이 유효한 데이터로 응답하지 않았습니다. 이는 대상에 잘못된 TLS 레코드가 있거나 잘못된 대상 그룹 프로토콜을 사용했음을 나타낼 수 있습니다. • TargetDataTimeout - 유휴 제한 시간에 도달했습니다. • TargetConnectionClosed - 대상은 응답을 완료하기 전에 연결을 종료했습니다. • ClientConnectionClosed - 클라이언트가 완전한 응답을 받기 전에 연결을 종료했습니다. • ClientRateLimited - 클라이언트가 연결 제한을 초과했고 VPC Lattice가 속도를 제한했습니다. • ClientAccessDenied - VPC Lattice가 리소스에 대한 액세스를 거부했습니	문자열

필드	설명	형식
	다. VPC Lattice가 액세스를 거부한 이유에 대한 자세한 내용은 authDeniedReason 를 사용합니다. • ClientProtocolError - 클라이언트가 이해되지 않은 데이터를 전송했습니다. 이는 클라이언트가 잘못된 TLS 레코드 또는 잘못된 프로토콜을 사용했음을 나타낼 수 있습니다. • ConnectionDurationExceeded - 연결이 최대 연결 기간 제한에 도달했습니다. • InternalError - 요청을 처리하는 동안 내부 오류가 발생했습니다.	

예제

다음은 로그 항목의 예시입니다.

```
{
    "callerPrincipalTags" : "{ "TagA": "ValA", "TagB": "ValB", ... }",
    "hostHeader": "example.com",
    "sslCipher": "-",
    "serviceNetworkArn": "arn:aws:vpc-lattice:us-west-2:123456789012:servicenetwork/
svn-1a2b3c4d",
    "resolvedUser": "Unknown",
    "authDeniedReason": "null",
    "requestMethod": "GET",
    "targetGroupArn": "arn:aws:vpc-lattice:us-west-2:123456789012:targetgroup/
tg-1a2b3c4d",
    "tlsVersion": "-",
    "userAgent": "-",
```

```
    "serverNameIndication": "-",
    "destinationVpcId": "vpc-0abcdef1234567890",
    "sourceIpPort": "178.0.181.150:80",
    "targetIpPort": "131.31.44.176:80",
    "serviceArn": "arn:aws:vpc-lattice:us-west-2:123456789012:service/svc-1a2b3c4d",
    "sourceVpcId": "vpc-0abcdef1234567890",
    "requestPath": "/billing",
    "startTime": "2023-07-28T20:48:45Z",
    "protocol": "HTTP/1.1",
    "responseCode": 200,
    "bytesReceived": 42,
    "bytesSent": 42,
    "duration": 375,
    "requestToTargetDuration": 1,
    "responseFromTargetDuration": 1,
    "grpcResponseCode": 1,
    "requestId": "a9f2c7a1-6b4f-4c79-9e87-ff5a1234a001"
}
```


리소스 액세스 로그 콘텐츠

다음 표에서는 리소스 액세스 로그 항목의 필드를 설명합니다.

필드	설명	형식
`serviceNetworkArn`	서비스 네트워크 ARN.	arn:*partition* vpc-lattice:*region*:*account*:servicenetwork/*id*
`serviceNetworkResourceAssociationId`	서비스 네트워크 리소스 ID입니다.	*snra-xxx*
`vpcEndpointId`	리소스에 액세스하는 데 사용된 엔드포인트 ID입니다.	문자열
`sourceVpcArn`	연결이 시작된 소스 VPC ARN 또는 VPC입니다.	문자열
`resourceConfigurationArn`	액세스한 리소스 구성의 ARN입니다.	문자열

필드	설명	형식
protocol	리소스 구성과 통신하는 데 사용되는 프로토콜입니다. 현재 tcp만 지원됩니다.	문자열
sourceIpPort	연결을 시작한 소스의 IP 주소 및 포트입니다.	*ip*:##
destinationIpPort	연결이 시작된 IP 주소 및 포트입니다. SN-E/SN-A의 IP가 됩니다.	*ip*:##
gatewayIpPort	리소스 게이트웨이가 리소스에 액세스하는 데 사용하는 IP 주소 및 포트입니다.	*ip*:##
resourceIpPort	리소스의 IP 주소 및 포트입니다.	*ip*:##

예제

다음은 로그 항목의 예시입니다.

```
{
    "eventTimestamp": "2024-12-02T10:10:10.123Z",
    "serviceNetworkArn": "arn:aws:vpc-lattice:us-west-2:1234567890:servicenetwork/
sn-1a2b3c4d",
    "serviceNetworkResourceAssociationId": "snra-1a2b3c4d",
    "vpcEndpointId": "vpce-01a2b3c4d",
    "sourceVpcArn": "arn:aws:ec2:us-west-2:1234567890:vpc/vpc-01a2b3c4d",
    "resourceConfigurationArn": "arn:aws:vpc-lattice:us-
west-2:0987654321:resourceconfiguration/rcfg-01a2b3c4d",
    "protocol": "tcp",
    "sourceIpPort": "172.31.23.56:44076",
    "destinationIpPort": "172.31.31.226:80",
    "gatewayIpPort": "10.0.28.57:49288",
    "resourceIpPort": "10.0.18.190:80"
}
```

액세스 로그 문제 해결

이 섹션에는 액세스 로그에 표시될 수 있는 HTTP 오류 코드에 대한 설명이 포함되어 있습니다.

오류 코드	가능한 원인
HTTP 400: 잘못된 요청	• 클라이언트가 HTTP 사양을 충족하지 않는 잘못된 형식의 요청을 전송했습니다. • 전체 요청 헤더 또는 100개 이상의 헤더에 대해 요청 헤더가 60K를 초과했습니다. • 클라이언트가 전체 요청 본문을 보내기 전에 연결을 종료했습니다.
HTTP 403: 금지됨	서비스에 대한 인증이 구성되었지만 수신되는 요청이 인증되거나 승인되지 않았습니다.
HTTP 404: 존재하지 않는 서비스	사용자가 존재하지 않거나 올바른 서비스 네트워크에 등록되지 않은 서비스에 연결을 시도하고 있습니다.
HTTP 500: 내부 서버 오류	VPC Lattice에서 대상에 연결하지 못하는 등의 오류가 발생했습니다.
HTTP 502: 잘못된 게이트웨이	VPC Lattice에 오류가 발생했습니다.

Amazon VPC Lattice에 대한 CloudTrail 로그

Amazon VPC Lattice는 사용자AWS CloudTrail, 역할 또는가 수행한 작업에 대한 레코드를 제공하는 서비스인와 통합됩니다 AWS 서비스. CloudTrail은 VPC Lattice에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 VPC Lattice 콘솔의 호출과 VPC Lattice API 작업에 대한 코드 호출이 포함됩니다. CloudTrail에서 수집한 정보를 사용하여 VPC Lattice에 수행된 요청, 요청이 수행된 IP 주소, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 관한 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 사용자의 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. 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. 계정의 모든 AWS 리전 에서 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 AWS 계정에 대한 추적 생성 및 조직에 대한 추적 생성을 참조하세요.

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

추가 작업을 모니터링하려면 액세스 로그를 사용하세요. 자세한 내용은 액세스 로그 단원을 참조하십시오.

CloudTrail의 VPC Lattice 관리 이벤트

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

Amazon VPC Lattice는 VPC Lattice 컨트롤 플레인 작업을 관리 이벤트로 로깅합니다. VPC Lattice가 CloudTrail에 로깅하는 Amazon VPC Lattice 컨트롤 플레인 작업 목록은 Amazon VPC Lattice API 참조를 참조하세요.

VPC Lattice 이벤트 예제

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

다음 예제에서는 CreateService 작업에 대한 CloudTrail 이벤트를 보여줍니다.


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "abcdef01234567890",
    "arn": "arn:abcdef01234567890",
    "accountId": "abcdef01234567890",
    "accessKeyId": "abcdef01234567890",
    "sessionContext": {
        "sessionIssuer": {
            "type": "Role",
            "principalId": "abcdef01234567890",
            "arn": "arn:abcdef01234567890",
            "accountId": "abcdef01234567890",
            "userName": "abcdef01234567890"
        },
        "webIdFederationData": {},
        "attributes": {
```

```
            "creationDate": "2022-08-16T03:34:54Z",
            "mfaAuthenticated": "false"
        }
    }
  },
  "eventTime": "2022-08-16T03:36:12Z",
  "eventSource": "vpc-lattice.amazonaws.com",
  "eventName": "CreateService",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "abcdef01234567890",
  "userAgent": "abcdef01234567890",
  "requestParameters": {
    "name": "rates-service"
  },
  "responseElements": {
    "name": "rates-service",
    "id": "abcdef01234567890",
    "arn": "arn:abcdef01234567890",
    "status": "CREATE_IN_PROGRESS"
  },
  "requestID": "abcdef01234567890",
  "eventID": "abcdef01234567890",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "abcdef01234567890",
  "eventCategory": "Management"
}
```

다음 예제에서는 DeleteService 작업에 대한 CloudTrail 이벤트를 보여줍니다.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "abcdef01234567890",
    "arn": "arn:ABCXYZ123456",
    "accountId": "abcdef01234567890",
    "accessKeyId": "abcdef01234567890",
    "sessionContext": {
        "sessionIssuer": {
            "type": "Role",
            "principalId": "abcdef01234567890",
```

```
            "arn": "arn:aws:iam::AIDACKCEVSQ6C2EXAMPLE:role/Admin",
            "accountId": "abcdef01234567890",
            "userName": "Admin"
        },
        "webIdFederationData": {},
        "attributes": {
            "creationDate": "2022-10-27T17:42:36Z",
            "mfaAuthenticated": "false"
        }
    }
  },
  "eventTime": "2022-10-27T17:56:41Z",
  "eventSource": "vpc-lattice.amazonaws.com",
  "eventName": "DeleteService",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "72.21.198.64",
  "userAgent": "abcdef01234567890",
  "requestParameters": {
    "serviceIdentifier": "abcdef01234567890"
  },
  "responseElements": {
    "name": "test",
    "id": "abcdef01234567890",
    "arn": "arn:abcdef01234567890",
    "status": "DELETE_IN_PROGRESS"
  },
  "requestID": "abcdef01234567890",
  "eventID": "abcdef01234567890",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "abcdef01234567890",
  "eventCategory": "Management"
}
```


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

Amazon VPC Lattice의 할당량

AWS 계정 에는 각에 대해 이전에 제한이라고 하는 기본 할당량이 있습니다 AWS 서비스. 다르게 표시되지 않는 한, 리전별로 각 할당량이 적용됩니다. 일부 할당량에 대한 증가를 요청할 수 있으며 다른 할당량은 늘릴 수 없습니다.

VPC Lattice에 대한 할당량을 보려면 Service Quotas 콘솔을 엽니다. 탐색 창에서 AWS 서비스를 선택하고 VPC Lattice를 선택합니다.

할당량 증가를 요청하려면 Service Quotas 사용 설명서의 할당량 증가 요청을 참조하세요.

AWS 계정 에는 VPC Lattice와 관련된 다음과 같은 할당량이 있습니다.

이름	기본값	조정 가능	설명
인증 정책 크기	지원되는 각 리전: 10킬로바이트	아니요	인증 정책에 사용되는 최대 JSON 파일 크기.
그룹 리소스 구성당 하위 리소스 구성	지원되는 각 리전: 60개	예	그룹 리소스 구성에서 최대 하위 리소스 구성 수입니다. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.
AWS 리전당 도메인 확인	지원되는 각 리전: 5개	예	계정당 생성할 수 있는 최대 도메인 확인 수입니다. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.
서비스당 리스너	지원되는 각 리전: 2개	예	서비스에 생성할 수 있는 최대 리스너 수. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.

이름	기본값	조정 가능	설명
서비스 네트워크당 리소스 구성	지원되는 각 리전: 500	예	서비스 네트워크와 연결된 최대 리소스 구성 수입니다. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.
AWS 리전당 리소스 구성	지원되는 각 리전: 2,000	예	AWS 계정이 AWS 리전당 가질 수 있는 최대 리소스 구성 수입니다. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.
VPC당 리소스 게이트웨이	지원되는 각 리전: 500	예	VPC의 최대 리소스 게이트웨이 수입니다. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.
리스너당 규칙	il-central-1: 5 각각의 지원되는 다른 리전: 10	예	서비스 리스너에 정의할 수 있는 최대 규칙 수. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.
연결당 보안 그룹	지원되는 각 리전: 5	아니요	VPC와 서비스 네트워크 간의 연결에 추가할 수 있는 최대 보안 그룹의 수.
서비스 네트워크당 서비스 연결	지원되는 각 리전: 500개	예	단일 서비스 네트워크에 연결할 수 있는 최대 서비스 수. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.

이름	기본값	조정 가능	설명
리전당 서비스 네트워크	il-central-1: 10 각각의 지원되는 다른 리전: 50	예	리전당 최대 서비스 네트워크 수. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.
리전당 서비스	il-central-1: 500 각각의 지원되는 다른 리전: 2,000	예	리전당 서비스의 최대 수. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.
리전당 대상 그룹	지원되는 각 리전: 500개	예	리전당 최대 대상 그룹 수. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.
서비스당 대상 그룹	il-central-1: 5 각각의 지원되는 다른 리전: 10	예	서비스에 연결할 수 있는 최대 대상 그룹 수. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.
대상 그룹당 대상	지원되는 각 리전: 1,000개	예	단일 대상 그룹과 연결할 수 있는 최대 대상 수. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.
서비스 네트워크당 VPC 연결	지원되는 각 리전: 500개	예	단일 서비스 네트워크에 연결할 수 있는 최대 VPC 수. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.

이름	기본값	조정 가능	설명
서비스 네트워크당 서비스 네트워크 유형의 VPC 엔드포인트	지원되는 각 리전: 200	예	서비스 네트워크와 연결된 서비스 네트워크 엔드포인트의 최대 수입니다. 추가 용량 및 한도 증가는 AWS Support에 문의하세요.

VPC Lattice에는 , use1-az3, , usw1-az2, apne1-az3, apne2-az2, euc1-az2, 가용 영역이 지원되지 않습니다euw1-az4cac1-az3ilc1-az2.

다음과 같은 제한이 적용됩니다.

제한	값	설명
가용 영역별 서비스당 대역폭	10Gbps	가용 영역별 서비스당 할당된 기본 대역폭입니다. 이를 늘릴 수 있습니다. 추가 지원을 받으려면 Solutions Architect(SA) 또는 Technical Account Manager(TAM)에 문의하십시오.
연결당 최대 전송 단위(MTU)	8500바이트	서비스가 수락할 수 있는 가장 큰 데이터 패킷의 크기입니다.
가용 영역별 서비스당 초당 요청 수	10,000	HTTP 서비스의 경우 가용 영역당 서비스당 초당 기본 요청 수입니다. 이를 늘릴 수 있습니다. 추가 지원을 받으려면 Solutions Architect(SA) 또는 Technical Account Manager(TAM)에 문의하십시오.
VPC Lattice 서비스의 연결당 연결 유휴 시간	1분	VPC Lattice 서비스에 대한 활성 요청(HTTP 및 GRPC의 경우) 또는 활성 데이터 전송(TLS-PASSTHROUGH의

제한	값	설명
		경우) 없이 연결이 유휴 상태로 유지될 수 있는 기본 시간입니다. HTTP 및 애플리케이션 수준 keepalive를 사용하여이 유휴 제한 시간을 최대 연결 수명 기간까지 연장할 수 있습니다. 이를 늘릴 수 있습니다. 추가 지원을 받으려면 Solutions Architect(SA) 또는 Technical Account Manager(TAM)에 문의하십시오.
VPC Lattice 서비스의 연결당 최대 연결 수명	10분	VPC Lattice 서비스를 위해 클라이언트와 서버 간에 연결을 열 수 있는 최대 시간입니다.
VPC Lattice 리소스에 대한 연결당 최대 연결 수명	NA	VPC Lattice는 리소스에 수명 연결 제한을 부과하지 않습니다. 클라이언트와 서버는 VPC Lattice 리소스의 유휴 제한 시간을 인식하면서 수명 연결 기간을 결정합니다.이 제한 시간은 350초입니다.
VPC Lattice 리소스에 대한 연결당 연결 유휴 시간	350초	TCP keepalive를 사용하여이 유휴 제한 시간을 연장할 수 있습니다.
VPC당 서비스 네트워크	서비스 네트워크 1개	연결을 통해 VPC를 하나의 서비스 네트워크에만 연결할 수 있습니다. VPC를 여러 서비스 네트워크에 연결하려면 서비스 네트워크 유형의 VPC 엔드포인트를 사용할 수 있습니다.

Amazon VPC Lattice 사용 설명서의 문서 기록

다음 표에서는 VPC Lattice에 대한 문서 릴리스를 설명합니다.

변경 사항	설명	날짜
리소스 게이트웨이에 구성 가능한 IP 주소 추가	VPC Lattice는 이제 리소스 게이트웨이에 대해 구성 가능한 IP 주소를 지원합니다.	2025년 10월 7일
에 대한 VPC Lattice 추가 Oracle Database@AWS	용 VPC Lattice가 Oracle Database@AWS 릴리스되었습니다.	2025년 6월 26일
관리 엔드포인트에 대한 듀얼 스택 지원 추가	VPC Lattice는 이제 모든 VPC Lattice 관리 APIs에 대해 듀얼 스택(IPv4 및 IPv6) 엔드포인트를 지원합니다.	2025년 4월 30일
리소스 공유 및 액세스	이제 VPC Lattice는 VPC 및 계정 경계에서 리소스 공유 및 액세스를 지원합니다. 여기에는 VPCLatticeReadOnlyAccess 및 VPCLatticeFullAccess 정책에 대한 업데이트가 포함됩니다.	2024년 12월 1일
TLS 패스스루	VPC Lattice는 이제 TLS 패스스루를 지원하므로 end-to-end 인증을 위해 TLS 종료를 수행할 수 있습니다.	2024년 5월 14일
Lambda 이벤트 구조 버전	VPC Lattice는 이제 Lambda 이벤트 구조의 새 버전을 지원합니다.	2023년 9월 7일

공유 VPC 관련 지원	참여자는 공유 VPC에서 VPC Lattice 대상 그룹을 생성할 수 있습니다.	2023년 7월 5일
정식 출시 릴리스	정식 출시(GA)를 위한 VPC Lattice 사용 설명서 릴리스	2023년 3월 31일
이제 VPC Lattice가 AWS 관리형 정책에 대한 변경 사항을 보고합니다.	관리형 정책에 대한 변경 사항은 "보안" 장의 "VPC Lattice에 대한AWS 관리형 정책"에 보고됩니다.	2023년 3월 29일
Application Load Balancer 대상 유형에 대한 지원	VPC Lattice는 이제 Application Load Balancer 유형 대상 그룹 생성을 지원합니다.	2023년 3월 29일
모든 인스턴스 유형 지원	VPC Lattice는 이제 모든 인스턴스 유형을 지원합니다.	2023년 3월 27일
IPv6 지원	VPC Lattice는 이제 IPv4 및 IPv6 IP 대상 그룹을 모두 지원합니다.	2023년 3월 27일
상태 확인을 위한 HTTP2 프로토콜 버전	이제 대상 그룹 프로토콜 버전이 HTTP2인 경우 상태 확인이 지원됩니다.	2023년 3월 27일
리스너 규칙의 고정 응답 작업	VPC Lattice 서비스의 리스너는 이제 전달 작업 외에도 고정 응답 작업을 지원합니다.	2023년 3월 27일
사용자 지정 도메인 이름 지원	이제 VPC Lattice 서비스에 대한 사용자 지정 도메인 이름을 구성할 수 있습니다.	2023년 2월 14일
BYOC(자체 인증서 가져오기) 지원	VPC Lattice는 사용자 지정 도메인 이름에 대해 ACM에서 자체 SSL/TLS 인증서를 사용할 수 있도록 지원합니다.	2023년 2월 14일

VPC Lattice는 이제 지원되지 않는 인스턴스 유형의 업데이트된 목록을 보고합니다.	지원되지 않는 인스턴스 목록에 세 개의 인스턴스가 추가되었습니다.	2023년 1월 26일
이제 VPC Lattice가 AWS 관리형 정책에 대한 변경 사항을 보고합니다.	2022년 12월 5일부터 관리형 정책의 변경 사항은 "보안" 장의 "VPC Lattice에 대한AWS 관리형 정책" 주제에 보고합니다. 나열된 첫 번째 변경 사항은 CloudWatch 모니터링에 필요한 권한을 추가하는 것입니다.	2022년 12월 5일
최초 릴리스	VPC Lattice 사용 설명서의 최초 릴리스	2022년 12월 5일

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