aws

ユーザーガイド

Amazon VPC Lattice

Amazon VPC Lattice: ユーザーガイド

Table of Contents

Amazon VPC Lattice とは

Amazon VPC Lattice は、アプリケーションのサービスおよびリソースの接続、保護、モニタリングに使用する、フルマネージド型アプリケーションネットワーキングサービスです。VPC Lattice は、単一の仮想プライベートクラウド (VPC)、または 1 つ以上のアカウントの複数の VPC で使用できます。

最新のアプリケーションは、HTTP API、データベースなどのリソース、DNS と IP アドレスエンドポイントで構成されるカスタムリソースなど、マイクロサービスと呼ばれることが多い複数の小さなモジュールコンポーネントで構成できます。モダナイゼーションには利点がありますが、これらのマイクロサービスとリソースを接続するときにネットワークの複雑さや課題が生じる可能性もあります。たとえば、開発者が異なるチームに分散している場合、複数のアカウントまたは VPCs。

VPC Lattice では、マイクロサービスをサービスとして参照し、リソース設定としてのみリソースを表します。これらは、VPC Lattice ユーザーガイドで表示され、使用される用語です。

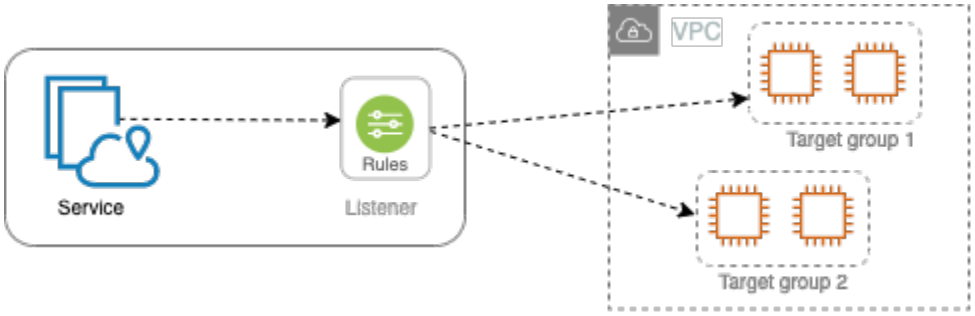
内容

主要コンポーネント

Amazon VPC Lattice を使用するには、その主要コンポーネントに精通している必要があります。

サービス

> 特定のタスクや機能を提供する、独立してデプロイ可能な単位のソフトウェアです。サービスは、アカウントまたは仮想プライベートクラウド (VPC) 内の EC2 インスタンスまたは ECS/EKS/Fargate コンテナで、または Lambda 関数として実行できます。VPC Lattice サービスには、ターゲットグループ、リスナー、ルールというコンポーネントがあります。



ターゲットグループ

アプリケーションまたはサービスを実行するリソース (ターゲットとも呼ばれる) のコレクションです。これらは Elastic Load Balancing が提供するターゲットグループと似ていますが、互換性はありません。サポートされているターゲットタイプには、EC2 インスタンス、IP アドレス、Lambda 関数、Application Load Balancer、Amazon ECS タスク、Kubernetes Pod などがあります。

Listener

接続リクエストをチェックし、ターゲットグループのターゲットにルーティングするプロセスです。リスナーにはプロトコルとポート番号を設定します。

ルール

VPC Lattice ターゲットグループのターゲットにリクエストを転送するリスナーのデフォルトのコンポーネントです。各ルールは優先度、1 つ以上のアクション、および 1 つ以上の条件で構成されています。ルールは、リスナーによるクライアントリクエストのルーティング方法を決定します。

[リソース]

リソースは、Amazon Relational Database Service (Amazon RDS) データベース、Amazon EC2 インスタンス、アプリケーションエンドポイント、ドメイン名ターゲット、IP アドレスなどのエンティティです。 AWS Resource Access Manager (AWS RAM) でリソース共有を作成し、リソースゲートウェイを作成し、リソース設定を定義することで、VPC 内のリソースを共有できます。

リソースゲートウェイ

リソースゲートウェイは、リソースが存在する VPC への進入ポイントです。

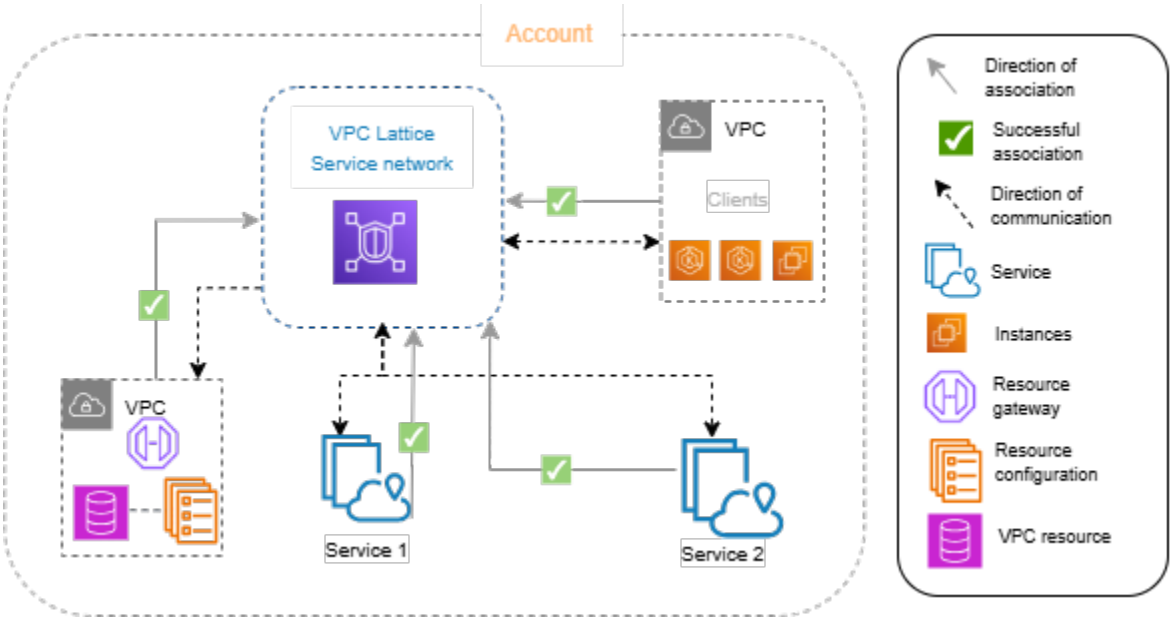
リソース設定

リソース設定は、単一のリソースまたはリソースのグループを表す論理オブジェクトです。リソースには、IP アドレス、ドメイン名ターゲット、または Amazon RDS データベースを使用できます。

サービスネットワーク

サービスとリソース設定の集合の論理境界。クライアントは、サービスネットワークに関連付けられている VPC 内に存在できます。同じサービスネットワークに関連付けられているクライアントとサービスは、権限があれば、相互に通信できます。

次の図の VPC とサービスは同じサービスネットワークに関連付けられているため、クライアントは両方のサービスと通信できます。



サービスディレクトリ

所有している、またはアカウントと共有されているすべての VPC Lattice サービスの中央レジストリ AWS RAM。

認証ポリシー

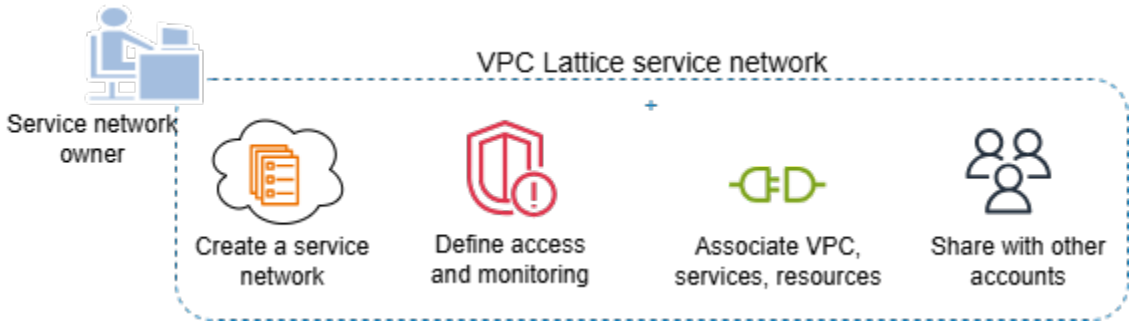
サービスへのアクセスを定義するために使用できる、きめ細やかな認可ポリシーです。個別の認証ポリシーを個々のサービスまたはサービスネットワークにアタッチできます。例えば、EC2 インスタンスの Auto Scaling グループで実行されている支払いサービスと、 AWS Lambdaで実行されている請求サービスとのやり取りを定めたポリシーを作成できます。

認証ポリシーは、リソース設定ではサポートされていません。サービスネットワークの認証ポリシーは、サービスネットワークのリソース設定には適用されません。

役割と責任

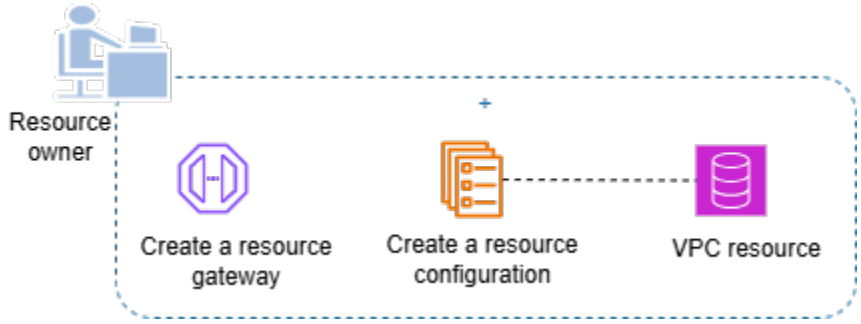
役割は、Amazon VPC Lattice 内の情報の設定とフローの担当者を決定します。通常、サービスネットワーク所有者とサービス所有者の 2 つの役割がありますが、それぞれの責任は重複する場合があります。

サービスネットワーク所有者 — サービスネットワーク所有者は通常、組織のネットワーク管理者またはクラウド管理者です。サービスネットワーク所有者は、サービスネットワークの作成、共有、プロビジョニングを行います。また、VPC Lattice 内のサービスネットワークまたはサービスにアクセスできるユーザーを管理します。サービスネットワーク所有者は、サービスネットワークに関連付けられたサービスの粗粒度のアクセス設定を定義できます。これらのコントロールは、認証ポリシーおよび認可ポリシーを使用してクライアントとサービス間の通信を管理するために使用されます。サービスまたはリソース設定がサービスネットワーク所有者のアカウントと共有されている場合、サービスまたはリソース設定を単一または複数のサービスネットワークに関連付けることもできます。



サービス所有者 – サービス所有者は通常、組織のソフトウェア開発者です。サービス所有者は VPC Lattice でサービスを作成し、ルーティングルールを定義し、サービスをサービスネットワークに関連付けます。また、きめ細かなアクセス設定を定義して、認証および承認されたサービスとクライアントにのみアクセスを制限することもできます。

リソース所有者 – リソース所有者は通常、組織のソフトウェア開発者であり、データベースなどのリソースの管理者として機能します。リソース所有者は、リソースのリソース設定を作成し、リソース設定のアクセス設定を定義し、リソース設定をサービスネットワークに関連付けます。



機能

VPC Lattice が提供するコア機能は以下のとおりです。

サービス検出

サービスネットワークに関連付けられた VPC のクライアントとサービスはすべて、同一サービスネットワーク内の他のサービスと通信できます。DNS は、クライアントからサービスへのトラフィックとサービスからサービスへのトラフィックを VPC Lattice エンドポイントを経由して転送します。クライアントがサービスにリクエストを送信する場合、クライアントはサービスの DNS 名を使用します。Route 53 リゾルバーはトラフィックを VPC Lattice に送信し、VPC Lattice が送信先サービスを識別します。

接続

Client-to-serviceおよびclient-to-resource間の接続は、ネットワークインフラストラクチャ内で確立されます AWS 。VPC をサービスネットワークに関連付けると、VPC 内の任意のクライアントは、必要なアクセス権がある場合、サービスネットワーク内のサービスとリソースに (リソース設定を介して) 接続できます。VPC Lattice は、重複する CIDR テクノロジーをサポートしています。

オンプレミスアクセス

VPC エンドポイント (を使用) を使用して、VPC からサービスネットワークへの接続を有効にできます AWS PrivateLink。タイプのサービスネットワークの VPC エンドポイントを使用すると、Direct Connect と VPN 経由でオンプレミスネットワークからサービスネットワーク内のサービスとリソースへのアクセスを有効にできます。VPC ピアリングを通過するトラフィック、または VPC AWS Transit Gateway エンドポイント経由でリソースやサービスにアクセスできるトラフィック。

オブザーバビリティ

VPC Lattice は、サービスネットワークを経由するリクエストとレスポンスごとにメトリクスとログを生成します。これは、アプリケーションの監視とトラブルシューティングに利用できます。デフォルトでは、メトリクスはサービス所有者アカウントに発行されます。サービス所有者とリソース所有者には、ログ記録を有効にし、サービスおよびリソースへのすべてのクライアントアクセス/リクエストのログを受け取るオプションがあります。サービスネットワーク所有者は、サービスネットワークでログ記録を有効にして、サービスネットワークに接続されている VPCs 内のクライアントからサービスおよびリソースへのすべてのアクセス/リクエストをログに記録することもできます。

VPC Lattice は、ログ Amazon CloudWatch グループ、Firehose 配信ストリーム、Amazon S3 バケットなどのサービスのモニタリングとトラブルシューティングに役立ちます。

セキュリティ

VPC Lattice は、複数のネットワークレイヤーに防御戦略を実装するために使用できるフレームワークを提供します。最初のレイヤーは、サービス、リソース設定、VPC 関連付け、およびサービスネットワークタイプの VPC エンドポイントの組み合わせです。VPC とサービス関連付け、またはサービスネットワークタイプの VPC エンドポイントがないと、クライアントはサービスにアクセスできません。同様に、VPC とリソース設定、サービス関連付け、またはサービスネットワークタイプの VPC エンドポイントがないと、クライアントはリソースにアクセスできません。

2 番目のレイヤーでは、VPC とサービスネットワーク間の関連付けにセキュリティグループをアタッチできます。3 番目と 4 番目のレイヤーは認証ポリシーで、サービスネットワークレベルおよびサービスレベルで個別に適用できます。

VPC Lattice へのアクセス

次のインターフェイスのいずれかを使用して、VPC Lattice の作成、アクセス、管理を行うことができます。

- AWS マネジメントコンソール - VPC Lattice へのアクセスに使用するウェブインターフェイスを提供します。
- AWS Command Line Interface (AWS CLI) – VPC Lattice を含む幅広い AWS サービスのコマンドを提供します。 AWS CLI は Windows、MacOS、Linux でサポートされています。CLI の詳細については、AWS Command Line Interface を参照してください。API の詳細については、「Amazon VPC Lattice API リファレンス」を参照してください。

- Kubernetes 用 VPC Lattice コントローラー — Kubernetes クラスターの VPC Lattice リソースを管理します。Kubernetes での VPC Lattice の使用の詳細については、「AWS ゲートウェイ API コントローラのユーザーガイド」を参照してください。
- CloudFormation - AWS のリソースをモデル化して設定するのに役立ちます。詳細については、「Amazon VPC Lattice リソースタイプリファレンス」を参照してください。

VPC Lattice サービスエンドポイント

エンドポイントは、 AWS ウェブサービスのエントリポイントとして機能する URL です。VPC Lattice は、次のエンドポイントタイプをサポートしています。

- the section called “IPv4 エンドポイント”
- デュアルスタックのエンドポイント (IPv4 と IPv6 の両方をサポート)

リクエストを行うと、使用するエンドポイントを指定できます。エンドポイントを指定しない場合、デフォルトで IPv4 エンドポイントが使用されます。別のエンドポイントタイプを使用するには、リクエストで指定する必要があります。これを行う方法の例については、「the section called “エンドポイントの指定”」を参照してください。使用可能なエンドポイントの表については、「Amazon VPC Lattice エンドポイント」を参照してください。

IPv4 エンドポイント

IPv4 エンドポイントは IPv4 トラフィックのみをサポートします。IPv4 エンドポイントは、すべてのリージョンで利用できます。

一般的なエンドポイントである `vpc-lattice.amazonaws.com` を指定する場合は、`us-east-1` のエンドポイントを使用します。別のリージョンを使用するには、関連するエンドポイントを指定します。例えば、`vpc-lattice.us-east-2.amazonaws.com` をエンドポイントとして指定した場合、リクエストは `us-east-2` エンドポイントに転送されます。

IPv4 エンドポイント名では、次の命名規則が使用されます。

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

例えば、`eu-west-1` リージョンの IPv4 エンドポイントは、`vpc-lattice.eu-west-1.amazonaws.com` です。

デュアルスタック (IPv4 および IPv6) エンドポイント

デュアルスタックエンドポイントは、IPv4 と IPv6 トラフィックの両方をサポートします。デュアルスタックエンドポイントは、すべてのリージョンで使用できます。デュアルスタックエンドポイントにリクエストを行うと、エンドポイント URL は、ネットワークとクライアントが使用するプロトコルに応じて IPv6 または IPv4 アドレスに解決されます。

デュアルスタックエンドポイント名には、次の命名規則が使用されます。

- `vpc-lattice.region.api.aws`

例えば、`eu-west-1` リージョンのデュアルスタックエンドポイント名は、`vpc-lattice.eu-west-1.api.aws` です。

エンドポイントの指定

次の例は、 の を使用して `us-east-2`リージョンのエンドポイントを指定する方法を示しています AWS CLI `vpc-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
```

- デュアルスタック

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

料金

VPC Lattice では、サービスがプロビジョニングされた時間、各サービスを通じて転送されたデータ量、リクエスト数に対してお支払いいただきます。リソース所有者は、各リソースとの間で転送されたデータに対して料金を支払います。サービスネットワーク所有者は、サービスネットワークに関連付けられたリソース設定に対して時間単位で料金を支払います。サービスネットワークに関連付けられた VPC を持つコンシューマーは、VPC からサービスネットワーク内のリソースとの間で転送されたデータに対して料金が発生します。詳細については、「Amazon VPC Lattice の料金」を参照してください。

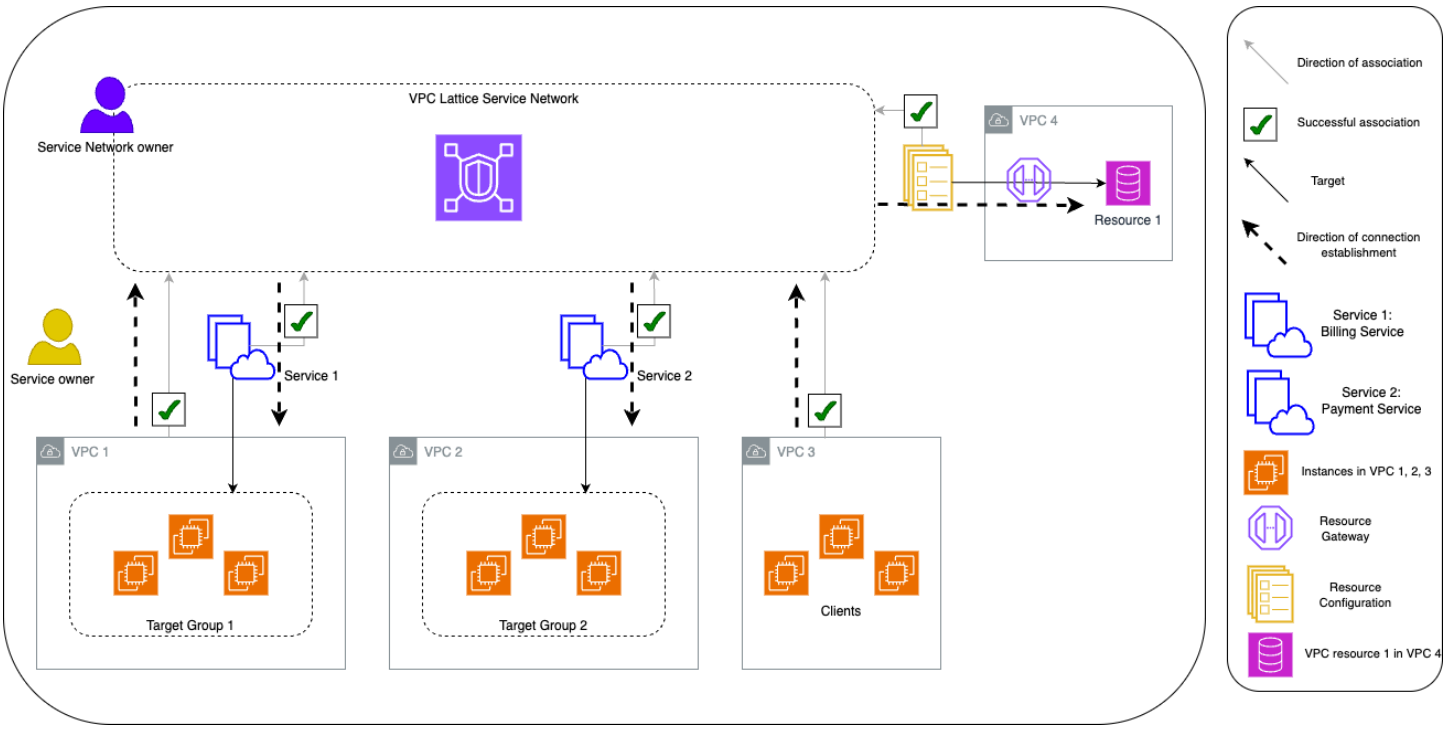
VPC Lattice の仕組み

VPC Lattice は、その中のすべてのサービスとリソースを簡単かつ効果的に検出、保護、接続、モニタリングできるように設計されています。VPC Lattice 内のそれぞれのコンポーネントは、サービスネットワークとの関連付けとアクセス設定に基づいて、サービスネットワーク内で単方向または双方向に通信します。アクセス設定は、この通信に必要な認証ポリシーおよび認可ポリシーで構成されます。

次の概要では、VPC Lattice 内のコンポーネント間の通信について説明します。

- VPC をサービスネットワークに接続するには、VPC の関連付けと タイプのサービスネットワークの VPC エンドポイントの 2 つの方法があります。
- サービスネットワークに関連付けられているサービスとリソースは、VPCs がサービスネットワークにも接続されているクライアントからリクエストを受信できます。
- クライアントは、同じサービスネットワークに接続されている VPC 内にある場合にのみ、サービスネットワークに関連付けられたサービスとリソースにリクエストを送信できます。VPC ピアリング接続、トランジットゲートウェイ、Direct Connect、または VPN を通過するクライアントトラフィックは、VPC が VPC エンドポイントを介してサービスネットワークに接続されている場合にのみ、リソースとサービスに到達できます。
- サービスネットワークに関連付けられている VPCs 内のサービスのターゲットはクライアントでもあり、サービスネットワークに関連付けられた他のサービスやリソースにリクエストを送信できます。
- サービスネットワークに関連付けられていない VPCs 内のサービスのターゲットはクライアントではなく、サービスネットワークに関連付けられた他のサービスやリソースにリクエストを送信することはできません。
- リソースがあるが、VPC がサービスネットワークに関連付けられていない VPCs 内のクライアントはクライアントではなく、サービスネットワークに関連付けられた他のサービスやリソースにリクエストを送信できません。

次のフロー図では、サンプルのシナリオを使用して、VPC Lattice 内のコンポーネント間の情報の流れと通信の方向を説明しています。2 つのサービスがサービスネットワークに関連付けられています。サービスとすべての VPCsの両方が、サービスネットワークと同じアカウントで作成されました。どちらのサービスも、サービスネットワークからのトラフィックを許可するように設定されています。



サービス 1 は、VPC 1 のターゲットグループ 1 に登録されたインスタンスグループで実行される課金アプリケーションです。サービス 2 は、VPC 2 のターゲットグループ 2 に登録されたインスタンスグループで実行される支払いアプリケーションです。VPC 3 は同じアカウントにあり、クライアントはありますが、サービスはありません。リソース 1 は、VPC 4 に顧客データがあるデータベースです。

次のリストは、VPC Lattice の一般的なタスクのワークフローを順番に説明しています。

1. サービスネットワークを作成する

   サービスネットワーク所有者がサービスネットワークを作成します。

2. [Create a service] (サービスを作成)

   サービス所有者が、それぞれのサービス (サービス 1 とサービス 2) を作成します。作成の際には、サービス所有者はリスナーを追加し、各サービスのターゲットグループにリクエストをルーティングするためのルールを定義します。

3. ルーティングを定義する

   サービス所有者が、各サービス (ターゲットグループ 1 とターゲットグループ 2) のターゲットグループを作成します。これを行うには、サービスを実行するターゲットインスタンスを指定します。また、これらのターゲットが存在する VPC を指定します。

上の図では、実線の矢印は、トラフィックをターゲットグループにルーティングするサービスと、リソースにルーティングするリソース設定を表します。

4. サービスをサービスネットワークに関連付ける

   サービスネットワーク所有者またはサービス所有者は、サービスをサービスネットワークに関連付けます。関連付けは、サービスからサービスネットワークを指すチェックマーク付きの矢印で表示されます。サービスをサービスネットワークに関連付けると、そのサービスはサービスネットワークに関連付けられた他のサービスと、サービスネットワークに接続されている VPCs 内のクライアントに検出可能になります。

   サービスネットワークとターゲットグループ間の破線の矢印は、接続確立の方向を示しています。サービスネットワークを使用してトラフィックフローをクライアントに返します。返されるトラフィックを表す矢印は、この図には含まれていません。

5. リソースゲートウェイの作成

   リソース所有者は、クライアントからリソース 1 への接続を有効にするために、VPC 4 にリソースゲートウェイを作成します。

6. リソース設定を作成する

   リソース所有者は、リソース 1 を表すリソース設定を作成し、リソース 1 のリソースゲートウェイを指定します。

7. リソース設定をサービスネットワークに関連付ける

   サービスネットワーク所有者またはリソース所有者は、リソース設定をサービスネットワークに関連付けます。関連付けは、リソース設定からサービスネットワークを指すチェックマークが付いた矢印として表示されます。リソース設定をサービスネットワークに関連付けると、そのリソース設定は、サービスネットワークに関連付けられた他のサービスと、サービスネットワークに接続されている VPCs 内のクライアントに検出可能になります。

   サービスネットワークからリソースへの破線の矢印は、クライアントからリクエストを受信するリソースを表します。サービスネットワークを使用してトラフィックフローをクライアントに返します。返されるトラフィックを表す矢印は、この図には含まれていません。

8. VPCs をサービスネットワークに接続する

   VPCsは、VPC をサービスネットワークに関連付けるか、VPC エンドポイントを作成するという 2 つの方法でサービスネットワークに接続できます。ここで、サービスネットワーク所有者は VPC 1 と VPC 3 をサービスネットワークに関連付けます。関連付けは、サービスネットワークを

指すチェックマーク付きの矢印を使用して表示されます。これらの関連付けにより、VPC 内のすべてのリソースがクライアントとして機能し、サービスネットワーク内のサービスにリクエストを行うことができます。VPC 1 とサービスネットワークの間の破線の矢印は、接続確立の方向を示しています。サービスネットワークは、サービス 1 ターゲットグループの対象となるリソースへの接続のみを開始します。VPC 1 のすべてのリソースはクライアントとして機能し、サービスネットワークサービスとリソースへの接続を開始できます。

VPC 2 には、関連付けを表す矢印やチェックマークはありません。これは、サービスネットワーク所有者またはサービス所有者がサービスネットワークに VPC 2 を関連付けていないことを意味します。このようになっているのは、この例では、サービス 2 がリクエストを受信し、同じリクエストを使用してレスポンスを送信するだけでよいためです。つまり、サービス 2 のターゲットはクライアントではないために、サービスネットワークの他のサービスにリクエストを送信する必要がありません。

同様に、VPC 4 には、関連付けを表す矢印やチェックマークはありません。つまり、サービスネットワーク所有者またはリソース所有者は、サービスネットワークに VPC 4 を関連付けていません。これは、リソース 1 がリクエストのみを受信し、同じリクエストを使用してレスポンスを送信するためです。サービスネットワーク内の他のサービスやリソースにリクエストを行うことはできません。

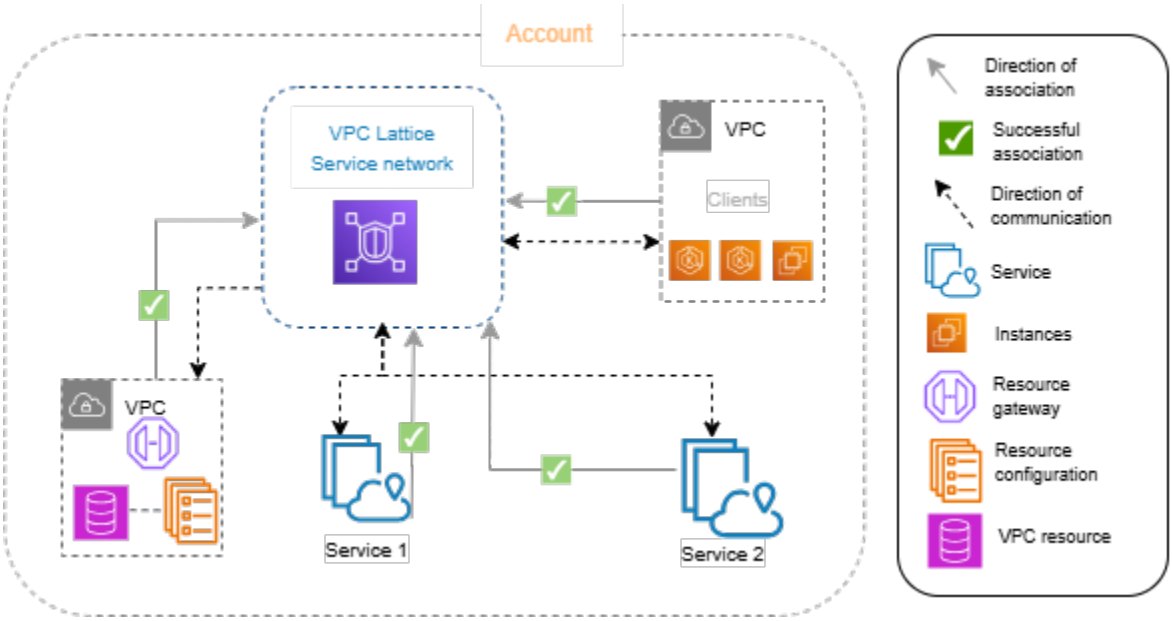
要約すると、次のシナリオが前の図に示されています。

- VPCs Lattice からリソースへの進入専用接続を持つ VPC。VPC 2 と VPC 4 は、これらのシナリオを表します。
- リソースから VPC Lattice への Egress Only 接続を持つ VPC。VPC 3 はこのシナリオを表します。
- VPC Lattice からリソースへの進入接続と、リソースから VPC Lattice への進入接続を持つ VPC。VPC 1 はこのシナリオを表します。

VPC Lattice のサービスネットワーク

サービスネットワークは、サービスとリソース設定の集合の論理的な境界です。ネットワークに関連付けられたサービスおよびリソース設定は、検出、接続、アクセシビリティ、オブザーバビリティについて承認できます。ネットワーク内のサービスおよびリソース設定にリクエストを行うには、サービスまたはクライアントが、関連付けまたは VPC エンドポイントを介してサービスネットワークに接続されている VPC 内にある必要があります。

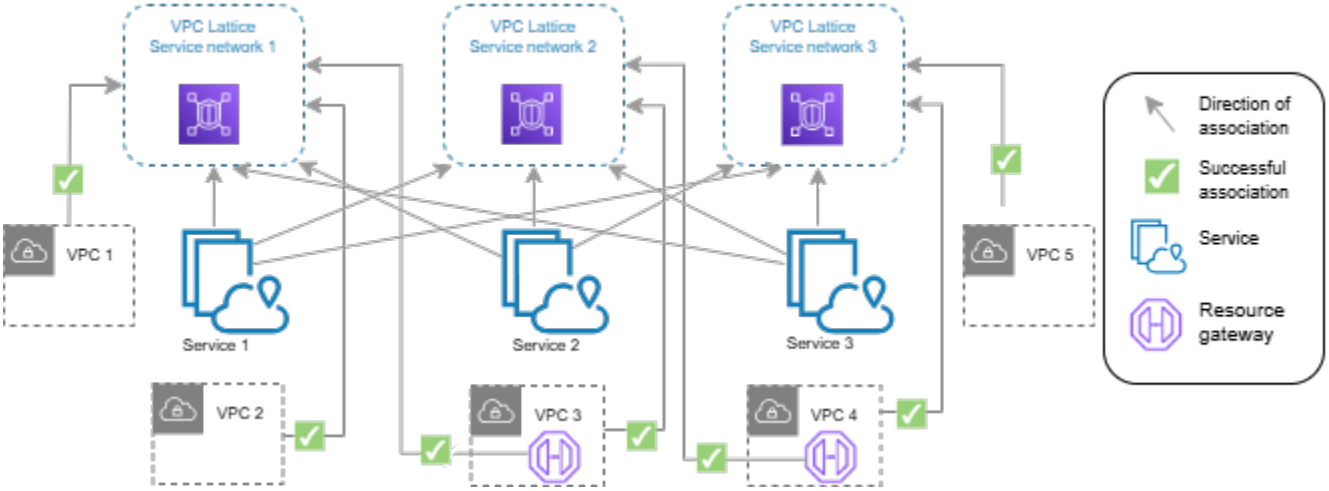
次の図は、Amazon VPC Lattice 内の一般的なサービスネットワークの主要コンポーネントを示しています。矢印のチェックマークは、サービスと VPC がサービスネットワークに関連付けられていることを示しています。サービスネットワークに関連付けられた VPC 内のクライアントは、サービスネットワークを介して両方のサービスと通信できます。



1 つ以上のサービスとリソース設定を複数のサービスネットワークに関連付けることができます。1 つのサービスネットワークに複数の VPCsを接続することもできます。VPC は、関連付けを通じて 1 つのサービスネットワークにのみ接続できます。VPC を複数のサービスネットワークに接続するには、サービスネットワークタイプの VPC エンドポイントを使用できます。タイプのサービスネットワークの VPC エンドポイントの詳細については、 AWS PrivateLink ユーザーガイドを参照してください。

次の図では、矢印はサービスとサービスネットワーク間の関連付け、および VPC とサービスネットワーク間の関連付けを表しています。複数のサービスが複数のサービスネットワークに関連付けられ、複数の VPC が各サービスネットワークに関連付けられていることがわかります。各 VPC には、サービスネットワークへの関連付けが 1 つだけあります。ただし、VPC 3 と VPC 4 は 2 つの

サービスネットワークに接続されます。VPC 3 は、VPC エンドポイントを介してサービスネットワーク 1 に接続します。同様に、VPC 4 は VPC エンドポイントを介してサービスネットワーク 2 に接続します。



詳細については、「Amazon VPC Lattice のクォータ」を参照してください。

内容

- VPC Lattice サービスネットワークを作成する
- VPC Lattice サービスネットワークの関連付けを管理する
- VPC Lattice サービスネットワークのアクセス設定を編集する
- VPC Lattice サービスネットワークのモニタリングの詳細を編集する
- VPC Lattice サービスネットワークのタグを管理する
- VPC Lattice サービスネットワークを削除する

VPC Lattice サービスネットワークを作成する

コンソールを使用してサービスネットワークを作成し、オプションでサービス、関連付け、アクセス設定、アクセスログを設定します。

コンソールを使用してサービスネットワークを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービスネットワーク] を選択します。
3. [サービスネットワークを作成] を選択します。

4. [識別子] には、名前、オプションの説明、オプションのタグを入力します。名前は 3～63 文字にしてください。小文字、数字、ハイフンを使用できます。名前の最初と最後は、文字または数字にしてください。ハイフンは連続して使用しないでください。説明の長さは、最大 256 文字です。タグを追加するには、[新しいタグを追加] を選択し、タグキーとタグ値を指定します。
5. (オプション) サービスを関連付けるには、[サービスの関連付け]、[サービス] からサービスを選択します。このリストには、自分のアカウントにあるサービスと、別のアカウントで共有されているサービスが含まれます。リストにサービスがない場合は、[Create an VPC Lattice service] を選択してサービスを作成できます。

   または、サービスネットワークを作成した後にサービスを関連付ける方法については、「the section called “サービスネットワークサービスの関連付けを管理する”」を参照してください。
6. (オプション) リソース設定を関連付けるには、リソース設定の関連付け、リソース設定からリソース設定サービスを選択します。リストには、アカウントにあるリソース設定と、別のアカウントから共有されているリソース設定が含まれます。リストにリソース設定がない場合は、Amazon VPC Lattice リソース設定の作成 を選択してリソース設定を作成できます。

   または、サービスネットワークを作成した後にリソース設定を関連付けるには、「」を参照してくださいthe section called “サービスネットワークリソースの関連付けを管理する”。
7. (オプション) VPC を関連付けるには、[Add VPC association] を選択します。[VPC] から関連付ける VPC を選択し、[セキュリティグループ] から最大 5 つのセキュリティグループを選択します。セキュリティグループを作成するには、[新しいセキュリティグループを作成] を選択します。

   または、このステップをスキップして、VPC エンドポイント (搭載) を使用して VPC をサービスネットワークに接続することもできます AWS PrivateLink。詳細については、 AWS PrivateLink ユーザーガイドの「サービスネットワークへのアクセス」を参照してください。
8. サービスネットワークを作成するときは、サービスネットワークを他のアカウントと共有するかどうかを決定する必要があります。選択内容は変更不可であり、サービスネットワークの作成後に変更することはできません。共有を許可する場合は、 を介してサービスネットワークを他のアカウントと共有できます AWS Resource Access Manager。

   サービスネットワークを他のアカウントと共有するには、 AWS RAM リソース共有からリソース共有を選択します。

   リソース共有を作成するには、 AWS RAM コンソールに移動し、リソース共有の作成を選択します。

9. [ネットワークアクセス] では、関連する VPC のクライアントがこのサービスネットワーク内のサービスにアクセスできるようにする場合は、デフォルトの認証タイプ [なし] のままにしておくことができます。認証ポリシーを適用してサービスへのアクセスを制御するには、[AWS IAM] を選択し、[認証ポリシー] で次のいずれかを実行します。
   - 入力フィールドにポリシーを入力します。コピーして貼り付けることができるポリシーの例の場合は、[ポリシーの例] を選択します。
   - [ポリシーテンプレートを適用] を選択し、[Allow authenticated and unauthenticated access] テンプレートを選択します。このテンプレートを使用すると、別のアカウントのクライアントは、リクエストに署名する (認証) か、匿名 (未認証) でサービスにアクセスできます。
   - [ポリシーテンプレートを適用] を選択し、[認証されたアクセスのみを許可] テンプレートを選択します。このテンプレートを使用すると、別のアカウントのクライアントは、リクエストに署名すること (認証) によってのみサービスにアクセスできます。
10. (オプション) アクセスログをオンにするには、[アクセスログ] トグルスイッチを選択し、アクセスログの保存先を次のように指定します。
    - [CloudWatch ロググループ] を選択し、CloudWatch ロググループを選択します。ロググループを作成するには、[CloudWatch でロググループを作成する] を選択します。
    - [S3 バケット] を選択し、プレフィックスを含む S3 バケットパスを入力します。S3 バケットを検索するには、[S3 を参照] を選択します。
    - [Kinesis Data Firehose 配信ストリーム] を選択し、配信ストリームを選択します。配信ストリームを作成するには、[Kinesis で配信ストリームを作成] を選択します。
11. (オプション) サービスネットワークを他のアカウントと共有するには、 AWS RAM リソース共有からリソース共有を選択します。リソース共有を作成するには、[RAM コンソールでリソース共有を作成] を選択します。
12. [概要] セクションで設定を確認し、[サービスネットワークを作成] を選択します。

を使用してサービスネットワークを作成するには AWS CLI

create-service-network コマンドを使用します。このコマンドは基本的なサービスネットワークのみを作成します。完全に機能するサービスネットワークを作成するには、サービスの関連付け、VPC の関連付け、アクセス設定を作成するコマンドも使用する必要があります。

VPC Lattice サービスネットワークの関連付けを管理する

サービスまたはリソース設定をサービスネットワークに関連付けると、サービスネットワークに接続されている VPCs内のクライアントがサービスおよびリソース設定にリクエストできるようになりま

す。VPC をサービスネットワークに接続すると、その VPC 内のすべてのターゲットがクライアントになり、サービスネットワーク内の他のサービスやリソース設定と通信できるようになります。

サービスネットワークリソース関連付けのプライベート DNS 対応プロパティは、サービスネットワークエンドポイントのプライベート DNS 対応プロパティとサービスネットワーク VPC 関連付けを上書きします。

サービスネットワーク所有者がサービスネットワークリソースの関連付けを作成し、プライベート DNS を有効にしない場合、VPC Lattice は、プライベート DNS がサービスネットワークエンドポイントまたはサービスネットワーク VPC の関連付けで有効であっても、サービスネットワークが接続されている VPCs でそのリソース設定のプライベートホストゾーンをプロビジョニングしません。

内容

サービスネットワークサービスの関連付けを管理する

自分のアカウントにあるサービスや、別のアカウントで共有されているサービスを関連付けることができます。これは、サービスネットワークを作成する際のオプションのステップです。ただし、サービスを関連付けるまで、サービスネットワークは完全に機能しません。サービス所有者は、自分のアカウントに必要なアクセス権があれば、自分のサービスをサービスネットワークに関連付けることができます。詳細については、「VPC Lattice のアイデンティティベースのポリシーの例」を参照してください。

サービスの関連付けを削除すると、そのサービスはサービスネットワーク内の他のサービスに接続できなくなります。

コンソールを使用してサービスの関連付けを管理するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービスネットワーク] を選択します。
3. サービスネットワークの名前を選択して、その詳細ページを開きます。
4. [サービスの関連付け] タブを選択します。
5. 関連付けを作成するには、次の手順を実行します。

a. [関連付けを作成] を選択します。
b. [サービス] からサービスを選択します。サービスを作成するには、[Amazon VPC Lattice サービスを作成] を選択します。
c. (オプション) タグを追加するには、[サービス関連付けのタグ] を展開して、[新しいタグを追加] を選択し、タグキーとタグ値を入力します。
d. [Save changes] (変更の保存) をクリックします。

6. 関連付けを削除するには、関連付けのチェックボックスをオンにし、[アクション]、[サービスの関連付けを削除] を選択します。確認を求められたら、**confirm**と入力し、[削除] を選択します。

を使用してサービス関連付けを作成するには AWS CLI

create-service-network-service-association コマンドを使用します。

を使用してサービス関連付けを削除するには AWS CLI

delete-service-network-service-association コマンドを使用します。

サービスネットワークリソースの関連付けを管理する

リソース設定は、単一のリソースまたはリソースのグループを表す論理オブジェクトです。アカウントに存在するリソース設定、または異なるアカウントから共有されているリソース設定を関連付けることができます。これは、サービスネットワークを作成する際のオプションのステップです。リソース設定の所有者は、アカウントに必要なアクセス権がある場合、リソース設定をサービスネットワークに関連付けることができます。詳細については、「VPC Lattice のアイデンティティベースのポリシーの例」を参照してください。

サービスネットワークとリソース設定間の関連付けを管理する

サービスネットワークとリソース設定間の関連付けを作成または削除できます。

コンソールを使用してリソース設定の関連付けを管理するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの PrivateLink と Lattice で、サービスネットワークを選択します。
3. サービスネットワークの名前を選択して、その詳細ページを開きます。
4. リソース設定の関連付けタブを選択します。

5. 関連付けを作成するには、次の手順を実行します。

   a. [関連付けを作成] を選択します。
   b. リソース設定で、リソース設定を選択します。
   c. DNS 名 で、VPC Lattice がリソース設定のドメイン名に基づいてリソース設定の関連付けのプライベートホストゾーンをプロビジョニングできるようにする Private DNS enabled を選択します。
   d. (オプション) タグを追加するには、[サービス関連付けのタグ] を展開して、[新しいタグを追加] を選択し、タグキーとタグ値を入力します。
   e. [Save changes] (変更の保存) をクリックします。
6. 関連付けを削除するには、関連付けのチェックボックスをオンにしてから、[アクション]、[削除] と選択します。確認を求められたら、**`confirm`**と入力し、[削除] を選択します。

を使用してリソース設定の関連付けを作成するには AWS CLI

create-service-network-resource-association コマンドを使用します。

を使用してリソース設定の関連付けを削除するには AWS CLI

delete-service-network-resource-association コマンドを使用します。

サービスネットワーク VPC の関連付けを管理する

クライアントがサービスネットワークに関連付けられた VPCs にある場合、クライアントはサービスネットワークに関連付けられたリソース設定で指定されたサービスとリソースにリクエストを送信できます。VPC ピアリング接続またはトランジットゲートウェイを通過するクライアントトラフィックは、サービスネットワークタイプの VPC エンドポイントを使用するサービスネットワークを介してのみ許可されます。

VPC の関連付けは、サービスネットワークを作成する際のオプションのステップです。ネットワーク所有者は、自分のアカウントに必要なアクセス権があれば、VPC をサービスネットワークに関連付けることができます。詳細については、「VPC Lattice のアイデンティティベースのポリシーの例」を参照してください。

リソース設定への VPC 関連付けを作成するときに、プライベート DNS 設定を指定できます。この設定により、VPC Lattice はリソースコンシューマーに代わってプライベートホストゾーンをプロビジョニングできます。詳細については、「the section called “リソースプロバイダーのカスタムドメイン名”」を参照してください。

VPC の関連付けを削除すると、その VPC 内のクライアントはサービスネットワーク内のサービスに接続できなくなります。

コンソールを使用して VPC の関連付けを管理するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービスネットワーク] を選択します。
3. サービスネットワークの名前を選択して、その詳細ページを開きます。
4. [VPC の関連付け] タブを選択します。
5. VPC の関連付けを作成するには、次の手順を実行します。
   a. [VPC の関連付けを作成] を選択します。
   b. [Add VPC association] を選択します。
   c. [VPC] から VPC を選択し、[セキュリティグループ] から最大 5 つのセキュリティグループを選択します。セキュリティグループを作成するには、[新しいセキュリティグループを作成] を選択します。
   d. (オプション) VPC Lattice がリソース設定のドメイン名に基づいてプライベートホストゾーンをプロビジョニングできるようにするには、DNS 名で DNS 名を有効にする を選択し、以下を実行します。
      i. プライベート DNS 設定で、設定を選択します。

         すべてのドメインを選択すると、VPC Lattice はリソース設定のカスタムドメイン名にプライベートホストゾーンをプロビジョニングします。
      ii. (オプション) 検証済みおよび指定されたドメインまたは指定されたドメインを選択した場合は、VPC Lattice でホストゾーンをプロビジョニングするドメインのカンマ区切りリストを入力します。VPC Lattice は、プライベートドメインリストと一致する場合にのみホストゾーンをプロビジョニングします。ワイルドカードマッチングを使用できます。
   e. (オプション) タグを追加するには、[VPC 関連付けタグ] を展開して、[新しいタグを追加] を選択し、タグキーとタグ値を入力します。
   f. [Save changes] (変更の保存) をクリックします。
6. 関連付けのセキュリティグループを編集するには、関連付けのチェックボックスをオンにし、[アクション]、[セキュリティグループの編集] を選択します。必要に応じて、セキュリティグループを追加または削除します。

7. 関連付けを削除するには、関連付けのチェックボックスをオンにし、[アクション]、[VPC の関連付けを削除] を選択します。確認を求められたら、**confirm**と入力し、[削除] を選択します。

を使用して VPC 関連付けを作成するには AWS CLI

create-service-network-vpc-association コマンドを使用します。

を使用して VPC 関連付けのセキュリティグループを更新するには AWS CLI

update-service-network-vpc-association コマンドを使用します。

を使用して VPC の関連付けを削除するには AWS CLI

delete-service-network-vpc-association コマンドを使用します。

サービスネットワーク VPC エンドポイントの関連付けを管理する

クライアントは、VPC の VPC エンドポイント (を使用 AWS PrivateLink) を介して、リソース設定で指定されたサービスおよびリソースにリクエストを送信できます。タイプのサービスネットワークの VPC エンドポイントは、VPC をサービスネットワークに接続します。VPC ピアリング接続、Transit Gateway、Direct Connect、または VPN を介して VPC の外部から送信されるクライアントトラフィックは、VPC エンドポイントを使用してサービスとリソース設定に到達できます。VPC エンドポイントを使用すると、VPC を複数のサービスネットワークに接続できます。VPC に VPC エンドポイントを作成すると、VPC の IP アドレス (マネージドプレフィックスリストの IP アドレスではなく) を使用して、サービスネットワークへの接続を確立します。

リソース設定への VPC 関連付けを作成するときに、プライベート DNS 設定を指定できます。この設定により、VPC Lattice はリソースコンシューマーに代わってプライベートホストゾーンをプロビジョニングできます。詳細については、「the section called “リソースプロバイダーのカスタムドメイン名”」を参照してください。

コンソールを使用して VPC エンドポイントの関連付けを管理するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービスネットワーク] を選択します。
3. サービスネットワークの名前を選択して、その詳細ページを開きます。
4. エンドポイントの関連付けタブを選択すると、サービスネットワークに接続された VPC エンドポイントが表示されます。

5. VPC エンドポイントのエンドポイント ID を選択して、詳細ページを開きます。次に、VPC エンドポイントの関連付けを変更または削除します。

コンソールを使用して新しい VPC エンドポイントの関連付けを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの VPC Lattice で、エンドポイントを選択します。
3. [エンドポイントを作成] を選択します。
4. [タイプ] で [サービスネットワーク] を選択します。
5. VPC に接続するサービスネットワークを選択します。
6. VPC、サブネット、セキュリティグループを選択します。
7. (オプション) プライベート DNS を有効にするには、プライベート DNS を有効にするを選択します。
8. (オプション) タグを追加するには、[VPC 関連付けタグ] を展開して、[新しいタグを追加] を選択し、タグキーとタグ値を入力します。
9. エンドポイントの作成 を選択します。

VPC エンドポイントの詳細については、「 AWS PrivateLink ユーザーガイド」の「サービスネットワークへのアクセス」を参照してください。

VPC Lattice サービスネットワークのアクセス設定を編集する

アクセス設定により、サービスネットワークへのクライアントアクセスを設定および管理できます。アクセス設定には、認証タイプと認証ポリシーが含まれます。認証ポリシーは、VPC Lattice 内のサービスに流れるトラフィックを認証および認可するのに役立ちます。サービスネットワークのアクセス設定は、サービスネットワークに関連付けられたリソース設定には適用されません。

認証ポリシーは、サービスネットワークレベル、サービスレベル、またはその両方で適用できます。通常、認証ポリシーはネットワーク所有者またはクラウド管理者によって適用されます。組織内からの認証された呼び出しを許可したり、特定の条件に一致する匿名の GET リクエストを許可したりするなど、粗粒度の認可を実装できます。サービスレベルでは、サービス所有者はより制限の厳しい高度なコントロールを適用できます。詳細については、「認証ポリシーを使用して VPC Lattice サービスへのアクセスを制御する」を参照してください。

コンソールを使用してアクセスポリシーを追加または更新するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービスネットワーク] を選択します。
3. サービスネットワークの名前を選択して、その詳細ページを開きます。
4. [アクセス] タブを選択して、現在のアクセス設定を確認します。
5. アクセス設定を更新するには、[アクセス設定を編集] を選択します。
6. 関連する VPC のクライアントがこのサービスネットワーク内のサービスにアクセスできるようにするには、[認証タイプ] に [なし] を選択します。
7. リソースポリシーをサービスネットワークに適用するには、[認証タイプ] に [AWS IAM] を選択し、[認証ポリシー] で次のいずれかを実行します。
   - 入力フィールドにポリシーを入力します。コピーして貼り付けることができるポリシーの例の場合は、[ポリシーの例] を選択します。
   - [ポリシーテンプレートを適用] を選択し、[Allow authenticated and unauthenticated access] テンプレートを選択します。このテンプレートを使用すると、別のアカウントのクライアントは、リクエストに署名する (認証) か、匿名 (未認証) でサービスにアクセスできます。
   - [ポリシーテンプレートを適用] を選択し、[認証されたアクセスのみを許可] テンプレートを選択します。このテンプレートを使用すると、別のアカウントのクライアントは、リクエストに署名すること (認証) によってのみサービスにアクセスできます。
8. [Save changes] (変更の保存) をクリックします。

を使用してアクセスポリシーを追加または更新するには AWS CLI

put-auth-policy コマンドを使用します。

VPC Lattice サービスネットワークのモニタリングの詳細を編集する

VPC Lattice はリクエストとレスポンスのたびにメトリクスとログを生成するため、アプリケーションのモニタリングとトラブルシューティングがより効率的になります。

アクセスログを有効にして、ログの送信先リソースを指定できます。VPC Lattice は、CloudWatch Log グループ、Firehose 配信ストリーム、S3 バケットのリソースにログを送信できます。

コンソールを使用してアクセスログを有効にするか、ログの送信先を更新するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービスネットワーク] を選択します。
3. サービスネットワークの名前を選択して、その詳細ページを開きます。
4. [モニタリング] タブを選択します。[アクセスログ] をチェックして、アクセスログが有効になっているかどうかを確認します。
5. アクセスログを有効または無効にするには、[アクセスログを編集] を選択し、[アクセスログ] トグルスイッチをオンまたはオフにします。
6. アクセスログを有効にする場合は、配信先のタイプを選択し、アクセスログの送信先を作成または選択する必要があります。また、配信先はいつでも変更できます。例えば、次のようになります。
   - [CloudWatch ロググループ] を選択し、CloudWatch ロググループを選択します。ロググループを作成するには、[CloudWatch でロググループを作成する] を選択します。
   - [S3 バケット] を選択し、プレフィックスを含む S3 バケットパスを入力します。S3 バケットを検索するには、[S3 を参照] を選択します。
   - [Kinesis Data Firehose 配信ストリーム] を選択し、配信ストリームを選択します。配信ストリームを作成するには、[Kinesis で配信ストリームを作成] を選択します。
7. [Save changes] (変更の保存) をクリックします。

を使用してアクセスログを有効にするには AWS CLI

create-access-log-subscription コマンドを使用します。

を使用してログの送信先を更新するには AWS CLI

update-access-log-subscription コマンドを使用します。

を使用してアクセスログを無効にするには AWS CLI

delete-access-log-subscription コマンドを使用します。

VPC Lattice サービスネットワークのタグを管理する

タグを使用すると、サービスネットワークを目的、所有者、環境などのさまざまな方法で分類することができます。

各サービスネットワークに複数のタグを追加できます。タグキーは、各サービスネットワークごとに一意である必要があります。すでにサービスネットワークに関連付けられているキーを持つタグを追加すると、そのタグの値が更新されます。使用できる文字は、アルファベット、スペース、数字 (UTF-8)、特殊文字 (+-=. _:/@) です。ただし、先頭または末尾にはスペースを使用しないでください。タグ値は大文字と小文字が区別されます。

コンソールを使用してタグを追加または削除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービスネットワーク] を選択します。
3. サービスネットワークの名前を選択して、その詳細ページを開きます。
4. [タグ] タブを選択します。
5. タグを追加するには、[タグを追加] を選択し、タグキーとタグ値を入力します。別のタグを追加するには、[新しいタグを追加] を選択します。タグの追加を完了したら、[Save changes] (変更の保存) を選択します。
6. タグを削除するには、タグのチェックボックスを選択し、[削除] を選択します。確認を求められたら、**`confirm`**と入力し、[削除] を選択します。

を使用してタグを追加または削除するには AWS CLI

tag-resource コマンドと untag-resource コマンドを使用します。

VPC Lattice サービスネットワークを削除する

サービスネットワークを削除する前に、まずサービスネットワークとサービス、リソース設定、VPC、または VPC エンドポイントとの関連付けをすべて削除する必要があります。サービスネットワークを削除すると、リソースポリシー、認証ポリシー、アクセスログサブスクリプションなど、サービスネットワークに関連するすべてのリソースも削除されます。

コンソールを使用してサービスネットワークを削除するには

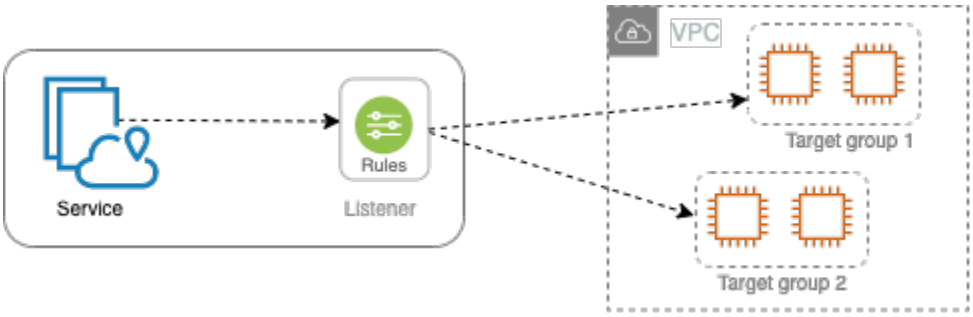
1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービスネットワーク] を選択します。
3. サービスネットワークのチェックボックスをオンにし、[アクション]、[サービスネットワークを削除] の順に選択します。
4. 確認を求められたら、「**`confirm`**」と入力してから、[削除] を選択します。

を使用してサービスネットワークを削除するには AWS CLI

delete-service-network コマンドを使用します。

VPC Lattice のサービス

VPC Lattice 内のサービスは、特定のタスクまたは機能を提供する、独立してデプロイ可能なソフトウェアユニットです。サービスは、アカウントまたは仮想プライベートクラウド (VPC) 内で、インスタンスやコンテナで実行したり、サーバーレス関数として実行したりできます。サービスには、リスナールールと呼ばれるルールを使用するリスナーがあります。リスナールールは、ターゲットへのトラフィックのルーティングに役立つように設定できます。サポートされているターゲットタイプには、EC2 インスタンス、IP アドレス、Lambda 関数、Application Load Balancer、Amazon ECS タスク、Kubernetes Pod などがあります。詳細については、「VPC Lattice のターゲットグループ」を参照してください。1 つのサービスを複数のサービスネットワークに関連付けることができます。次の図は、VPC Lattice 内の一般的なサービスの主要コンポーネントを示しています。



サービスは名前と説明を付けて作成できます。ただし、サービスへのトラフィックを制御し、モニタリングするには、アクセス設定とモニタリングの詳細を含めることが重要です。サービスからターゲットにトラフィックを送信するには、リスナーをセットアップしてルールを設定する必要があります。サービスネットワークからサービスにトラフィックが流れるようにするには、サービスをサービスネットワークに関連付ける必要があります。

ターゲットへの接続には、アイドルタイムアウトと全体的な接続タイムアウトがあります。アイドル接続タイムアウトは 1 分です。この時間が過ぎると接続が閉じられます。最大継続時間は 10 分です。この時間が過ぎると、その接続を介した新しいストリームは許可されなくなり、既存のストリームを閉じる処理が開始されます。

タスク

- ステップ 1: VPC Lattice サービスを作成する
- ステップ 2: ルーティングを定義する
- ステップ 3: ネットワークの関連付けを作成する
- ステップ 4: 確認して作成する
- VPC Lattice サービスの関連付けを管理する

- VPC Lattice サービスのアクセス設定を編集する
- VPC Lattice サービスのモニタリングの詳細を編集する
- VPC Lattice サービスのタグを管理する
- VPC Lattice サービスのカスタムドメイン名を設定する
- VPC Lattice の独自の証明書を使用する (BYOC)
- VPC Lattice サービスを削除する

ステップ 1: VPC Lattice サービスを作成する

アクセス設定とモニタリングの詳細を含む基本的な VPC Lattice サービスを作成します。ただし、ルーティング設定を定義してサービスネットワークに関連付けるまで、サービスは完全には機能しません。

コンソールを使用して基本サービスを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. [サービスを作成] を選択します。
4. [識別子] では、次の手順を実行します。

   a. サービスの名前を入力します。名前は 3～40 文字で、小文字、数字、ハイフンを使用する必要があります。名前の最初と最後は、文字または数字でなければなりません。ダブルハイフンは使用しないでください。

   b. (オプション) サービスネットワークの説明を入力します。説明は、作成中または作成後に設定または変更できます。説明の長さは、最大 256 文字です。
5. サービスのカスタムドメイン名を指定するには、カスタムドメイン名の設定を指定を選択し、カスタムドメイン名を入力します。

   HTTPS リスナーの場合、VPC Lattice が TLS 終了の実行に使用する証明書を選択できます。ここで証明書を選択しない場合は、サービスの HTTPS リスナーを作成するときに選択できます。

   TCP リスナーの場合は、サービスのカスタムドメイン名を指定する必要があります。証明書を指定した場合、その証明書は使用されません。代わりに、アプリケーションで TLS 終了を実行します。
6. サービスネットワークに関連付けられた VPC のクライアントがサービスにアクセスできるようにするには、[サービスアクセス] で [なし] を選択します。認証ポリシーを適用してサービスへの

アクセスを制御するには、[AWS IAM] を選択します。リソースポリシーをサービスに適用するには、認証ポリシーに対して次のいずれかを実行します。

- 入力フィールドにポリシーを入力します。コピーして貼り付けることができるポリシーの例の場合は、[ポリシーの例] を選択します。
- [ポリシーテンプレートを適用] を選択し、[Allow authenticated and unauthenticated access] テンプレートを選択します。このテンプレートを使用すると、別のアカウントのクライアントは、リクエストに署名する (認証) か、匿名 (未認証) でサービスにアクセスできます。
- [ポリシーテンプレートを適用] を選択し、[認証されたアクセスのみを許可] テンプレートを選択します。このテンプレートを使用すると、別のアカウントのクライアントは、リクエストに署名すること (認証) によってのみサービスにアクセスできます。

7. (オプション) アクセスログを有効にするには、[アクセスログ] トグルスイッチをオンにし、アクセスログの保存先を次のように指定します。
   - [CloudWatch ロググループ] を選択し、CloudWatch ロググループを選択します。ロググループを作成するには、[CloudWatch でロググループを作成する] を選択します。
   - [S3 バケット] を選択し、プレフィックスを含む S3 バケットパスを入力します。S3 バケットを検索するには、[S3 を参照] を選択します。
   - [Kinesis Data Firehose 配信ストリーム] を選択し、配信ストリームを選択します。配信ストリームを作成するには、[Kinesis で配信ストリームを作成] を選択します。
8. (オプション) サービスを他のアカウントと共有するには、 AWS RAM リソース共有からリソース共有を選択します。リソース共有を作成するには、[RAM コンソールでリソース共有を作成] を選択します。
9. 設定を確認してサービスを作成するには、[スキップして確認と作成に進む] を選択します。それ以外の場合は、[次へ] を選択してサービスのルーティング設定を定義します。

ステップ 2: ルーティングを定義する

指定したターゲットにサービスがトラフィックを送信できるように、リスナーを使用してルーティング設定を定義します。

前提条件

リスナーを追加する前に、VPC Lattice ターゲットグループを作成する必要があります。詳細については、「the section called “ターゲットグループの作成”」を参照してください。

コンソールを使用してサービスのルーティングを定義するには

1. [リスナーの追加] を選択します。
2. [リスナー名] には、カスタムのリスナー名を指定するか、リスナーのプロトコルとポートをリスナー名として使用できます。指定するカスタム名は最大 63 文字で、アカウント内のサービスごとに一意である必要があります。使用できる文字は a ~ z、0 ~ 9、- (ハイフン) です。最初または最後の文字をハイフンにしたり、別のハイフンの直後にハイフンを入れたりすることはできません。作成後にリスナー名を変更することはできません。
3. プロトコルを選択し、ポート番号を入力します。
4. [デフォルトアクション] では、トラフィックを受信する VPC Lattice ターゲットグループを選択し、このターゲットグループの重み付けを選択します。オプションで、デフォルトアクションに別のターゲットグループを追加できます。[アクションを追加] を選択し、別のターゲットグループを選択して、その重みを指定します。
5. (オプション) 別のルールを追加するには、[ルールを追加] を選択し、ルールの名前、優先度、条件、アクションを入力します。

   各ルールに 1 ~ 100 の範囲で優先度を指定できます。リスナーは同じ優先度の複数のルールを持つことはできません。ルールは優先順位の低~高順によって評価されます。デフォルトのルールが最後に評価されます。

   [条件] にはパス一致条件のパスパターンを入力します。各文字列の最大サイズは 200 文字です。比較では、大文字と小文字は区別されません。
6. (オプション) タグを追加するには、[リスナータグ] を展開して、[新しいタグを追加] を選択し、タグキーとタグ値を入力します。
7. 設定を確認してサービスを作成するには、[スキップして確認と作成に進む] を選択します。それ以外の場合は、[次へ] を選択してサービスをサービスネットワークに関連付けます。

ステップ 3: ネットワークの関連付けを作成する

クライアントが通信できるように、サービスをサービスネットワークに関連付けます。

コンソールを使用してサービスをサービスネットワークに関連付けるには

1. [VPC Lattice サービスネットワーク] では、サービスネットワークを選択します。サービスネットワークを作成するには、[VPC Lattice ネットワークを作成] を選択します。サービスを複数のサービスネットワークに関連付けることができます。

2. (オプション) タグを追加するには、[サービスネットワークの関連付けタグ] を展開して、[新しいタグを追加] を選択し、タグキーとタグ値を入力します。
3. [次へ] を選択します。

ステップ 4: 確認して作成する

コンソールを使用して設定を確認し、サービスを作成するには

1. サービスの設定を確認します。
2. サービス設定の一部を変更する必要がある場合は、[編集] を選択します。
3. 設定の確認または編集が終了したら、[VPC Lattice サービスを作成] を選択します。
4. サービスにカスタムドメイン名を指定した場合は、サービスの作成後に DNS ルーティングを設定する必要があります。詳細については、「the section called “カスタムドメイン名を設定する”」を参照してください。

VPC Lattice サービスの関連付けを管理する

サービスをサービスネットワークに関連付けると、クライアント (サービスネットワークに関連付けられた VPC 内のリソース) がこのサービスにリクエストを送信できるようになります。自分のアカウントにあるサービスや、別のアカウントで共有されているサービスを関連付けることができます。サービスを作成する場合、このステップは任意です。ただし、作成後は、サービスネットワークに関連付けるまで、そのサービスは他のサービスと通信できません。サービス所有者は、自分のアカウントに必要なアクセス権があれば、自分のサービスをサービスネットワークに関連付けることができます。詳細については、「VPC Lattice の仕組み」を参照してください。

コンソールを使用してサービスネットワークの関連付けを管理するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. サービスの名前を選択して、その詳細ページを開きます。
4. [サービスネットワークの関連付け] タブを選択します。
5. 関連付けを作成するには、次の手順を実行します。

    a. [関連付けを作成] を選択します。

b. [VPC Lattice サービスネットワーク] からサービスネットワークを選択します。サービスネットワークを作成するには、[VPC Lattice ネットワークを作成] を選択します。
c. (オプション) タグを追加するには、[サービス関連付けのタグ] を展開して、[新しいタグを追加] を選択し、タグキーとタグ値を入力します。
d. [Save changes] (変更の保存) をクリックします。

6. 関連付けを削除するには、関連付けのチェックボックスをオンにし、[アクション]、[ネットワークの関連付けの削除] を選択します。確認を求められたら、**`confirm`**と入力し、[削除] を選択します。

を使用してサービスネットワークの関連付けを作成するには AWS CLI

create-service-network-service-association コマンドを使用します。

を使用してサービスネットワークの関連付けを削除するには AWS CLI

delete-service-network-service-association コマンドを使用します。

VPC Lattice サービスのアクセス設定を編集する

アクセス設定により、サービスへのクライアントアクセスを設定および管理できます。アクセス設定には、認証タイプと認証ポリシーが含まれます。認証ポリシーは、VPC Lattice 内のサービスに流れるトラフィックを認証および認可するのに役立ちます。

認証ポリシーは、サービスネットワークレベル、サービスレベル、またはその両方で適用できます。サービスレベルでは、サービス所有者はより制限の厳しい高度なコントロールを適用できます。通常、認証ポリシーはネットワーク所有者またはクラウド管理者によって適用されます。組織内からの認証済みの呼び出しを許可したり、特定の条件に一致する匿名の GET リクエストを許可したりするなど、粒度の粗い認可を実装できます。詳細については、「認証ポリシーを使用して VPC Lattice サービスへのアクセスを制御する」を参照してください。

コンソールを使用してアクセスポリシーを追加または更新するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. サービスの名前を選択して、その詳細ページを開きます。
4. [アクセス] タブを選択して、現在のアクセス設定を確認します。
5. アクセス設定を更新するには、[アクセス設定を編集] を選択します。

6. 関連するサービスネットワーク内の VPC のクライアントがサービスにアクセスできるようにするには、[認証タイプ] に [なし] を選択します。
7. リソースポリシーを適用してサービスへのアクセスを制御するには、[認証タイプ] に [AWS IAM] を選択し、[認証ポリシー] で次のいずれかを実行します。
   - 入力フィールドにポリシーを入力します。コピーして貼り付けることができるポリシーの例の場合は、[ポリシーの例] を選択します。
   - [ポリシーテンプレートを適用] を選択し、[Allow authenticated and unauthenticated access] テンプレートを選択します。このテンプレートを使用すると、別のアカウントのクライアントは、リクエストに署名する (認証) か、匿名 (未認証) でサービスにアクセスできます。
   - [ポリシーテンプレートを適用] を選択し、[認証されたアクセスのみを許可] テンプレートを選択します。このテンプレートを使用すると、別のアカウントのクライアントは、リクエストに署名すること (認証) によってのみサービスにアクセスできます。
8. [Save changes] (変更の保存) をクリックします。

を使用してアクセスポリシーを追加または更新するには AWS CLI

put-auth-policy コマンドを使用します。

VPC Lattice サービスのモニタリングの詳細を編集する

VPC Lattice はリクエストとレスポンスのたびにメトリクスとログを生成するため、アプリケーションのモニタリングとトラブルシューティングがより効率的になります。

アクセスログを有効にして、ログの送信先リソースを指定できます。VPC Lattice は、CloudWatch Log グループ、Firehose 配信ストリーム、S3 バケットのリソースにログを送信できます。

コンソールを使用してアクセスログを有効にするか、ログの送信先を更新するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. サービスの名前を選択して、その詳細ページを開きます。
4. [モニタリング] タブを選択し、[ログ] を選択します。[アクセスログ] をチェックして、アクセスログが有効になっているかどうかを確認します。
5. アクセスログを有効または無効にするには、[アクセスログを編集] を選択し、[アクセスログ] トグルスイッチをオンまたはオフにします。

6. アクセスログを有効にする場合は、配信先のタイプを選択し、アクセスログの送信先を作成または選択する必要があります。また、配信先はいつでも変更できます。例えば、次のようになります。
   - [CloudWatch ロググループ] を選択し、CloudWatch ロググループを選択します。ロググループを作成するには、[CloudWatch でロググループを作成する] を選択します。
   - [S3 バケット] を選択し、プレフィックスを含む S3 バケットパスを入力します。S3 バケットを検索するには、[S3 を参照] を選択します。
   - [Kinesis Data Firehose 配信ストリーム] を選択し、配信ストリームを選択します。配信ストリームを作成するには、[Kinesis で配信ストリームを作成] を選択します。
7. [Save changes] (変更の保存) をクリックします。

を使用してアクセスログを有効にするには AWS CLI

create-access-log-subscription コマンドを使用します。

を使用してログの送信先を更新するには AWS CLI

update-access-log-subscription コマンドを使用します。

を使用してアクセスログを無効にするには AWS CLI

delete-access-log-subscription コマンドを使用します。

VPC Lattice サービスのタグを管理する

タグを使用すると、サービスを目的、所有者、環境などのさまざまな方法で分類することができます。

各サービスに複数のタグを追加できます。タグキーは、サービスごとに一意である必要があります。既にサービスに関連付けられているキーを持つタグを追加すると、そのタグの値を更新します。使用できる文字は、アルファベット、スペース、数字 (UTF-8)、特殊文字 (+-=. _:/@) です。ただし、先頭または末尾にはスペースを使用しないでください。タグ値は大文字と小文字が区別されます。

コンソールを使用してタグを追加または削除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. サービスの名前を選択して、その詳細ページを開きます。

4. [タグ] タブを選択します。
5. タグを追加するには、[タグを追加] を選択し、タグキーとタグ値を入力します。別のタグを追加するには、[新しいタグを追加] を選択します。タグの追加を完了したら、[Save changes] (変更の保存) を選択します。
6. タグを削除するには、タグのチェックボックスを選択し、[削除] を選択します。確認を求められたら、**confirm**と入力し、[削除] を選択します。

を使用してタグを追加または削除するには AWS CLI

tag-resource コマンドと untag-resource コマンドを使用します。

VPC Lattice サービスのカスタムドメイン名を設定する

新しいサービスを作成すると、VPC Lattice は次の構文でサービスの一意の完全修飾ドメイン名 (FQDN) を生成します。

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

ただし、VPC Lattice が提供するドメイン名は、ユーザーが覚えやすいものではありません。カスタムドメイン名は、ユーザーに提供できるシンプルで直感的な URLs です。VPC Lattice が生成した DNS 名ではなく、`www.parking.example.com` のようなカスタムドメイン名をサービスに使用する場合は、VPC Lattice サービスを作成するときに設定できます。クライアントがカスタムドメイン名を使用してリクエストを行うと、DNS サーバーが VPC Lattice が生成したドメイン名に解決します。

前提条件

- サービス用に登録されたドメイン名が必要です。ドメイン名をまだ登録していない場合は、Amazon Route 53 やその他の商用レジストラから登録できます。
- HTTPS リクエストを受信するには、 AWS Certificate Managerで独自の証明書を提供する必要があります。VPC Lattice はフォールバックとしてデフォルト証明書をサポートしていません。そのため、カスタムドメイン名に対応する SSL/TLS 証明書を提供しない場合、カスタムドメイン名への HTTPS 接続はすべて失敗します。詳細については、「VPC Lattice の独自の証明書を使用する (BYOC)」を参照してください。

制約事項と考慮事項

- 1 つのサービスに複数のカスタムドメイン名を設定することはできません。
- サービスの作成後にカスタムドメイン名を変更することはできません。
- カスタムドメイン名は、サービスネットワークごとに一意である必要があります。つまり、同じサービスネットワーク内に (別のサービス用の) 既存のカスタムドメイン名を使用してサービスを作成することはできません。

次の手順は、サービスのカスタムドメイン名を設定する方法を示しています。

AWS マネジメントコンソール

サービスのカスタムドメイン名を設定するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. [サービスを作成] を選択します。[ステップ 1: サービスを作成する] に移動します。
4. [カスタムドメイン設定] セクションで、[カスタムドメイン設定を指定] を選択します。
5. カスタムドメイン名を入力します。
6. HTTPS リクエストを処理するには、[カスタム SSL/TLS 証明書] でカスタムドメイン名と一致する SSL/TLS 証明書を選択します。証明書がまだない場合、または今すぐ追加しない場合は、HTTPS リスナーの作成時に証明書を追加できます。ただし、証明書がないと、カスタムドメイン名は HTTPS リクエストを処理できません。詳細については、「HTTPS リスナーの追加」を参照してください。
7. サービスを作成するためのその他の情報をすべて追加し終えたら、[作成] を選択します。

AWS CLI

サービスのカスタムドメイン名を設定するには

create-service コマンドを使用します。

```
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
```

上記のコマンドの --name にサービスの名前を入力します。--custom-domain-name には、parking.example.com などのサービスのドメイン名を入力します。--certificate-

`arn` には、ACM の証明書の ARN を入力します。証明書 ARN は、 AWS Certificate Managerのアカウントで利用できます。

カスタムドメイン名をサービスに関連付ける

まず、カスタムドメイン名をまだ登録していない場合は、登録します。インターネットのドメイン名は、Internet Corporation for Assigned Names and Numbers (ICANN) によって管理されています。ドメイン名は、ドメイン名のレジストリを管理する ICANN 認定機関であるドメイン名レジストラを使用して登録します。レジストラのウェブサイトで、ドメイン名の登録に関する詳細な手順と料金情報を確認します。詳細については、以下のリソースを参照してください。

- Amazon Route 53 を使用してドメイン名を登録するには、Amazon Route 53 開発者ガイドの「Route 53 を使用したドメイン名の登録」を参照してください。
- 認定されているレジストラのリストについては、「認定レジストラディレクトリ」を参照してください。

次に、ドメインレジストラなどの DNS サービスを使用して、クエリをサービスにルーティングするレコードを作成します。詳細については、DNS サービスのドキュメントを参照してください。前出と別に、DNS サービスとして Route 53 を使用する方法もあります。

Route 53 を使用している場合は、エイリアスレコードまたは CNAME レコードを使用して、クエリをサービスにルーティングできます。ゾーン頂点とも呼ばれる DNS 名前空間の上部ノードにエイリアスレコードを作成できるため、エイリアスレコードを使用することをお勧めします。

Route 53 を使用している場合は、まずホストゾーンを作成する必要があります。ホストゾーンには、ドメインのインターネット上のトラフィックをルーティングする方法に関する情報が含まれています。プライベートホストゾーンまたはパブリックホストゾーンを作成したら、 などのカスタムドメイン名が `parking.example.com`などの VPC Lattice 自動生成されたドメイン名にマッピングされるようにレコードを作成します`my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws`。このマッピングがないと、カスタムドメイン名は VPC Lattice で機能しません。

次の手順は、Route 53 を使用してプライベートホストゾーンまたはパブリックホストゾーンを作成する方法を示しています。

AWS マネジメントコンソール

Route 53 を使用してサービスにクエリをルーティングするエイリアスレコードを作成するには、「Amazon VPC Lattice サービスドメインエンドポイントへのトラフィックのルーティング」を参照してください。

サービスに VPC Lattice が生成したドメイン名を使用します。たとえば、 値`my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws`に使用します。この自動生成されたドメイン名は、サービスページの VPC Lattice コンソールにあります。

AWS CLI

ホストゾーンにエイリアスレコードを作成するには

1. サービスの VPC Lattice 生成ドメイン名を取得します (例: `my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws`)。
2. エイリアスを設定するには、以下のコマンドを使用します。

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

change-set.json ファイルは、次の JSON 例の内容で JSON ファイルを作成し、ローカルマシンに保存します。上記のコマンドの *file://~/Desktop/change-set.json* を、ローカルマシンに保存されている JSON ファイルのパスに置き換えます。次の JSON の "Type" は A または AAAA レコードタイプであることに注意してください。

```
{
    "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 Lattice の独自の証明書を使用する (BYOC)

HTTPS リクエストを処理するには、カスタムドメイン名をセットアップする前に、 AWS Certificate Manager (ACM) で独自の SSL/TLS 証明書を準備する必要があります。これらの証明書には、サービスのカスタムドメイン名と一致するサブジェクト代替名 (SAN) または共通名 (CN) が必要です。SAN がある場合は、SAN リスト内でのみ一致がチェックされます。SAN がない場合は、CN に一致があるかどうかがチェックされます。

VPC Lattice は、Server Name Indication (SNI) を使用して HTTPS リクエストを処理します。DNS は、カスタムドメイン名とこのドメイン名に一致する証明書に基づいて HTTPS リクエストを VPC Lattice サービスにルーティングします。ACM でドメイン名の SSL/TLS 証明書をリクエストするか、これを ACM にインポートするには、「AWS Certificate Manager ユーザーガイド」の「証明書を発行して管理する」と「証明書のインポート」を参照してください。ACM で独自の証明書をリクエストまたはインポートできない場合は、VPC Lattice が生成したドメイン名と証明書を使用してください。

VPC Lattice は、サービスごとにカスタム証明書を 1 つだけ受け入れます。ただし、1 つのカスタム証明書を複数のカスタムドメインに使用できます。つまり、1 つのカスタムドメイン名で作成したすべての VPC Lattice サービスに同じ証明書を使用できます。

ACM コンソールを使用して証明書を表示するには、[証明書] を開いて証明書 ID を選択します。[関連リソース] に、その証明書に関連付けられた VPC Lattice サービスが表示されます。

制約事項と考慮事項

- VPC Lattice では、関連する証明書のサブジェクト代替名 (SAN) または共通名 (CN) の 1 レベル深いワイルドカード一致が可能です。例えば、カスタムドメイン名 `parking.example.com` でサービスを作成し、独自の証明書を SAN の `*.example.com` に関連付けるとします。`parking.example.com` にリクエストが送信されると、VPC Lattice は SAN を apex ドメイン `example.com` を持つ任意のドメイン名と照合します。ただし、カスタムドメインが `parking.different.example.com` で、証明書の SAN が `*.example.com` である場合、リクエストは失敗します。

- VPC Lattice は 1 レベルのワイルドカードドメイン一致をサポートします。つまり、ワイルドカードは第 1 レベルのサブドメインとしてのみ使用でき、1 つのサブドメインレベルのみを保護します。例えば、証明書の SAN が `*.example.com` の場合、`parking.*.example.com` はサポートされません。
- VPC Lattice は、ドメイン名ごとに 1 つのワイルドカードをサポートします。つまり、`*.*.example.com` は無効です。詳細については、「AWS Certificate Manager ユーザーガイド」の「パブリック証明書をリクエストする」を参照してください。
- VPC Lattice は 2048 ビットの RSA キーを使用した証明書のみをサポートしています。
- ACM の SSL/TLS 証明書は、関連付けている VPC Lattice サービスと同じリージョンにある必要があります。

証明書のプライベートキーを保護する

ACM を使用して SSL/TLS 証明書をリクエストすると、ACM はパブリックキーとプライベートキーペアを生成します。証明書をインポートすると、キーペアが生成されます。パブリックキーは証明書の一部となります。プライベートキーを安全に保存するために、ACM は AWS KMS KMS キーと呼ばれる別のキーを作成し、エイリアス aws/acm. はこのキー AWS KMS を使用して証明書のプライベートキーを暗号化します。詳細については、「AWS Certificate Manager ユーザーガイド」の「Data protection in AWS Certificate Manager」を参照してください。

VPC Lattice AWS は、TLS Connection Manager を使用します。TLS Connection Manager は、証明書のプライベートキーを AWS のサービス保護および使用するためにのみアクセス可能なサービスです。ACM 証明書を使用して VPC Lattice サービスを作成すると、VPC Lattice は証明書を AWS TLS Connection Manager に関連付けます。これを行うには、 AWS マネージドキー AWS KMS に対して で権限を作成します。この許可により、TLS Connection Manager は AWS KMS を使用して証明書のプライベートキーを復号できます。TLS 接続マネージャは、証明書と復号された (プレーンテキストの) プライベートキーを使用して VPC Lattice サービスのクライアントとの安全な接続 (SSL/TLS セッション) を確立します。証明書と VPC Lattice サービスとの関連付けが解除されると、この許可は廃止されます。詳細については、「AWS Key Management Service デベロッパーガイド」の「グラント」を参照してください。

詳細については、「保管中の暗号化」を参照してください。

VPC Lattice サービスを削除する

VPC Lattice サービスを削除するには、まず、サービスとサービスネットワークとの関連付けをすべて削除する必要があります。サービスを削除すると、リソースポリシー、認証ポリシー、リスナー、

リスナールール、アクセスログサブスクリプションなど、サービスに関連するすべてのリソースも削除されます。

コンソールを使用してサービスを削除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. [サービス] ページで、削除するサービスを選択し、[アクション]、[サービスを削除] の順に選択します。
4. 確認を求めるメッセージが表示されたら、[削除] を選択してください。

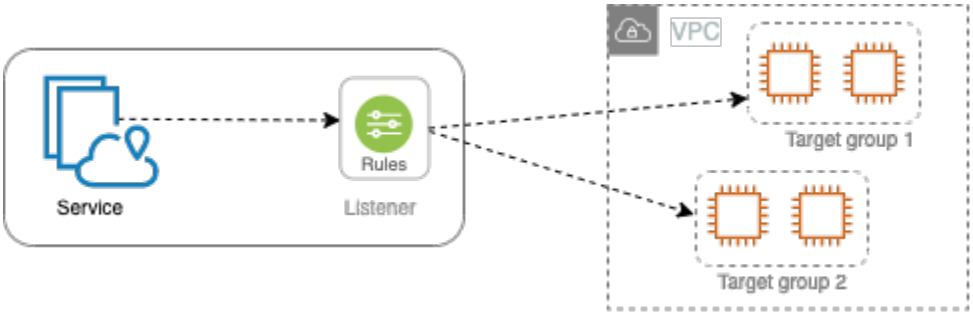
を使用してサービスを削除するには AWS CLI

delete-service コマンドを使用します。

VPC Lattice のターゲットグループ

VPC Lattice ターゲットグループは、アプリケーションまたはサービスを実行するターゲット、つまりコンピューティングリソースのコレクションです。サポートされているターゲットタイプには、EC2 インスタンス、IP アドレス、Lambda 関数、Application Load Balancer、Amazon ECS タスク、Kubernetes Pod などがあります。ターゲットグループには既存のサービスをアタッチすることもできます。VPC Lattice での Kubernetes の使用の詳細については、「AWS ゲートウェイ API コントローラのユーザーガイド」を参照してください。

各ターゲットグループは、1 つ以上の登録されているターゲットにリクエストをルーティングするために使用されます。リスナーのルールを作成するときに、ターゲットグループと条件を指定します。ルールの条件が満たされると、トラフィックが該当するターゲットグループに転送されます。さまざまなタイプのリクエストに応じて別のターゲットグループを作成できます。例えば、一般的なリクエスト用にターゲットグループを作成し、パスやヘッダー値など、特定のルール条件を含むリクエスト用に別のターゲットグループを作成できます。



サービスのヘルスチェック設定は、ターゲットグループ単位で定義します。各ターゲットグループはデフォルトのヘルスチェック設定を使用します。ただし、ターゲットグループを作成したときや、後で変更したときに上書きした場合を除きます。リスナーのルールでターゲットグループを指定すると、サービスは、ターゲットグループに登録されたすべてのターゲットの状態を継続的にモニタリングします。サービスは、正常な登録済みターゲットにリクエストをルーティングします。

ターゲットグループをサービスリスナーのルールで指定するには、サービスと同じアカウントにそのターゲットグループがある必要があります。

VPC Lattice ターゲットグループは Elastic Load Balancing が提供するターゲットグループと似ていますが、これらには互換性はありません。

内容

- VPC Lattice ターゲットグループにターゲットを登録する
- VPC Lattice ターゲットグループのヘルスチェック
- ルーティング設定
- ルーティングアルゴリズム
- [Target type (ターゲットタイプ)]
- IP アドレスタイプ
- VPC Lattice の HTTP ターゲット
- VPC Lattice のターゲットとしての Lambda 関数
- VPC Lattice のターゲットとしての Application Load Balancer
- プロトコルバージョン
- VPC Lattice ターゲットグループのタグ
- VPC Lattice ターゲットグループを削除する

VPC Lattice ターゲットグループを作成する

ターゲットグループにターゲットを登録します。デフォルトでは、VPC Lattice サービスはターゲットグループに指定したポートとプロトコルを使用して登録済みターゲットにリクエストを送信します。ターゲットグループに各ターゲットを登録するときに、このポートを上書きできます。

ターゲットグループ内のターゲットにトラフィックをルーティングするには、リスナーを作成するか、リスナーのルールを作成するときに、アクションでターゲットグループを指定します。詳細については、「VPC Lattice サービスのリスナールール」を参照してください。複数のリスナーに同じターゲットグループを指定できますが、そのリスナーは同じサービスに属している必要があります。サービスでターゲットグループを使用するには、ターゲットグループが他のサービスのリスナーによって使用されていないことを確認する必要があります。

ターゲットグループのタグはいつでも追加または削除できます。詳細については、「VPC Lattice ターゲットグループにターゲットを登録する」を参照してください。ターゲットグループのヘルスチェック設定を変更することもできます。詳細については、「VPC Lattice ターゲットグループのヘルスチェック」を参照してください。

ターゲットグループの作成

以下の手順を実行すると、ターゲットグループを作成し、必要に応じてターゲットを登録できます。

コンソールを使用してターゲットグループを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で [ターゲットグループ] を選択します。
3. [ターゲットグループの作成] を選択します。
4. [ターゲットタイプの選択] で、以下のいずれかを選択します。
   - インスタンス ID でターゲットを登録するには、[インスタンス] を選択します。
   - IP アドレスでターゲットを登録するには、[IP アドレス] を選択します。
   - Lambda 関数をターゲットとして登録するには、[Lambda 関数] を登録します。
   - Application Load Balancer をターゲットとして登録するには、[Application Load Balancer] を選択します
5. [ターゲットグループ名] に、ターゲットグループの名前を入力します。この名前は、各 AWS リージョンのアカウントで一意であること、最大 32 文字であること、英数字またはハイフンのみであること、ハイフンで開始または終了することはできません。
6. [プロトコル] と [ポート] で、必要に応じてデフォルト値を変更できます。デフォルトのプロトコルは [HTTPS] で、デフォルトのポートは [443] です。

   ターゲットタイプが [Lambda 関数] の場合は、プロトコルやポートを指定できません。
7. [IP アドレスタイプ] で、ターゲットを IPv4 アドレスで登録するには [IPv4] を選択し、ターゲットを IPv6 アドレスで登録するには [IPv6] を選択します。ターゲットグループの作成後はこの設定を変更できません。

   このオプションは、ターゲットタイプが [IP アドレス] の場合にのみ使用できます。
8. [VPC] で、Virtual Private Cloud (VPC) を選択します。

   ターゲットタイプが [Lambda 関数] の場合、このオプションは使用できません。
9. [プロトコルバージョン] で、必要に応じてデフォルト値を変更します。デフォルトは [HTTP1] です。

   ターゲットタイプが [Lambda 関数] の場合、このオプションは使用できません。
10. [ヘルスチェック] で、必要に応じてデフォルト設定を変更します。詳細については、「VPC Lattice ターゲットグループのヘルスチェック」を参照してください。

    ターゲットタイプが [Lambda 関数] の場合、ヘルスチェックは使用できません。
11. [Lambda イベント構造バージョン] でバージョンを選択します。詳細については、「the section called “VPC Lattice サービスからのイベントを受け取る”」を参照してください。

このオプションは、ターゲットタイプが [Lambda 関数] の場合にのみ使用できます。

12. (オプション) タグを追加するには、[タグ] を展開し、[新しいタグを追加] を選択して、タグキーとタグ値を入力します。
13. [次へ] を選択します。
14. [ターゲットの登録] では、このステップをスキップするか、以下の手順を実行してターゲットを追加できます。
    - ターゲットタイプがインスタンスである場合は、インスタンスを選択し、[保留中として以下を含める] を選択します。
    - ターゲットタイプが [IP addresses] (IP アドレス) の場合は、以下を実行してください。
        a. [ネットワークを選択] では、ターゲットグループに選択した VPC をそのまま使用するか、[その他のプライベート IP アドレス] を選択します。
        b. [Specify IPs and define ports] に IP アドレスを入力し、ポートを入力します。デフォルトのポートは、ターゲットグループのポートです。
        c. [保留中として以下を含める] をクリックします。
    - ターゲットタイプが [Lambda 関数] の場合は、Lambda 関数を選択します。Lambda 関数を作成するには、[新しい Lambda 関数を作成] を選択します。
    - ターゲットタイプが [Application Load Balancer] の場合、Application Load Balancer を選択します。Application Load Balancer を作成するには、[create an Application Load Balancer] を選択します。
15. [ターゲットグループの作成] を選択します。

    VPC Lattice がターゲットを登録するまでに数分かかる場合があります。詳細については、「DNS の変更が Route 53 とパブリックリゾルバーに反映されるまでに時間がかかるのはなぜですか？」を参照してください。

を使用してターゲットグループを作成するには AWS CLI

ターゲットグループを作成するには create-target-group コマンド、ターゲットを追加するには register-targets コマンドを使用します。

共有サブネット

参加者は VPC Lattice ターゲットグループを共有 VPC に作成できます。共有サブネットには以下のルールが適用されます。

- VPC Lattice サービスのすべての部分 (リスナー、ターゲットグループ、ターゲットなど) は、同じアカウントで作成する必要があります。これらは、VPC Lattice サービスの所有者が所有するサブネット、または VPC Lattice サービスの所有者と共有するサブネットに作成できます。
- ターゲットグループに登録するターゲットは、ターゲットグループと同じアカウントで作成する必要があります。
- VPC の所有者のみが、VPC をサービスネットワークに関連付けることができます。サービスネットワークに関連付けられている共有 VPC の参加者リソースは、サービスネットワークに関連付けられているサービスにリクエストを送信できます。ただし、管理者はセキュリティグループ、ネットワーク ACL、認証ポリシーを使用してこれを防ぐことができます。

VPC Lattice の共有可能なリソースの詳細については、「VPC Lattice エンティティを共有する」を参照してください。

VPC Lattice ターゲットグループにターゲットを登録する

サービスは、クライアントにとって単一の通信先として機能し、正常な登録済みターゲットに受信トラフィックを分散します。各ターゲットは、1 つ以上のターゲットグループに登録できます。

アプリケーションの需要が高まった場合、需要に対処するため、1 つ以上のターゲットグループに追加のターゲットを登録できます。登録処理が完了し、ターゲットが最初のヘルスチェックに合格するとすぐに、サービスは新しく登録したターゲットへのリクエストのルーティングを開始します。

アプリケーションの需要が低下した場合や、ターゲットを保守する必要がある場合、ターゲットグループからターゲットを登録解除することができます。ターゲットを登録解除するとターゲットグループから削除されますが、ターゲットにそれ以外の影響は及びません。登録解除するとすぐに、サービスはターゲットへのリクエストのルーティングを停止します。ターゲットは、未処理のリクエストが完了するまで DRAINING 状態になります。リクエストの受信を再開する準備ができると、ターゲットをターゲットグループに再度登録することができます。

ターゲットグループのターゲットの種類により、ターゲットグループにターゲットを登録する方法が決定されます。詳細については、「[Target type (ターゲットタイプ)]」を参照してください。

以下のコンソール手順を使用して、ターゲットを登録または登録解除します。または、 AWS CLIの register-targets コマンドと deregister-targets コマンドを使用します。

内容

- Lambda 関数の登録または登録解除
- Application Load Balancer の登録または登録解除

インスタンス ID によるターゲットの登録または登録解除

ターゲットインスタンスは、ターゲットグループに指定された仮想プライベートクラウド (VPC) に存在している必要があります。また、インスタンスの登録時の状態は `running` である必要があります。

インスタンス ID でターゲットを登録する場合は、Auto Scaling グループでサービスを使用できます。Auto Scaling グループにターゲットグループをアタッチし、そのグループがスケールアウトすると、Auto Scaling グループによって起動されたインスタンスが自動的にターゲットグループに登録されます。Auto Scaling グループからターゲットグループをデタッチした場合、インスタンスはターゲットグループから自動的に登録解除されます。詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「VPC Lattice ターゲットグループを使用して、トラフィックを Auto Scaling グループにルーティングする」を参照してください。

コンソールを使用してターゲットをインスタンス ID で登録または登録解除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で [ターゲットグループ] を選択します。
3. ターゲットグループの名前を選択して、その詳細ページを開きます。
4. [Targets] タブを選択します。
5. インスタンスを登録するには、[ターゲットの登録] を選択します。インスタンスを選択し、インスタンスポートを入力して、[保留中として以下を含める] を選択します。インスタンスの追加が完了したら、[ターゲットの登録] を選択します。
6. インスタンスを登録解除するには、インスタンスを選択してから [登録解除] を選択します。

IP アドレスによるターゲットの登録または登録解除

ターゲットの IP アドレスは、ターゲットグループに指定した VPC のサブネットのものである必要があります。同じ VPC に別のサービスの IP アドレスを登録することはできません。VPC エンドポイントまたはパブリックにルーティング可能な IP アドレスは登録できません。

コンソールを使用してターゲットを IP アドレスで登録または登録解除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。

2. ナビゲーションペインの [VPC Lattice] で [ターゲットグループ] を選択します。
3. ターゲットグループの名前を選択して、その詳細ページを開きます。
4. [Targets] タブを選択します。
5. IP アドレスを登録するには、[ターゲットの登録] を選択します。IP アドレスごとにネットワークを選択し、IP アドレスかポートを入力して、[保留中として以下を含める] を選択します。アドレスの指定が終了したら、[ターゲットの登録] を選択します。
6. IP アドレスの登録を解除するには、IP アドレスを選択して [登録解除] を選択します。

Lambda 関数の登録または登録解除

ターゲットグループに単一の Lambda 関数を登録できます。トラフィックを Lambda 関数に送信する必要がなくなった場合は、登録を解除できます。Lambda 関数の登録を解除すると、未処理のリクエストは HTTP 5XX エラーで失敗します。ターゲットグループの Lambda 関数を置き換えるよりも、新しいターゲットグループを作成することをお勧めします。

コンソールを使用して Lambda 関数を登録または登録解除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で [ターゲットグループ] を選択します。
3. ターゲットグループの名前を選択して、その詳細ページを開きます。
4. [Targets] タブを選択します。
5. 登録された Lambda 関数が表示されない場合は、[ターゲットの登録] を選択します。Lambda 関数を選択し、[ターゲットの登録] を選択します。
6. Lambda 関数を登録解除するには、[登録解除] を選択します。確認を求められたら、「**confirm**」と入力し、[登録解除] を選択します。

Application Load Balancer の登録または登録解除

各ターゲットグループに単一の Application Load Balancer を登録できます。トラフィックをロードバランサーに送信する必要がなくなった場合は、登録を解除できます。ロードバランサーの登録を解除すると、未処理のリクエストは HTTP 5XX エラーで失敗します。ターゲットグループの Application Load Balancer を置き換えるよりも、新しいターゲットグループを作成することをお勧めします。

コンソールを使用して Application Load Balancer を登録または登録解除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で [ターゲットグループ] を選択します。
3. ターゲットグループの名前を選択して、その詳細ページを開きます。
4. [Targets] タブを選択します。
5. 登録された Application Load Balancer が表示されない場合は、[ターゲットの登録] を選択します。Application Load Balancer を選択し、[ターゲットの登録] を選択します。
6. Application Load Balancer を登録解除するには、[登録解除] を選択します。確認を求められたら、「**`confirm`**」と入力し、[登録解除] を選択します。

VPC Lattice ターゲットグループのヘルスチェック

サービスは、ステータスをテストするため、登録されたターゲットに定期的にリクエストを送信します。これらのテストは、ヘルスチェックと呼ばれます。

各 VPC Lattice サービスは、リクエストを正常なターゲットにのみルーティングします。各サービスは、ターゲットが登録されているターゲットグループのヘルスチェック設定を使用して、各ターゲットの状態を確認します。ターゲットは、登録後に正常と見なされるためには、1 つのヘルスチェックに合格する必要があります。各ヘルスチェックが完了すると、サービスはヘルスチェック用に確立された接続を終了します。

制約事項と考慮事項

- ターゲットグループのプロトコルバージョンが HTTP1 の場合、ヘルスチェックはデフォルトで有効になります。
- ターゲットグループのプロトコルバージョンが HTTP2 の場合、ヘルスチェックはデフォルトでは有効になりません。ただし、ヘルスチェックを有効にして、プロトコルバージョンを HTTP1 または HTTP2 に手動で設定できます。
- ヘルスチェックでは gRPC ターゲットグループのプロトコルバージョンはサポートされません。ただし、ヘルスチェックを有効にする場合は、ヘルスチェックプロトコルのバージョンを HTTP1 または HTTP2 に指定する必要があります。
- ヘルスチェックでは Lambda ターゲットグループはサポートされません。
- ヘルスチェックでは、Application Load Balancer ターゲットグループはサポートされません。ただし、Elastic Load Balancing を使用して Application Load Balancer のターゲットのヘルスチェック

を有効にできます。詳細については、「Application Load Balancer ユーザーガイド」の「ターゲットグループのヘルスチェック」を参照してください。

ヘルスチェックの設定

次の表に示すように、ターゲットグループのターゲットのヘルスチェックを設定します。表で使用される設定名は、API で使用される名前です。サービスは、指定されたポート、プロトコル、および ping パスを使用して、HealthCheckIntervalSeconds 秒ごとに、登録された各ターゲットにヘルスチェックリクエストを送信します。各ヘルスチェックリクエストは独立しており、結果は間隔全体で存続します。ターゲットが応答するまでにかかる時間は、次のヘルスチェックリクエストまでの間隔に影響を与えません。ヘルスチェックが UnhealthyThresholdCount 連続失敗数のしきい値を超えると、サービスはターゲットをサービス停止中の状態にします。ヘルスチェックが HealthyThresholdCount 連続成功数のしきい値を超えると、サービスはターゲットを実行中の状態に戻します。

設定	説明
HealthCheckProtocol	ターゲットでヘルスチェックを実行するときにサービスが使用するプロトコル。使用可能なプロトコルは HTTP および HTTPS です。デフォルトは HTTP プロトコルです。
HealthCheckPort	ターゲットでヘルスチェックを実行するときにサービスが使用するポート。デフォルトでは、各ターゲットがサービスからトラフィックを受信するポートが使用されます。
HealthCheckPath	ターゲットでのヘルスチェックの送信先。 プロトコルバージョンが HTTP1 または HTTP2 の場合は、有効な URI (/パス?クエリ) を指定します。デフォルトは / です。
HealthCheckTimeoutSeconds	ヘルスチェックを失敗と見なす、ターゲットからレスポンスがない時間 (秒単位)。範囲は 1 ~ 120 秒です。ターゲットタイプが `INSTANCE` または `IP` の場合、デフォルトは 5 秒です。こ

設定	説明
	の設定をデフォルト値にリセットするには、0 を指定します。
HealthCheckIntervalSeconds	個々のターゲットのヘルスチェックの概算間隔 (秒単位)。範囲は 5 ~ 300 秒です。ターゲットタイプが INSTANCE または IP の場合、デフォルトは 30 秒です。この設定をデフォルト値にリセットするには、0 を指定します。
HealthyThresholdCount	非正常なターゲットが正常であると見なされるまでに必要なヘルスチェックの連続成功回数。範囲は 2 ~ 10 です。デフォルトは 5 です。この設定をデフォルト値にリセットするには、0 を指定します。
UnhealthyThresholdCount	ターゲットが異常であると見なされるまでに必要なヘルスチェックの連続失敗回数。範囲は 2 ~ 10 です。デフォルトは 2 です。この設定をデフォルト値にリセットするには、0 を指定します。

設定	説明
マッチャー	ターゲットからの正常なレスポンスを確認するために使用するコード。これらは、コンソールでは [成功コード] と呼ばれます。 プロトコルバージョンが HTTP1 または HTTP2 の場合、指定できる値は 200～499 です。複数の値 (例: "200,202") または値の範囲 (例: "200-299") を指定できます。デフォルト値は 200 です。 gRPC のヘルスチェックプロトコルのバージョンは現在サポートされていません。ただし、ターゲットグループのプロトコルバージョンが gRPC の場合、ヘルスチェックの設定で HTTP1 または HTTP2 のプロトコルバージョンを指定できます。

ターゲットのヘルスステータスをチェックする

ターゲットグループに登録されたターゲットのヘルスステータスをチェックできます。

コンソールを使用してターゲットのヘルスステータスをチェックするには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で [ターゲットグループ] を選択します。
3. ターゲットグループの名前を選択して、その詳細ページを開きます。
4. [Targets] (ターゲット) タブの [Health status] (ヘルスステータス) 列は、各ターゲットのステータスを示します。ステータスの値が `Healthy` 以外の場合は、[ヘルスステータスの詳細] 列に詳細情報が表示されます。

を使用してターゲットの状態を確認するには AWS CLI

list-targets コマンドを使用します。このコマンドの出力にはターゲットのヘルス状態が含まれます。ステータスの値が `Healthy` 以外の場合は、理由コードも出力に含まれています。

異常なターゲットに関する E メール通知を受信するには

CloudWatch アラームを使用して、異常なターゲットに関する詳細を送信する Lambda 関数を開始します。

ヘルスチェックの設定を変更する

ターゲットグループのヘルスチェック設定はいつでも変更できます。

コンソールを使用してヘルスチェックの設定を変更するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で [ターゲットグループ] を選択します。
3. ターゲットグループの名前を選択して、その詳細ページを開きます。
4. [ヘルスチェック] タブの [ヘルスチェックの設定] で [編集] を選択します。
5. 必要に応じてヘルスチェックの設定を変更します。
6. [Save changes] (変更の保存) をクリックします。

を使用してヘルスチェックの設定を変更するには AWS CLI

update-target-group コマンドを使用します。

ルーティング設定

デフォルトでは、サービスはターゲットグループの作成時に指定したプロトコルとポート番号を使用して、リクエストをターゲットにルーティングします。または、ターゲットグループへの登録時にターゲットへのトラフィックのルーティングに使用されるポートを上書きすることもできます。

ターゲットグループでは、次のプロトコルとポートがサポートされています。

- プロトコル: HTTP、HTTPS、TCP
- ポート: 1 ~ 65535

ターゲットグループが HTTPS プロトコルで設定されている場合、または HTTPS ヘルスチェックを使用している場合、ターゲットへの TLS 接続はリスナーのセキュリティポリシーを使用します。VPC Lattice は、ターゲットにインストールした証明書を使用して、ターゲットとの TLS 接続を確立します。VPC Lattice はこれらの証明書を検証しません。したがって、自己署名証明書または期

限切れの証明書を使用できます。VPC Lattice とターゲット間のトラフィックはパケットレベルで認証されるため、ターゲットの証明書が有効でなくてもman-in-the-middle攻撃やスプーフィングのリスクはありません。

TCP ターゲットグループは TLS リスナーでのみサポートされます。

ルーティングアルゴリズム

デフォルトでは、ラウンドロビンルーティングアルゴリズムが正常なターゲットへのリクエストのルーティングに使用されます。

リクエストを受け取ると、VPC Lattice サービスは以下のプロセスを使用します。

1. リスナールールを優先度順に評価して、適用するルールを決定します。
2. デフォルトのラウンドロビンアルゴリズムを使用して、ルールアクションのターゲットグループからターゲットを選択します。それぞれのターゲットグループでルーティングは個別に実行され、複数のターゲットグループに登録されているターゲットの場合も同じです。

ターゲットグループに異常なターゲットのみが含まれている場合、そのヘルスステータスにかかわらず、リクエストはすべてのターゲットにルーティングされます。つまり、すべてのターゲットが同時にヘルスチェックに失敗すると、VPC Lattice サービスがオープンに失敗します。フェイルオープンの効果は、ヘルスステータスにかかわらず、ラウンドロビンアルゴリズムに基づいて、すべてのターゲットへのトラフィックを許可することです。

[Target type (ターゲットタイプ)]

ターゲットグループを作成するときは、そのターゲットの種類を指定します。それにより、このターゲットグループ内でターゲットを登録するときに指定するターゲットの種類が決定されます。ターゲットグループを作成した後で、ターゲットタイプを変更することはできません。

可能なターゲットの種類は次のとおりです。

`INSTANCE`

インスタンス ID で指定されたターゲット。

`IP`

ターゲットは IP アドレスです。

`LAMBDA`

ターゲットは Lambda 関数です。

`ALB`

ターゲットは Application Load Balancer です。

考慮事項

- ターゲットタイプが `IP` である場合、ターゲットグループの VPC のサブネットの IP アドレスを指定する必要があります。この VPC 外部の IP アドレスを登録する必要がある場合は、`ALB` タイプのターゲットグループを作成し、その IP アドレスを Application Load Balancer に登録します。
- ターゲットタイプが `IP` である場合、VPC エンドポイントやパブリックにルーティング可能な IP アドレスは登録できません。
- ターゲットタイプが `LAMBDA` である場合、1 つの Lambda 関数を登録できます。サービスが Lambda 関数のリクエストを受け取ると、Lambda 関数を呼び出します。1 つのサービスに複数の Lambda 関数を登録するには、複数のターゲットグループを使用する必要があります。
- ターゲットタイプが の場合`ALB`、単一の内部 Application Load Balancer を最大 2 つの VPC Lattice サービスのターゲットとして登録できます。その場合、Application Load Balancer を 2 つの異なるターゲットグループに登録します。これは 2 つの異なる VPC Lattice サービスによって使用されます。また、ターゲットの Application Load Balancer には、ターゲットグループポートと一致するポートを持つリスナーが 1 つ以上必要です。
- ECS タスクは、起動時に VPC Lattice ターゲットグループに自動的に登録できます。ターゲットグループは、`IP` のターゲットタイプを持つ必要があります。詳細については、「Amazon Elastic Container Service デベロッパーガイド」の「Amazon ECS サービスで VPC Lattice を使用する」を参照してください。

  または、Amazon ECS サービスの Application Load Balancer をタイプ の VPC Lattice ターゲットグループに登録します`ALB`。詳細については、「Amazon Elastic Container Service デベロッパーガイド」の「ロードバランシングを使用して Amazon ECS サービストラフィックを分散する」を参照してください。
- EKS ポッドをターゲットとして登録するには、Kubernetes サービスから IP アドレスを取得する AWS ゲートウェイ API コントローラーを使用します。
- ターゲットグループプロトコルが TCP の場合、サポートされているターゲットタイプは `INSTANCE`、`IP`、または のみです`ALB`。

IP アドレスタイプ

ターゲットタイプ IP のターゲットグループを作成すると、ターゲットグループの IP アドレスタイプを指定できます。これにより、ターゲットにリクエストやヘルスチェックを送信するためにロードバランサーが使用するアドレスの種類が指定されます。指定できる値は IPv4 および IPv6 です。デフォルトは IPV4 です。

考慮事項

- IP アドレスタイプ IPv6 でターゲットグループを作成する場合、ターゲットグループに指定する VPC には IPv6 アドレス範囲が必要です。
- ターゲットグループに登録する IP アドレスは、ターゲットグループの IP アドレスタイプと一致する必要があります。例えば、IP アドレスタイプが IPv4 の場合、IPv6 アドレスをターゲットグループに登録できません。
- ターゲットグループに登録する IP アドレスは、ターゲットグループに指定した VPC の IP アドレスの範囲内にある必要があります。

VPC Lattice の HTTP ターゲット

HTTP リクエストと HTTP レスポンスは、ヘッダーフィールドを使用して HTTP メッセージに関する情報を送信します。HTTP ヘッダーは自動的に追加されます。ヘッダーフィールドはコロンで区切られた名前と値のペアであり、キャリッジリターン (CR) とラインフィード (LF) で区切ります。HTTP ヘッダーフィールドの標準セットは、「メッセージヘッダー」RFC 2616 で定義されています。アプリケーションで広く使用されている標準以外の HTTP ヘッダーもあります。例えば、x-forwarded プレフィックスが付いた標準外の HTTP ヘッダーがあります。

x-forwarded ヘッダー

Amazon VPC Lattice は、以下の x-forwarded ヘッダーを追加します。

x-forwarded-for

送信元 IP アドレス。

x-forwarded-port

送信先ポート

`x-forwarded-proto`

接続プロトコル (`http` | `https`)。

発信者 ID ヘッダー

Amazon VPC Lattice は、以下の発信者 ID ヘッダーを追加します。

`x-amzn-lattice-identity`

ID 情報。 AWS 認証が成功すると、以下のフィールドが表示されます。

- `Principal` — 認証されたプリンシパル。
- `PrincipalOrgID` — 認証されたプリンシパルの組織の ID。
- `PrincipalOrgPath` – 認証されたプリンシパルの組織パス。
- `SessionName` - 認証されたセッションの名前。

Roles Anywhere の認証情報が使用され、認証が成功すると、以下のフィールドが表示されます。

- `X509Issuer/OU` — 発行者 (OU)。
- `X509SAN/DNS` — サブジェクト代替名 (DNS)。
- `X509SAN/NameCN` — 発行者代替名 (名前/CN)。
- `X509SAN/URI` — サブジェクト代替名 (URI)。
- `X509Subject/CN` — サブジェクト名 (CN)。

`x-amzn-lattice-identity-tags`

プリンシパル ID と任意のプリンシパルタグ。形式は次のとおりです。

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

VPC Lattice は、値内のセミコロン (;) をバックスラッシュ (\) でエスケープします。

`x-amzn-lattice-network`

VPC。形式は次のとおりです。

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

`x-amzn-lattice-target`

ターゲット。形式は次のとおりです。

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

VPC Lattice のリソース ARN の詳細については、「Amazon VPC Lattice で定義されるリソースタイプ」を参照してください。

発信者 ID ヘッダーを偽装することはできません。VPC Lattice は、受信リクエストからこれらのヘッダーを削除します。これらの ID ヘッダーは、次の形式を使用して空の値をサポートするマップを表します。解析するときは、これらのヘッダーの KEYs の特定の順序に依存しないでください。新しい KEYsはいつでも追加されることを想定し、空の値を処理する準備を整える必要があります。

形式は次のとおりです。

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

VPC Lattice のターゲットとしての Lambda 関数

Lambda 関数を VPC Lattice ターゲットグループのターゲットとして登録し、Lambda 関数のターゲットグループにリクエストを転送するリスナールールを設定できます。サービスが Lambda 関数をターゲットとしてターゲットグループにリクエストを転送すると、Lambda 関数を呼び出し、リクエストのコンテンツを JSON 形式で Lambda 関数に渡します。

制限事項

- Lambda 関数とターゲットグループは、同じアカウントおよび同じリージョンにある必要があります。
- Lambda 関数に送信できるリクエストボディの最大サイズは 6 MB です。
- Lambda 関数が送信できるレスポンス JSON の最大サイズは 6 MB です。
- プロトコルは HTTP または HTTPS である必要があります。

Lambda 関数の準備

VPC Lattice サービスで Lambda 関数を使用している場合は、以下の推奨事項が適用されます。

Lambda 関数を呼び出すアクセス許可

ターゲットグループを作成し、 AWS マネジメントコンソール または を使用して Lambda 関数を登録すると AWS CLI、VPC Lattice はユーザーに代わって Lambda 関数ポリシーに必要なアクセス許可を追加します。

また、次の API 呼び出しを使用すると、自分で権限を追加できます。

```
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 関数のバージョニング

ターゲットグループごとに 1 つの Lambda 関数を登録できます。Lambda 関数を変更し、VPC Lattice サービスが常に現行バージョンの Lambda 関数を呼び出せるようにするには、関数のエイリアスを作成し、VPC Lattice サービスに Lambda 関数を登録するときに関数 ARN にエイリアスを含めます。詳細については、「 AWS Lambda デベロッパーガイド」の「Lambda 関数のバージョン」および「Lambda 関数のエイリアスを作成する」を参照してください。

Lambda 関数のターゲットグループの作成

リクエストルーティングで使用されるターゲットグループを作成します。リクエストのコンテンツが、コンテンツをこのターゲットグループに転送するアクションを含むリスナールールと一致する場合、VPC Lattice サービスは登録された Lambda 関数を呼び出します。

コンソールを使用してターゲットグループを作成し、Lambda 関数を登録するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で [ターゲットグループ] を選択します。
3. [ターゲットグループの作成] を選択します。
4. [ターゲットタイプの選択] で [Lambda 関数] を選択します。
5. [ターゲットグループ名] に、ターゲットグループの名前を入力します。
6. [Lambda イベント構造バージョン] でバージョンを選択します。詳細については、「the section called “VPC Lattice サービスからのイベントを受け取る”」を参照してください。

7. (オプション) タグを追加するには、[タグ] を展開し、[新しいタグを追加] を選択して、タグキーとタグ値を入力します。
8. [次へ] を選択します。
9. [Lambda 関数] で、次のいずれかを実行します。
   - 既存の Lambda 関数を選択します。
   - 新しい Lambda 関数を作成し、その関数を選択します。
   - Lambda 関数は後ほど登録します。
10. [ターゲットグループの作成] を選択します。

を使用してターゲットグループを作成し、Lambda 関数を登録するには AWS CLI

create-target-group と register-targets コマンドを使用します。

VPC Lattice サービスからのイベントを受け取る

VPC Lattice サービスは、HTTP と HTTPS の両方を経由するリクエストの Lambda 呼び出しをサポートします。このサービスは JSON 形式でイベントを送信し、すべてのリクエストに `X-Forwarded-For` ヘッダーを追加します。

Base64 エンコード

`content-encoding` ヘッダーが存在し、コンテンツタイプが以下のいずれでもない場合、サービス Base64 は本文をエンコードします。

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

`content-encoding` ヘッダーが存在しない場合、Base64 エンコーディングはコンテンツタイプによって異なります。上述のコンテンツタイプの場合、サービスは Base64 エンコーディングせずに本文をそのまま送信します。

イベント構造の形式

`LAMBDA` タイプのターゲットグループを作成または更新するときには、Lambda 関数が受け取るイベント構造のバージョンを指定できます。指定できるバージョンは `V1` と `V2` です。

Exampleサンプルイベント: V2

```
{
    "version": "2.0",
    "path": "/",
    "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

リクエスト本文。プロトコルが HTTP、HTTPS、gRPC の場合にのみ存在します。

`headers`

リクエストの HTTP ヘッダー。プロトコルが HTTP、HTTPS、gRPC の場合にのみ存在します。

`identity`

ID 情報。有効なフィールドには以下のものがあります。

- `principal` — 認証されたプリンシパル。 AWS 認証が成功した場合にのみ表示されます。
- `principalOrgID` — 認証されたプリンシパルの組織の ID。 AWS 認証が成功した場合にのみ表示されます。
- `sessionName` - 認証されたセッションの名前。 AWS 認証が成功した場合にのみ表示されます。
- `sourceVpcArn` — リクエストが発生した VPC の ARN。ソース VPC が特定できる場合にのみ表示されます。
- `type` – 認証ポリシーが使用され、 AWS 認証が成功`AWS_IAM`した場合、値は です。

Roles Anywhere の認証情報が使用され、認証が成功すると、以下のフィールドが表示される場合があります。

- `x509IssuerOu` — 発行者 (OU)。
- `x509SanDns` — サブジェクト代替名 (DNS)。
- `x509SanNameCn` — 発行者代替名 (名前/CN)。
- `x509SanUri` — サブジェクト代替名 (URI)。
- `x509SubjectCn` — サブジェクト名 (CN)。

`isBase64Encoded`

本文が base64 エンコードされているかどうかを示します。プロトコルが
HTTP、HTTPS、gRPC であり、リクエスト本文がまだ文字列でない場合にのみ表示されます。

`method`

リクエストの HTTP メソッド。プロトコルが HTTP、HTTPS、gRPC の場合にのみ存在します。

`path`

リクエストのパス。プロトコルが HTTP、HTTPS、gRPC の場合にのみ存在します。

`queryStringParameters`

HTTP クエリ文字列パラメータ。プロトコルが HTTP、HTTPS、gRPC の場合にのみ存在します。

`serviceArn`

リクエストを受け取るサービスの ARN。

`serviceNetworkArn`

リクエストを配信するサービスネットワークの ARN。

`targetGroupArn`

リクエストを受け取るターゲットグループの ARN。

`timeEpoch`

時間 (秒単位)。

Exampleサンプルイベント: V1

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

VPC Lattice サービスへのレスポンス

Lambda 関数からのレスポンスには、Base64 エンコーディングのステータス、ステータスコード、およびヘッダーが含まれます。本文は省略できます。

レスポンス本文にバイナリコンテンツを含めるには、コンテンツを Base64 でエンコードし、`isBase64Encoded` を `true` に設定する必要があります。サービスはコンテンツをデコードしてバイナリコンテンツを取得し、そのコンテンツを HTTP レスポンスの本文でクライアントに送信します。

VPC Lattice サービスでは、`Connection` や `Transfer-Encoding` などのホップバイホップ方式のヘッダーは受け付けません。サービスがクライアントにレスポンスを送信する前に計算するため、`Content-Length` ヘッダーは省略できます。

Lambda 関数からのレスポンスの例を次に示します。

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

複数値ヘッダー

VPC Lattice は、クライアントからのリクエスト、または複数の値を持つヘッダーを含むか、同じヘッダーを複数回含む Lambda 関数からのレスポンスをサポートします。VPC Lattice はすべての値をターゲットに渡します。

次の例では、異なる値header1を持つ という名前の 2 つのヘッダーがあります。

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

V2 イベント構造では、VPC Lattice はリスト内の値を送信します。例えば、次のようになります。

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

V1 イベント構造では、VPC Lattice は値を 1 つの文字列にまとめます。例えば、次のようになります。

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

複数値のクエリ文字列パラメータ

VPC Lattice は、同じキーに対して複数の値を持つクエリパラメータをサポートします。

次の例では、異なる値QS1を持つ という名前の 2 つのパラメータがあります。

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

V2 イベント構造では、VPC Lattice はリスト内の値を送信します。例えば、次のようになります。

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

V1 イベント構造では、VPC Lattice は最後に渡された値を使用します。例えば、次のようになります。

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

Lambda 関数の登録解除

トラフィックを Lambda 関数に送信する必要がなくなった場合は、登録を解除できます。Lambda 関数の登録を解除すると、未処理のリクエストは HTTP 5XX エラーで失敗します。

Lambda 関数を置き換えるには、新しいターゲットグループを作成し、新しい関数を新しいターゲットグループに登録し、リスナールールを更新して既存のターゲットグループではなく新しいターゲットグループを使用することをお勧めします。

コンソールを使用して Lambda 関数を登録解除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で [ターゲットグループ] を選択します。
3. ターゲットグループの名前を選択して、その詳細ページを開きます。
4. [ターゲット] タブで、[登録解除] を選択します。
5. 確認を求められたら、「**confirm**」と入力し、[登録解除] を選択します。

を使用して Lambda 関数の登録を解除するには AWS CLI

deregister-targets コマンドを使用します。

VPC Lattice のターゲットとしての Application Load Balancer

VPC Lattice ターゲットグループを作成し、1 つの内部 Application Load Balancer をターゲットとして登録し、このターゲットグループにトラフィックを転送するように VPC Lattice サービスを設定できます。このシナリオでは、トラフィックがターゲットに到達するとすぐに、Application Load Balancer がルーティングの決定を引き継ぎます。この設定では、Application Load Balancer のレイヤー 7 リクエストベースのルーティング機能を、IAM 認証と認可などの VPC Lattice がサポートする機能や VPC とアカウントの間の接続と組み合わせて使用できます。

制限事項

- タイプ ALB の VPC Lattice ターゲットグループでは、1 つの内部 Application Load Balancer をターゲットとして登録できます。
- 1 つの Application Load Balancer を、2 つの異なる VPC Lattice サービスで使用される最大 2 つの VPC Lattice ターゲットグループのターゲットとして登録できます。
- VPC Lattice は、ALB タイプのターゲットグループにはヘルスチェックを行いません。ただし、Elastic Load Balancing のターゲットには、ロードバランサーレベルで個別にヘルスチェックを設定できます。詳細については、「Application Load Balancer ユーザーガイド」の「ターゲットグループのヘルスチェック」を参照してください。

前提条件

VPC Lattice ターゲットグループにターゲットとして登録する Application Load Balancer を作成します。ロードバランサーは次の基準を満たしている必要があります。

- ロードバランサースキームは [内部] です。
- Application Load Balancer は VPC Lattice ターゲットグループと同じアカウントにあり、[アクティブ] 状態である必要があります。
- Application Load Balancer は、VPC Lattice ターゲットグループと同じ VPC にある必要があります。
- Application Load Balancer の HTTPS リスナーを使用すると、TLS を終了できます。ただし、VPC Lattice サービスがロードバランサーと同じ SSL/TLS 証明書を使用している場合に限ります。
- VPC Lattice サービスのクライアント IP を `X-Forwarded-For` リクエストヘッダーに保存するには、Application Load Balancer の属性 `routing.http.xff_header_processing.mode` を `Preserve` に設定する必要があります。値が `Preserve` の場合、ロードバランサーは HTTP リクエストの `X-Forwarded-For` ヘッダーを保持し、変更を加えずにターゲットに送信します。

詳細については、「Application Load Balancer ユーザーガイド」の「Application Load Balancer の作成」を参照してください。

ステップ 1: ALB タイプのターゲットグループを作成する

以下の手順に従って、ターゲットグループを作成します。VPC Lattice はALBターゲットグループのヘルスチェックをサポートしていないことに注意してください。ただし、Application Load Balancer

のターゲットグループのヘルスチェックを設定できます。詳細については、「Application Load Balancer ユーザーガイド」の「ターゲットグループのヘルスチェック」を参照してください。

ターゲットグループを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で [ターゲットグループ] を選択します。
3. [ターゲットグループの作成] を選択します。
4. [Specify target group details] ページの [基本設定] で、[Application Load Balancer] をターゲットタイプとして選択します。
5. [ターゲットグループ名] に、ターゲットグループの名前を入力します。
6. Protocol で、**HTTP**、**HTTPS**、または を選択します**TCP**。ターゲットグループのプロトコルは、内部 Application Load Balancer のリスナーのプロトコルと一致する必要があります。
7. [ポート] で、ターゲットグループのポートを指定します。このポートは、内部 Application Load Balancer のリスナーのポートと一致する必要があります。または、ここで指定するターゲットグループのポートと一致するリスナーポートを内部 Application Load Balancer に追加することもできます。
8. [VPC] で、内部 Application Load Balancer を作成したときに選択したものと同じ仮想プライベートクラウド (VPC) を選択します。これは VPC Lattice リソースを含む VPC になります。
9. [プロトコルバージョン] で、Application Load Balancer がサポートするプロトコルバージョンを選択します。
10. (オプション) 必要なタグを追加します。
11. [次へ] を選択します。

ステップ 2: Application Load Balancer をターゲットとして登録する

今すぐロードバランサーをターゲットとして登録できますが、後で登録することもできます。

Application Load Balancer をターゲットとして登録するには

1. [今すぐ登録] を選択します。
2. [Application Load Balancer] で、内部 Application Load Balancer を選択します。
3. [ポート] はデフォルトのままにするか、必要に応じて別のポートを指定します。このポートは、Application Load Balancer の既存のリスナーポートと一致する必要があります。一致するポートがない状態で続行すると、トラフィックが Application Load Balancer に到達しません。

4. [ターゲットグループの作成] を選択します。

プロトコルバージョン

デフォルトでは、サービスは HTTP/1.1 を使用してターゲットにリクエストを送信します。プロトコルバージョンを使用して、HTTP/2 または grPC を使用するターゲットにリクエストを送信できます。

次の表は、リクエストプロトコルとターゲットグループのプロトコルバージョンの組み合わせの結果をまとめたものです。

リクエストプロトコル	プロトコルバージョン	結果
HTTP/1.1	HTTP/1.1	成功
HTTP/2	HTTP/1.1	成功
gRPC	HTTP/1.1	エラー
HTTP/1.1	HTTP/2	エラー
HTTP/2	HTTP/2	成功
gRPC	HTTP/2	ターゲットが grPC をサポートしている場合は成功
HTTP/1.1	gRPC	エラー
HTTP/2	gRPC	POST リクエストの場合は成功
gRPC	gRPC	成功

gRPC プロトコルバージョンの考慮事項

- サポートされているリスナープロトコルは HTTPS だけです。
- サポートされているターゲットタイプは、`INSTANCE` と `IP` のみです。

- サービスは、gRPC リクエストを解析し、パッケージ、サービス、メソッドに基づいて、適切なターゲットグループに gRPC 呼び出しをルーティングします。
- Lambda 関数をターゲットとして使用することはできません。

HTTP/2 プロトコルバージョンの考慮事項

- サポートされているリスナープロトコルは HTTPS だけです。ターゲットグループのプロトコルには、HTTP または HTTPS を選択できます。
- サポートされているリスナールールは、転送と固定レスポンスのみです。
- サポートされているターゲットタイプは、`INSTANCE` と `IP` のみです。
- サービスは、クライアントからのストリーミングをサポートします。サービスは、ターゲットへのストリーミングをサポートしていません。

VPC Lattice ターゲットグループのタグ

タグを使用すると、ターゲットグループを目的、所有者、環境などさまざまな方法で分類することができます。

各ターゲットグループに対して複数のタグを追加できます。タグキーは、各ターゲットグループで一意である必要があります。すでにターゲットグループに関連付けられているキーを持つタグを追加すると、そのキーの値が更新されます。

不要になったタグは、削除することができます。

制限事項

- リソースあたりのタグの最大数 – 50
- キーの最大長 – 127 文字 (Unicode)
- 値の最大長 – 255 文字 (Unicode)
- タグのキーと値は大文字と小文字が区別されます。使用できる文字は、UTF-8 で表現できる文字、スペース、および数字と、特殊文字 (+、-、=、.、_、:、/、@) です。ただし、先頭または末尾にはスペースを使用しないでください。
- タグ名または値に `aws:` プレフィックスを使用しないでください。このプレフィックスは AWS 使用のために予約されています。このプレフィックスが含まれるタグの名前または値は編集または削除できません。このプレフィックスを持つタグは、リソースあたりのタグ数の制限時には計算されません。

コンソールを使用してターゲットグループのタグを更新するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で [ターゲットグループ] を選択します。
3. ターゲットグループの名前を選択して、その詳細ページを開きます。
4. [タグ] タブを選択します。
5. タグを追加するには、[タグを追加] を選択し、タグキーとタグ値を入力します。別のタグを追加するには、[新しいタグを追加] を選択します。タグの追加を完了したら、[Save changes] (変更の保存) を選択します。
6. タグを削除するには、タグのチェックボックスを選択し、[削除] を選択します。確認を求められたら、**confirm**と入力し、[削除] を選択します。

を使用してターゲットグループのタグを更新するには AWS CLI

tag-resource コマンドと untag-resource コマンドを使用します。

VPC Lattice ターゲットグループを削除する

ターゲットグループがリスナールールの転送アクションによって参照されていない場合は、これを削除できます。ターゲットグループを削除しても、ターゲットグループに登録されたターゲットには影響が及びません。登録済み EC2 インスタンスが必要なくなった場合は停止または終了できます。

コンソールを使用してターゲットグループを削除するには

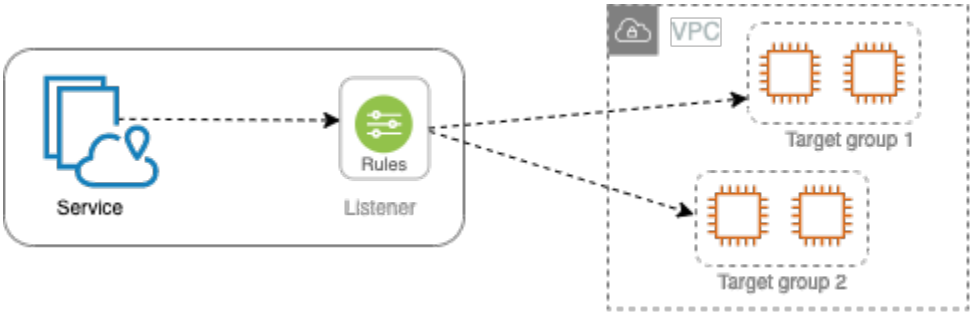
1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[ターゲットグループ] を選択します。
3. ターゲットグループのチェックボックスを選択し、[アクション]、[削除] を選択します。
4. 確認を求められたら、**confirm**と入力し、[削除] を選択します。

を使用してターゲットグループを削除するには AWS CLI

delete-target-group コマンドを使用します。

VPC Lattice サービスのリスナー

VPC Lattice サービスの使用を開始する前に、リスナーを追加する必要があります。リスナーとは、設定したプロトコルとポートを使用して接続リクエストをチェックするプロセスです。リスナーに定義したルールによって、サービスが登録済みターゲットにリクエストをルーティングする方法が決まります。



内容

リスナーの設定

リスナーは次のポートとプロトコルをサポートします。

- プロトコル: HTTP、HTTPS、TLS
- ポート: 1 ~ 65535

リスナープロトコルが HTTPS の場合、VPC Lattice は VPC Lattice が生成した FQDN に関連付けられた TLS 証明書をプロビジョニングして管理します。VPC Lattice は HTTP/1.1 と HTTP/2 の TLS をサポートしています。HTTPS リスナーを使用してサービスを設定すると、VPC Lattice では Application-Layer Protocol Negotiation (ALPN) を使用して HTTP プロトコルを自動的に決定しま

す。ALPN が存在しない場合、VPC Lattice はデフォルトで HTTP/1.1 に設定します。詳細については、「HTTPS リスナー」を参照してください。

VPC Lattice は HTTP、HTTPS、HTTP/1.1、HTTP/2 をリッスンして、これらのプロトコルおよびバージョンのいずれかでターゲットと通信します。リスナーとターゲットグループのプロトコルが一致する必要はありません。VPC Lattice はプロトコルとバージョン間のアップグレードおよびダウングレードのプロセス全体を管理します。詳細については、「プロトコルバージョン」を参照してください。

TLS リスナーを作成して、アプリケーションが VPC Lattice の代わりに暗号化されたトラフィックを復号化するようにできます。詳細については、「TLS リスナー」を参照してください。

VPC Lattice は WebSocketsネイティブにサポートしていません。ただし、TLS リスナーを使用するか、VPC Lattice リソースを介してルーティングすることで、Websocket ベースのサービスに接続できます。

VPC Lattice サービスの HTTP リスナー

リスナーとは接続リクエストをチェックするプロセスです。VPC Lattice サービスを作成するときにリスナーを定義できます。リスナーはいつでもサービスに追加できます。

このページの情報はサービスの HTTP リスナー作成の際に参照してください。他のプロトコルを使用するリスナーの作成については、HTTPS リスナー「」および「」を参照してくださいTLS リスナー。

前提条件

- 転送アクションをデフォルトのリスナールールに追加するには、利用可能な VPC Lattice ターゲットグループを指定する必要があります。詳細については、「VPC Lattice ターゲットグループを作成する」を参照してください。
- 複数のリスナーに同じターゲットグループを指定できますが、そのリスナーは同じサービスに属している必要があります。VPC Lattice サービスでターゲットグループを使用するには、他の VPC Lattice サービスのリスナーによって使用されていないことを確認する必要があります。

HTTP リスナーを追加する

リスナーとルールはいつでもサービスに追加できます。クライアントからサービスへの接続用のプロトコルとポート、デフォルトのリスナールールの VPC Lattice ターゲットグループでリスナーを設定します。詳細については、「リスナーの設定」を参照してください。

コンソールを使用した HTTP リスナーを追加するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. サービスの名前を選択して、その詳細ページを開きます。
4. [ルーティング] タブで [リスナーを追加] を選択します。
5. [リスナー名] には、カスタムのリスナー名を指定するか、リスナーのプロトコルとポートをリスナー名として使用できます。指定するカスタム名は最大 63 文字で、アカウント内のサービスごとに一意である必要があります。使用できる文字は a ~ z、0 ~ 9、- (ハイフン) です。最初または最後の文字をハイフンにしたり、別のハイフンの直後にハイフンを入れたりすることはできません。作成後に名前を変更することはできません。
6. [プロトコル : ポート] では [HTTP] を選択し、ポート番号を入力します。
7. [デフォルトアクション] ではトラフィックを受信する VPC Lattice ターゲットグループを選択し、このターゲットグループの重み付けを選択します。ターゲットグループの重み付けによって、トラフィックを受信する優先順位が決まります。例えば、2 つのターゲットグループの重み付けが同じ場合、それぞれのターゲットグループはトラフィックの半分を受信します。ターゲットグループを 1 つのみ指定した場合、トラフィックのすべてがその 1 つのターゲットグループに送信されます。

   オプションで、デフォルトアクションに別のターゲットグループを追加できます。[アクションを追加] をクリックし、ターゲットグループを選択して重み付けを指定します。
8. (オプション) 別のルールを追加するには、[ルールを追加] を選択し、ルールの名前、優先度、条件、アクションを入力します。

   各ルールに 1 ~ 100 の範囲で優先度を指定できます。リスナーは同じ優先度の複数のルールを持つことはできません。ルールは優先順位の低~高順によって評価されます。デフォルトのルールが最後に評価されます。詳細については、「リスナールール」を参照してください。
9. (オプション) タグを追加するには、[リスナータグ] を展開し、[新しいタグを追加] を選択して、タグキーとタグ値を入力します。
10. 設定を確認し、[追加] をクリックします。

を使用して HTTP リスナーを追加するには AWS CLI

create-listener コマンドを使用してデフォルトのルールを含むリスナーを作成し、create-rule コマンドを使用して追加のリスナールールを作成します。

VPC Lattice サービスの HTTPS リスナー

リスナーとは接続リクエストをチェックするプロセスです。サービスを作成するときにリスナーを定義します。リスナーはいつでも VPC Lattice のサービスに追加できます。

TLS バージョン 1.2 または TLS バージョン 1.3 を使用して VPC Lattice との HTTPS 接続を直接終了する HTTPS リスナーを作成できます。VPC Lattice は VPC Lattice が生成した完全修飾ドメイン名 (FQDN) に関連付けられた TLS 証明書をプロビジョニングして管理します。VPC Lattice は HTTP/1.1 と HTTP/2 の TLS をサポートしています。HTTPS リスナーを使用してサービスを設定すると、VPC Lattice では Application-Layer Protocol Negotiation (ALPN) 経由で HTTP プロトコルを自動的に決定します。ALPN が存在しない場合、VPC Lattice はデフォルトで HTTP/1.1 に設定します。

VPC Lattice はマルチテナンシーアーキテクチャを使用するため、同じエンドポイントで複数のサービスをホストできます。VPC Lattice はすべてのクライアントリクエストに TLS と Server Name Indication (SNI) を使用します。Encrypted Client Hello (ECH) と Encrypted Server Name Indication (ESNI) はサポートされていません。

VPC Lattice は HTTP、HTTPS、HTTP/1.1、HTTP/2 をリッスンして、これらのプロトコルおよびバージョンのいずれかでターゲットと通信します。このリスナーとターゲットグループの設定が一致している必要はありません。VPC Lattice はプロトコルとバージョン間のアップグレードおよびダウングレードのプロセス全体を管理します。詳細については、「プロトコルバージョン」を参照してください。

アプリケーションがトラフィックを復号化できるようにするには、代わりに TLS リスナーを作成します。TLS パススルーでは、VPC Lattice は TLS を終了しません。詳細については、「TLS リスナー」を参照してください。

目次

セキュリティポリシー

VPC Lattice は、TLSv1.2 プロトコルと SSL/TLS 暗号のリストを組み合わせたセキュリティポリシーを使用します。プロトコルによってクライアントとサーバーの間で安全な接続が確立され、ク

ライアントと VPC Lattice のサービスの間で受け渡しされるすべてのデータがプライベートになります。暗号とは、暗号化キーを使用してコード化されたメッセージを作成する暗号化アルゴリズムです。プロトコルは複数の暗号を使用して、データを暗号化します。接続ネゴシエーションのプロセス時に、クライアントと VPC Lattice でそれぞれサポートされる暗号とプロトコルのリストが優先される順に表示されます。デフォルトでは、サーバーのリストで最初にクライアントの暗号と一致した暗号が安全な接続用に選択されます。

VPC Lattice では、次の TLS 1.2 SSL/TLS 暗号が優先順序で使用されます。

- 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

VPC Lattice では、次の TLS 1.3 SSL/TLS 暗号もこの優先順位で使用されます。

- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256

ALPN ポリシー

Application-Layer Protocol Negotiation (ALPN) は、初期 TLS ハンドシェイク hello メッセージで送信される TLS 拡張機能です。ALPN を使用すると、アプリケーションレイヤーは HTTP/1 や HTTP/2 などのセキュアな接続上で使用するプロトコルをネゴシエートできます。

クライアントが ALPN 接続を開始すると、VPC Lattice はクライアントの ALPN 設定リストを ALPN ポリシーと比較します。クライアントが ALPN ポリシーからのプロトコルをサポートしている場合、VPC Lattice は ALPN ポリシーの設定リストに基づいて接続を確立します。サポートしていない場合、サービスで ALPN は使用されません。

VPC Lattice は次の ALPN ポリシーをサポートしています。

`HTTP2Preferred`

HTTP/1.1 よりも HTTP/2 を優先します。ALPN 設定リストは h2、http/1.1 です。

HTTPS リスナーの追加

クライアントからサービスへの接続用のプロトコルとポート、デフォルトのリスナールールのターゲットグループでリスナーを設定します。詳細については、「リスナーの設定」を参照してください。

前提条件

- 転送アクションをデフォルトのリスナールールに追加するには、利用可能な VPC Lattice ターゲットグループを指定する必要があります。詳細については、「VPC Lattice ターゲットグループを作成する」を参照してください。
- 複数のリスナーに同じターゲットグループを指定できますが、そのリスナーは同じ VPC Lattice サービスに属している必要があります。VPC Lattice サービスでターゲットグループを使用するには、他の VPC Lattice サービスのリスナーによって使用されていないことを確認する必要があります。
- VPC Lattice が提供する証明書を使用するか、独自の証明書を にインポートできます AWS Certificate Manager。詳細については、「the section called “BYOC”」を参照してください。

コンソールを使用した HTTPS リスナーを追加するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. サービスの名前を選択して、その詳細ページを開きます。
4. [ルーティング] タブで [リスナーを追加] を選択します。
5. [リスナー名] には、カスタムのリスナー名を指定するか、リスナーのプロトコルとポートをリスナー名として使用できます。指定するカスタム名は最大 63 文字で、アカウント内のサービスごとに一意である必要があります。使用できる文字は a ~ z、0 ~ 9、- (ハイフン) です。最初または最後の文字をハイフンにしたり、別のハイフンの直後にハイフンを入れたりすることはできません。作成後にリスナー名を変更することはできません。
6. [プロトコル : ポート] では [HTTPS] を選択し、ポート番号を入力します。
7. [デフォルトアクション] ではトラフィックを受信する VPC Lattice ターゲットグループを選択し、このターゲットグループの重み付けを選択します。ターゲットグループの重み付けによっ

て、トラフィックを受信する優先順位が決まります。例えば、2 つのターゲットグループの重み付けが同じ場合、それぞれのターゲットグループはトラフィックの半分を受信します。ターゲットグループを 1 つのみ指定した場合、トラフィックのすべてがその 1 つのターゲットグループに送信されます。

オプションで、デフォルトアクションに別のターゲットグループを追加できます。[アクションを追加] をクリックし、ターゲットグループを選択して重み付けを指定します。

8. (オプション) 別のルールを追加するには、[ルールを追加] を選択し、ルールの名前、優先度、条件、アクションを入力します。

   各ルールに 1 ~ 100 の範囲で優先度を指定できます。リスナーは同じ優先度の複数のルールを持つことはできません。ルールは優先順位の低~高順によって評価されます。デフォルトのルールが最後に評価されます。詳細については、「リスナールール」を参照してください。

9. (オプション) タグを追加するには、[リスナータグ] を展開し、[新しいタグを追加] を選択して、タグキーとタグ値を入力します。

10. [HTTPS リスナー証明書の設定] では、サービス作成時にカスタムドメイン名を指定しなかった場合、VPC Lattice が TLS 証明書を自動的に生成して、リスナーを通過するトラフィックを保護します。

    カスタムドメイン名でサービスを作成したものの、一致証明書を指定しなかった場合は、この時点で [カスタム SSL/TLS 証明書] から証明書を選択して指定できます。それ以外の場合は、サービス作成時に指定した証明書が既に選択されています。

11. 設定を確認し、[追加] をクリックします。

を使用して HTTPS リスナーを追加するには AWS CLI

create-listener コマンドを使用してデフォルトのルールを含むリスナーを作成し、create-rule コマンドを使用して追加のリスナールールを作成します。

VPC Lattice サービスの TLS リスナー

リスナーとは接続リクエストをチェックするプロセスです。VPC Lattice サービスを作成するときにリスナーを定義できます。リスナーはいつでもサービスに追加できます。

VPC Lattice が暗号化されたトラフィックを復号化せずにアプリケーションに渡すように、TLS リスナーを作成できます。

VPC Lattice で暗号化されたトラフィックを復号し、暗号化されていないトラフィックをアプリケーションに送信する場合は、代わりに HTTPS リスナーを作成します。詳細については、「HTTPS リスナー」を参照してください。

考慮事項

TLS リスナーには、次の考慮事項が適用されます。

- VPC Lattice サービスにはカスタムドメイン名が必要です。サービスカスタムドメイン名は、サービス名表示 (SNI) の一致として使用されます。サービスの作成時に証明書を指定した場合、その証明書は使用されません。
- TLS リスナーに許可される唯一のルールは、デフォルトのルールです。
- TLS リスナーのデフォルトアクションは、TCP ターゲットグループへの転送アクションである必要があります。
- デフォルトでは、TCP ターゲットグループのヘルスチェックは無効になっています。TCP ターゲットグループのヘルスチェックを有効にする場合は、プロトコルとプロトコルバージョンを指定する必要があります。
- TLS リスナーは、 client-hello メッセージの SNI フィールドを使用してリクエストをルーティングします。一致する条件が client-hello と完全に一致する場合は、ターゲットでワイルドカード証明書と SAN 証明書を使用できます。
- すべてのトラフィックはクライアントからターゲットに暗号化されたままであるため、VPC Lattice は HTTP ヘッダーを読み取ることができず、HTTP ヘッダーを挿入または削除することもできません。したがって、TLS リスナーには以下の制限があります。
  - 接続時間は 10 分に制限されています
  - 認証ポリシーは匿名プリンシパルに制限されます
  - Lambda ターゲットはサポートされていません
- Websocket 接続では、TLS リスナーを使用して 、VPC Lattice サービスに接続できます。以下の制限があります。
  - 接続時間は 10 分に制限されています
  - 認証ポリシーは匿名プリンシパルに制限されます
  - Lambda ターゲットはサポートされていません
- Encrypted Client Hello (ECH) はサポートされていません。
- Encrypted Server Name Indication (ESNI) はサポートされていません。

TLS リスナーを追加する

クライアントからサービスへの接続用のプロトコルとポート、デフォルトのリスナールールのターゲットグループでリスナーを設定します。詳細については、「リスナーの設定」を参照してください。

コンソールを使用して TLS リスナーを追加するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. サービスの名前を選択して、その詳細ページを開きます。
4. [ルーティング] タブで [リスナーを追加] を選択します。
5. [リスナー名] には、カスタムのリスナー名を指定するか、リスナーのプロトコルとポートをリスナー名として使用できます。指定するカスタム名は最大 63 文字で、アカウント内のサービスごとに一意である必要があります。使用できる文字は a ~ z、0 ~ 9、- (ハイフン) です。最初または最後の文字をハイフンにしたり、別のハイフンの直後にハイフンを入れたりすることはできません。作成後にリスナー名を変更することはできません。
6. [プロトコル] で [TLS] を選択します。[Port] (ポート) には、ポート番号を入力します。
7. ターゲットグループに転送するには、TCP プロトコルを使用してトラフィックを受信する VPC Lattice ターゲットグループを選択し、このターゲットグループに割り当てる重みを選択します。オプションで、別のターゲットグループを追加できます。ターゲットグループを追加 を選択し、ターゲットグループを選択して重みを入力します。
8. (オプション) タグを追加するには、[リスナータグ] を展開し、[新しいタグを追加] を選択して、タグキーとタグ値を入力します。
9. 設定を確認し、[追加] をクリックします。

を使用して TLS リスナーを追加するには AWS CLI

create-listener コマンドを使用して、デフォルトのルールでリスナーを作成します。TLS_PASSTHROUGH プロトコルを指定します。

VPC Lattice サービスのリスナールール

各リスナーにはデフォルトのルールと、定義できる追加のルールがあります。各ルールは優先度、1 つ以上のアクション、および 1 つ以上の条件で構成されています。ルールの追加や編集はいつでも行うことができます。

内容

デフォルトのルール

リスナーを作成するときは、デフォルトのルールのアクションを定義します。デフォルトのルールに条件を定義することはできません。リスナーのルールに設定された条件のいずれも満たされない場合は、デフォルトのルールのアクションが実行されます。

ルールの優先順位

各ルールには優先順位があります。ルールは優先順位の低~高順によって評価されます。デフォルトのルールが最後に評価されます。デフォルト以外のルールの優先度はいつでも変更できます。デフォルトルールの優先順位は変更できません。

ルールアクション

VPC Lattice サービスのリスナーは、転送アクションと固定レスポンスアクションをサポートしています。

転送アクション

`forward` アクションを使用して、1 つ以上の VPC Lattice ターゲットグループにリクエストをルーティングできます。`forward` アクションに複数のターゲットグループを指定する場合は、ターゲットグループごとに重みを指定する必要があります。各ターゲットグループの重みは、0 ~ 999 の値です。加重ターゲットグループを持つリスナールールと一致するリクエストは、それらの重みに基づいてこれらのターゲットグループに分散されます。たとえば、ターゲットグループを 2 つ指定し、それぞれ重みが 10 の場合、各ターゲットグループはリクエストの半分を受け取ります。2 つのターゲットグループ (1 つは重みが 10 で、もう 1 つは重みが 20) を指定すると、重みが 20 のターゲットグループは他のターゲットグループの 2 倍の数のリクエストを受信します。

固定レスポンスアクション

クライアントリクエストを破棄し、カスタムの HTTP レスポンスを返すには、`fixed-response` アクションを使用します。このアクションを使用して、404 または 500 レスポンスコードを返すことができます。

Exampleの固定レスポンスアクションの例 AWS CLI

ルールの作成時または更新時にアクションを指定できます。次のアクションは指定されたステータスコードを含む固定レスポンスを送信します。

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

ルールの条件

ルールのアクションごとにタイプと設定情報があります。ルールの条件が満たされると、アクションが実行されます。

ルールでは次の一致基準がサポートされています。

ヘッダー一致

各リクエストの HTTP ヘッダーに基づきルーティングされます。HTTP ヘッダー条件を使用して、リクエストの HTTP ヘッダーに基づいてリクエストをルーティングするルールを設定できます。標準またはカスタムの HTTP ヘッダーフィールドの名前を指定できます。ヘッダー名と一致評価では大文字と小文字は区別されません。この設定を変更するには、大文字と小文字の区別をオンにします。ワイルドカード文字はヘッダー名ではサポートされていません。ヘッダー一致ではプレフィックス、完全一致、部分一致がサポートされています。

メソッド一致

各リクエストの HTTP リクエストメソッドに基づきルーティングされます。

HTTP リクエストメソッド条件を使用して、リクエストの HTTP リクエストメソッドに基づいてリクエストをルーティングするルールを設定できます。標準またはカスタムの HTTP メソッドを指定できます。メソッド一致では大文字と小文字は区別されます。メソッド名は完全に一致する必要があります。ワイルドカード文字はサポートされていません。

パス一致

リクエスト URL のパスパターンの一致に基づきルーティングされます。

パスの条件を使用して、リクエスト内の URL に基づいてリクエストをルーティングするルールを定義できます。ワイルドカード文字はサポートされていません。パス一致ではプレフィックスと完全一致がサポートされています。

ルールの追加

リスナールールはいつでも追加できます。

コンソールを使用してリスナールールを追加する場合

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. サービスの名前を選択して、その詳細ページを開きます。
4. [ルーティング] タブで [リスナーを編集] を選択します。
5. [リスナールール] を展開し、[ルールを追加] を選択します。
6. [ルール名] にルールの名前を入力します。
7. [優先度] には 1 から 100 までの優先度を入力します。ルールは優先順位の低~高順によって評価されます。デフォルトのルールが最後に評価されます。
8. [条件] にはパス一致条件のパスパターンを入力します。各文字列の最大サイズは 200 文字です。比較では、大文字と小文字は区別されません。ワイルドカード文字はサポートされていません。

   ヘッダー一致またはメソッド一致ルール条件を追加するには、 AWS CLI または AWS SDK を使用します。
9. [アクション] では、VPC Lattice ターゲットグループを選択します。
10. [Save changes] (変更の保存) をクリックします。

を使用してルールを追加するには AWS CLI

create-rule コマンドを使用します。

ルールを更新する

リスナールールはいつでも更新できます。優先度、条件、ターゲットグループ、各ターゲットグループの重み付けを変更できます。ルールの名前は変更できません。

コンソールを使用してリスナールールを更新する場合

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. サービスの名前を選択して、その詳細ページを開きます。
4. [ルーティング] タブで [リスナーを編集] を選択します。
5. 必要に応じて、ルールの優先度、条件、アクションを変更します。
6. 更新内容を確認して [変更を保存] をクリックします。

を使用してルールを更新するには AWS CLI

update-rule コマンドを使用します。

ルールの削除

リスナーのデフォルト以外のルールはいつでも削除できます。リスナーのデフォルトのルールは削除できません。リスナーを削除すると、そのルールはすべて削除されます。

コンソールを使用してリスナールールを削除する場合

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. サービスの名前を選択して、その詳細ページを開きます。
4. [ルーティング] タブで [リスナーを編集] を選択します。
5. ルールを見つけ、[削除] をクリックします。
6. [Save changes] (変更の保存) をクリックします。

を使用してルールを削除するには AWS CLI

delete-rule コマンドを使用します。

VPC Lattice サービスのリスナーを削除する

リスナーの削除はいつでも行うことができます。リスナーを削除すると、そのルールは自動的にすべて削除されます。

コンソールを使用してリスナーを削除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] を選択します。
3. サービスの名前を選択して、その詳細ページを開きます。
4. [ルーティング] タブで [リスナーの削除] を選択します。
5. 確認を求められたら、**confirm**と入力し、[削除] を選択します。

を使用してリスナーを削除するには AWS CLI

delete-listener コマンドを使用します。

Amazon VPC Lattice の VPC リソース

VPC リソースは、組織内の他のチームや外部の独立系ソフトウェアベンダー (ISV) パートナーと共有できます。VPC リソースは、Amazon RDS データベース、ドメイン名、IP アドレスなどの AWS ネイティブリソースにすることができます。リソースは VPC またはオンプレミスネットワークにあり、負荷分散する必要はありません。を使用して AWS RAM 、リソースにアクセスできるプリンシパルを指定します。リソースにアクセスできるリソースゲートウェイを作成します。また、共有するリソースまたはリソースのグループを表すリソース設定を作成します。

リソースを共有するプリンシパルは、VPC エンドポイントを使用してこれらのリソースにプライベートにアクセスできます。リソース VPC エンドポイントを使用して、VPC Lattice サービスネットワーク内の 1 つのリソースにアクセスするか、複数のリソースをプールし、サービスネットワーク VPC エンドポイントを使用してサービスネットワークにアクセスできます。

以下のセクションでは、VPC Lattice で VPC リソースを作成および管理する方法を説明します。

トピック

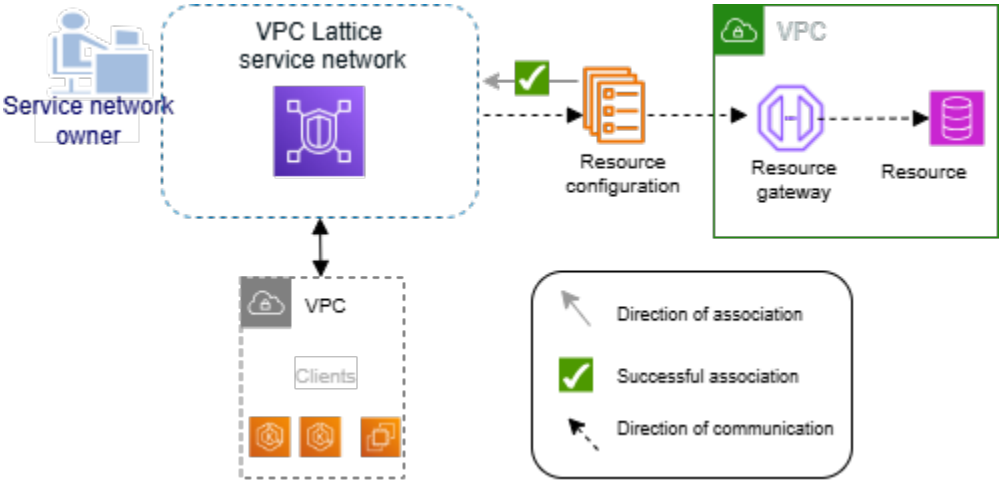
- VPC Lattice のリソースゲートウェイ
- VPC リソースのリソース設定

VPC Lattice のリソースゲートウェイ

リソースゲートウェイは、リソースが存在する VPC へのトラフィックを受信するポイントです。リソースゲートウェイは複数のアベイラビリティーゾーンを対象としています。

VPC 内のリソースを他の VPC またはアカウントからアクセスできるようにしようと計画している場合は、VPC にリソースゲートウェイが必要です。共有するすべてのリソースは、リソースゲートウェイに関連付けられます。他の VPC またはアカウント内のクライアントが VPC 内のリソースにアクセスする場合、リソースはその VPC 内のリソースゲートウェイからローカルに送られるトラフィックを認識します。トラフィックの送信元 IP アドレスは、アベイラビリティーゾーン内のリソースゲートウェイの IP アドレスです。それぞれに複数のリソースを持つ複数のリソース設定をリソースゲートウェイにアタッチできます。

次の図は、クライアントがリソースゲートウェイを介してリソースにアクセスする方法を示しています。



内容

考慮事項

リソースゲートウェイには、以下の考慮事項が適用されます。

- すべてのアベイラビリティーゾーンからリソースにアクセスできるようにするには、可能な限り多くのアベイラビリティーゾーンを対象とするリソースゲートウェイを作成する必要があります。
- VPC エンドポイントとリソースゲートウェイのアベイラビリティーゾーンが少なくとも 1 つ重複している必要があります。
- VPC には最大 100 個のリソースゲートウェイを設定できます。詳細については、「Quotas for VPC Lattice」を参照してください。
- VPC Lattice は、リソースゲートウェイに新しい ENIs を追加する場合があります。
- 共有 VPC サブネットを持つリソースゲートウェイ:
  - リソースゲートウェイは、VPC を所有するアカウントによってのみ、共有 VPC サブネットにデプロイできます。

- リソースゲートウェイのリソース設定は、リソースゲートウェイを所有するアカウントによってのみ作成できます。

セキュリティグループ

セキュリティグループをリソースゲートウェイにアタッチできます。リソースゲートウェイ用のセキュリティグループルールは、リソースゲートウェイからリソースへのアウトバウンドトラフィックを制御します。

リソースゲートウェイからデータベースリソースに送られるトラフィック向けに推奨されるアウトバウンドルール

トラフィックをリソースゲートウェイからリソースに送るには、リソースが受け入れるリスナープロトコルとポート範囲に関するアウトバウンドルールを作成する必要があります。

目的地	プロトコル	ポート範囲	コメント
##### CIDR ##	TCP	3306	リソースゲートウェイからデータベースへのトラフィックを許可します。

IP アドレスのタイプ

リソースゲートウェイには、IPv4、IPv6、またはデュアルスタックのアドレスを設定できます。以下の説明にあるように、リソースゲートウェイの IP アドレスタイプには、リソースゲートウェイのサブネット、およびリソースの IP アドレスタイプとの互換性がある必要があります。

- IPv4 – リソースゲートウェイネットワークインターフェイスに IPv4 アドレスを割り当てます。このオプションは、選択したすべてのサブネットに IPv4 アドレス範囲があり、リソースにも IPv4 アドレスがある場合にのみサポートされます。このオプションを使用すると、リソースゲートウェイ ENI あたりの IPv4 アドレスの数を設定できます。
- IPv6 – リソースゲートウェイネットワークインターフェイスに IPv6 アドレスを割り当てます。このオプションは、選択したすべてのサブネットが IPv6 限定のサブネットで、リソースにも IPv6 アドレスがある場合にのみサポートされます。このオプションを使用すると、IPv6 アドレスが自動的に割り当てられるため、管理する必要はありません。

- デュアルスタック – リソースゲートウェイネットワークインターフェイスに IPv4 アドレスと IPv6 アドレスの両方を割り当てます。このオプションは、選択したすべてのサブネットに IPv4 と IPv6 両方のアドレス範囲があり、リソースに IPv4 または IPv6 アドレスのどちらかがある場合にのみサポートされます。このオプションを使用すると、リソースゲートウェイ ENI あたりの IPv4 アドレスの数を設定できます。

リソースゲートウェイの IP アドレスタイプは、クライアント、またはリソースへのアクセス時に経由する VPC エンドポイントの IP アドレスタイプに依存しません。

ENI あたりの IPv4 アドレス

リソースゲートウェイに IPv4 またはデュアルスタックの IP アドレスタイプが設定されている場合は、リソースゲートウェイの各 ENI に割り当てられる IPv4 アドレスの数を設定できます。リソースゲートウェイを作成するときは、1 ~ 62 個の IPv4 アドレスから選択します。IPv4 アドレスの数を設定した後で値を変更することはできません。

IPv4 アドレスはネットワークアドレス変換に使用され、リソースに対する同時 IPv4 接続の最大数を決定します。各 IPv4 アドレスは、送信先 IP あたり最大 55,000 の同時接続をサポートできます。デフォルトで、すべてのリソースゲートウェイには ENI ごとに 16 個の IPv4 アドレスが割り当てられます。

リソースゲートウェイが IPv6 アドレスタイプを使用している場合、リソースゲートウェイは ENI ごとに /80 CIDR を自動的に受け取ります。この値は変更できません。接続あたりの最大送信単位 (MTU) は 8500 バイトです。

VPC Lattice でリソースゲートウェイを作成する

コンソールを使用してリソースゲートウェイを作成します。

コンソールを使用してリソースゲートウェイを作成する

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [PrivateLink と Lattice] で [リソースゲートウェイ] を選択します。
3. [リソースゲートウェイを作成] を選択します。
4. リソースゲートウェイ名には、 AWS アカウント内で一意の名前を入力します。
5. [IP アドレスタイプ] で、リソースゲートウェイの IP アドレスタイプを選択します。

- [IP アドレスタイプ] に [IPv4] または [デュアルスタック] を選択した場合は、リソースゲートウェイの ENI あたりの IPv4 アドレスの数を入力できます。

  デフォルトは、ENI あたり 16 個の IPv4 アドレスです。これは、バックエンドリソースとの接続の形成に適した IP の数です。

6. VPC の場合は、リソースゲートウェイを作成する VPC とサブネットを選択します。
7. セキュリティグループでは、最大 5 つのセキュリティグループを選択して、VPC からサービスネットワークへのインバウンドトラフィックを制御します。
8. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力してください。
9. [リソースゲートウェイを作成] を選択します。

を使用してリソースゲートウェイを作成するには AWS CLI

create-resource-gateway コマンドを使用します。

VPC Lattice でリソースゲートウェイを削除する

コンソールを使用してリソースゲートウェイを削除します。

コンソールを使用してリソースゲートウェイを削除する

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [PrivateLink と Lattice] で [リソースゲートウェイ] を選択します。
3. 削除するリソースゲートウェイのチェックボックスをオンにして、[アクション]、[削除] の順に選択します。確認を求められたら、**`confirm`**と入力し、[削除] を選択します。

を使用してリソースゲートウェイを削除するには AWS CLI

delete-resource-gateway コマンドを使用します。

VPC リソースのリソース設定

リソース設定は、他の VPC やアカウント内のクライアントがアクセスできるようにしたいリソースまたはリソースのグループを表します。リソース設定を定義することで、他の VPC やアカウント内のクライアントからの VPC 内のリソースに対するプライベートかつセキュアな一方向ネットワーク

接続を許可できます。リソース設定は、トラフィックを受信するために経由するリソースゲートウェイに関連付けられます。別の VPC からリソースにアクセスするには、リソース設定が必要です。

内容

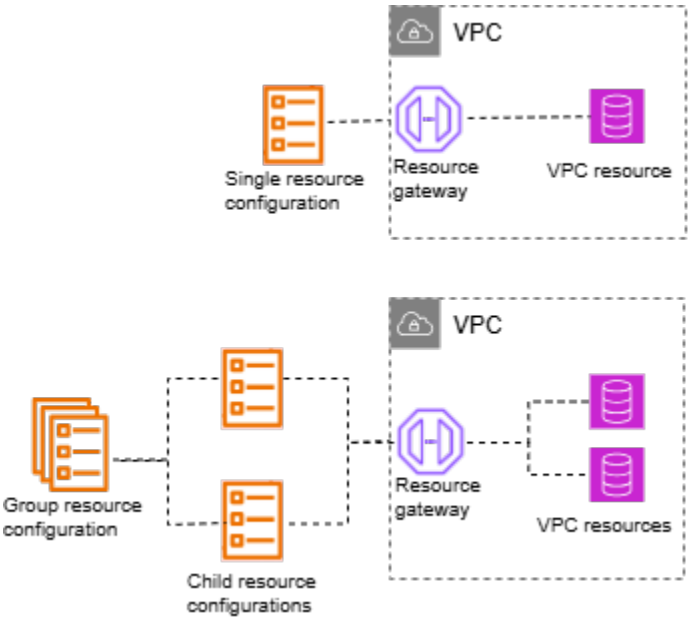
リソース設定のタイプ

リソース設定にはいくつかのタイプがあります。異なるタイプは、異なる種類のリソースを表すために役立ちます。タイプは次のとおりです。

- 単一リソース設定: IP アドレスまたはドメイン名を表します。個別に共有できます。
- グループリソース設定: 子リソース設定のコレクションです。これは、DNS エンドポイントと IP アドレスエンドポイントのグループを表すために使用できます。
- 子リソース設定: グループリソース設定のメンバーです。IP アドレスまたはドメイン名を表します。個別に共有することはできません。グループの一部としてのみ共有できます。グループに追加したり、グループから削除したりできます。追加されると、グループにアクセスできるクライアントが自動的にアクセスできるようになります。

- ARN リソース設定: AWS サービスによってプロビジョニングされるサポートされているリソースタイプを表します。グループと子の関係は自動的に処理されます。

次の図は、単一、子、およびグループのリソース設定を示しています。



プロトコル

リソース設定を作成するときは、リソースがサポートするプロトコルを定義できます。現在、TCP プロトコルのみがサポートされています。

リソースゲートウェイ

リソース設定はリソースゲートウェイに関連付けられています。リソースゲートウェイは、リソースが存在する VPC へのイングレスポイントとして機能する一連の ENI です。複数のリソース設定を同じリソースゲートウェイに関連付けることができます。他の VPCs またはアカウントのクライアントが VPC 内のリソースにアクセスすると、リソースはその VPC 内のリソースゲートウェイの IP アドレスからローカルに送信されるトラフィックを確認します。

リソースプロバイダーのカスタムドメイン名

リソースプロバイダーは、リソースコンシューマーがリソース設定にアクセスするために使用できる `example.com`などのリソース設定にカスタムドメイン名をアタッチできます。カスタムドメイン名は、リソースプロバイダーによって所有および検証することも、サードパーティーまたは AWS ドメインにすることもできます。リソースプロバイダーは、リソース設定を使用して、キャッシュクラス

ターと Kafka クラスター、TLS ベースのアプリケーション、またはその他の AWS リソースを共有できます。

リソース設定のプロバイダーには、次の考慮事項が適用されます。

- リソース設定には、1 つのカスタムドメインのみを含めることができます。
- リソース設定のカスタムドメイン名は変更できません。
- カスタムドメイン名は、すべてのリソース設定コンシューマーに表示されます。
- VPC Lattice のドメイン名検証プロセスを使用して、カスタムドメイン名を検証できます。詳細については、「」を参照してくださいthe section called “ドメインの作成と検証”。
- タイプグループと子のリソース設定では、まずグループリソース設定でグループドメインを指定する必要があります。その後、子リソース設定には、グループドメインのサブドメインであるカスタムドメインを含めることができます。グループにグループドメインがない場合は、子の任意のカスタムドメイン名を使用できますが、VPC Lattice はリソースコンシューマーの VPC 内の子ドメイン名のホストゾーンをプロビジョニングしません。

リソースコンシューマーのカスタムドメイン名

リソースコンシューマーがカスタムドメイン名を持つリソース設定への接続を有効にすると、VPC Lattice が VPC 内の Route 53 プライベートホストゾーンを管理できるようになります。リソースコンシューマーには、VPC Lattice がプライベートホストゾーンを管理できるようにするドメインの詳細なオプションがあります。

リソースコンシューマーは、リソースエンドポイント、サービスネットワークエンドポイント、またはサービスネットワーク VPC 関連付けを介してリソース設定への接続を有効にするときに、`private-dns-enabled`パラメータを設定できます。`private-dns-enabled` パラメータに加えて、コンシューマーは DNS オプションを使用して、VPC Lattice がプライベートホストゾーンを管理するドメインを指定できます。コンシューマーは、次のプライベート DNS 設定から選択できます。

`ALL_DOMAINS`

VPC Lattice は、すべてのカスタムドメイン名にプライベートホストゾーンをプロビジョニングします。

`VERIFIED_DOMAINS_ONLY`

VPC Lattice は、カスタムドメイン名がプロバイダーによって検証された場合にのみ、プライベートホストゾーンをプロビジョニングします。

VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS

VPC Lattice は、リソースコンシューマーが指定するすべての検証済みカスタムドメイン名と他のドメイン名にプライベートホストゾーンをプロビジョニングします。リソースコンシューマーは、 `private DNS specified domains`パラメータでドメイン名を指定します。

SPECIFIED_DOMAINS_ONLY

VPC Lattice は、リソースコンシューマーによって指定されたドメイン名のプライベートホストゾーンをプロビジョニングします。リソースコンシューマーは、 `private DNS specified domains` パラメータでドメイン名を指定します。

プライベート DNS を有効にすると、VPC Lattice はリソース設定に関連付けられたカスタムドメイン名のプライベートホストゾーンを VPC に作成します。デフォルトでは、プライベート DNS 設定は に設定されますVERIFIED_DOMAINS_ONLY。つまり、プライベートホストゾーンは、カスタムドメイン名がリソースプロバイダーによって検証された場合にのみ作成されます。プライベート DNS 設定を ALL_DOMAINSまたは に設定するとSPECIFIED_DOMAINS_ONLY、VPC Lattice はカスタムドメイン名の検証ステータスに関係なくプライベートホストゾーンを作成します。特定のドメインに対してプライベートホストゾーンが作成されると、VPC からそのドメインへのすべてのトラフィックは VPC Lattice を介してルーティングされます。これらのカスタムドメイン名へのトラフィックが VPC Lattice を通過する場合にのみALL_DOMAINS、VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS、、またはSPECIFIED_DOMAINS_ONLY設定を使用することをお勧めします。

リソースコンシューマーは、プライベート DNS 設定を に設定することをお勧めしますVERIFIED_DOMAINS_ONLY。これにより、コンシューマーは、VPC Lattice がリソースコンシューマーのアカウントで検証済みドメインのプライベートホストゾーンをプロビジョニングすることのみを許可することで、セキュリティ境界を強化できます。

プライベート DNS 指定ドメインのドメインを選択するには、リソースコンシューマーは などの完全修飾ドメイン名を入力するmy.example.comか、 などのワイルドカードを使用できます*.example.com。

リソース設定のコンシューマーには、次の考慮事項が適用されます。

- プライベート DNS 対応パラメータは変更できません。
- VPC でプライベートホストを作成するには、サービスネットワークリソースの関連付けでプライベート DNS を有効にする必要があります。リソース設定の場合、サービスネットワークリソー

ス関連付けのプライベート DNS 対応ステータスは、サービスネットワークエンドポイントまたはサービスネットワーク VPC 関連付けのプライベート DNS 対応ステータスを上書きします。

サービスネットワーク所有者のカスタムドメイン名

サービスネットワークリソース関連付けのプライベート DNS 対応プロパティは、サービスネットワークエンドポイントのプライベート DNS 対応プロパティとサービスネットワーク VPC 関連付けを上書きします。

サービスネットワーク所有者がサービスネットワークリソースの関連付けを作成し、プライベート DNS を有効にしない場合、VPC Lattice は、プライベート DNS がサービスネットワークエンドポイントまたはサービスネットワーク VPC の関連付けで有効であっても、サービスネットワークが接続されている VPCs でそのリソース設定のプライベートホストゾーンをプロビジョニングしません。

ARN タイプのリソース設定の場合、プライベート DNS フラグは true でイミュータブルです。

リソース定義

リソース設定では、次のいずれかの方法でリソースを識別します。

- Amazon リソースネーム (ARN) 別: AWS サービスによってプロビジョニングされるサポートされているリソースタイプは、ARN によって識別できます。Amazon RDS データベースのみがサポートされています。パブリックアクセスが可能なクラスターのリソース設定を作成することはできません。
- ドメイン名ターゲット別: パブリックに解決可能な任意のドメイン名を使用できます。ドメイン名が VPC 外にある IP をポイントする場合は、VPC 内に NAT ゲートウェイが必要です。
- IP アドレスを使用: IPv4 の場合は、10.0.0.0/8、100.64.0.0/10、172.16.0.0/12、192.168.0.0/16 の範囲からプライベート IP を指定します。IPv6 の場合は、VPC から IP を指定します。パブリック IP はサポートされていません。

ポート範囲

リソース設定を作成するときは、リクエストを受け入れるポートを定義できます。他のポートでのクライアントアクセスは許可されません。

リソースへのアクセス

コンシューマーは、VPC エンドポイントを使用する、またはサービスネットワークを経由することで、VPC からリソース設定に直接アクセスできます。コンシューマーとして、ユーザーはアカウント内のリソース設定、または AWS RAM経由で別のアカウントから共有されたリソース設定に対する VPC からのアクセスを有効にできます。

- リソース設定への直接的なアクセス

  AWS PrivateLink VPC にリソースタイプ (リソースエンドポイント) の VPC エンドポイントを作成して、VPC からリソース設定にプライベートにアクセスできます。リソースエンドポイントの作成方法に関する詳細については、「AWS PrivateLink User Guide」の「Accessing VPC resources」を参照してください。
- サービスネットワーク経由でのリソース設定へのアクセス

  リソース設定をサービスネットワークに関連付けて、VPC をサービスネットワークに接続することができます。VPC をサービスネットワークに接続するには、関連付けを使用するか、 AWS PrivateLink サービスネットワーク VPC エンドポイントを使用します。

  サービスネットワークの関連付けの詳細については、「Manage the associations for a VPC Lattice service network」を参照してください。

  サービスネットワーク VPC エンドポイントの詳細については、「AWS PrivateLink User Guide」の「Access service networks」を参照してください。

VPC でプライベート DNS が有効化されていると、同じリソース設定にリソースエンドポイントとサービスネットワークエンドポイントを作成することはできません。

サービスネットワークタイプとの関連付け

リソース設定をコンシューマーアカウントと共有する場合、例えば Account-B は を介して AWS RAM、リソース VPC エンドポイントまたはサービスネットワークを介してリソース設定に直接アクセスできます。

Account-B がサービスネットワーク経由でリソース設定にアクセスするには、リソース設定をサービスネットワークに関連付ける必要があります。サービスネットワークはアカウント間で共有できます。そのため、Account-B はそのサービスネットワーク (リソース設定が関連付けられているもの) を Account-C と共有し、Account-C がリソースにアクセスできるようにすることが可能です。

このような推移的共有を防ぐため、アカウント間で共有できるサービスネットワークにはリソース設定を追加できないと指定することができます。これを指定しておくと、Account-B がリソース設定を共有のサービスネットワークに追加したり、将来別のアカウントと共有したりすることができなくなります。

サービスネットワークのタイプ

リソース設定を Account-B などの別のアカウントと共有する場合、Account-B は AWS RAM 3 つの方法のいずれかでリソース設定で指定されたリソースにアクセスできます。

- リソースタイプの VPC エンドポイント (リソース VPC エンドポイント)。
- サービスネットワークタイプの VPC エンドポイント (サービスネットワーク VPC エンドポイント)。
- サービスネットワーク VPC の関連付け。

  サービスネットワークの関連付けを使用する場合、各リソースには、 AWS 所有およびルーティング不可の 129.224.0.0/17 ブロックからサブネットごとに IP が割り当てられます。これは、VPC Lattice が VPC Lattice ネットワーク経由でトラフィックをサービスにルーティングするために使用するマネージドプレフィックスリストに加えて割り当てられるものです。これらの IP は、どちらも VPC ルートテーブルに更新されます。

サービスネットワーク VPC エンドポイントとサービスネットワーク VPC の関連付けの場合、リソース設定は Account-B のサービスネットワークに関連付ける必要があります。 サービスネットワークはアカウント間で共有できます。そのため、Account-B はそのサービスネットワーク (リソース設定が含まれているもの) を Account-C と共有し、Account-C がリソースにアクセスできるようにすることが可能です。このような推移的共有を防ぐため、アカウント間で共有できるサービスネットワークへのリソース設定の追加を禁止することができます。禁止する場合、共有されているサービスネットワーク、または別のアカウントと共有できるサービスネットワークに Account-B がリソース設定を追加できなくなります。

を使用したリソース設定の共有 AWS RAM

リソース設定は と統合されています AWS Resource Access Manager。リソース設定は、 AWS RAM経由で別のアカウントと共有できます。リソース設定を AWS アカウントと共有すると、そのアカウントのクライアントはリソースにプライベートにアクセスできます。リソース設定は、 AWS RAMのリソース共有を使用して共有できます。

AWS RAM コンソールを使用して、追加されたリソース共有、アクセスできる共有リソース、およびリソースを共有している AWS アカウントを表示します。詳細については、「AWS RAM User Guide」の「Resources shared with you」を参照してください。

リソース設定と同じアカウントの別の VPC からリソースにアクセスするには、リソース設定を共有する必要はありません AWS RAM。

モニタリング

リソース設定でモニタリングログを有効にできます。ログの送信先を選択できます。

ドメインの作成と検証

ドメイン名の検証は、特定のドメインの所有権を証明することができるエンティティです。リソースプロバイダーとして、ドメインとそのサブドメインをリソース設定のカスタムドメイン名として使用できます。リソースコンシューマーは、リソース設定を記述するときに、カスタムドメイン名の検証ステータスを確認できます。

ドメイン検証を開始する

VPC Lattice を使用してドメイン名の検証を開始し、DNS ゾーンを使用してプロセスを完了します。

AWS マネジメントコンソール

ドメイン名の検証を開始するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの PrivateLink と Lattice で、ドメイン検証を選択します。
3. ドメイン検証の開始 を選択します。
4. ドメイン名には、所有しているドメイン名を入力します。
5. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力してください。
6. Start domain name verification を選択します。

ドメイン名の検証が正常に開始されると、VPC Lattice は `Id`と を返します`txtMethodConfig`。を使用して、ドメイン名の検証`txtMethodConfig`を完了します。

AWS CLI

次の`start-domain-verification`コマンドは、ドメイン名の検証を開始します。

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

出力は次のようになります。

```
{
    "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 Lattice は Idと を返しますtxtMethodConfig。を使用して、ドメイン名の検証txtMethodConfigを完了します。この例では、 txtMethodConfigは次のとおりです。

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

ドメイン名の検証を完了する

ドメイン名の検証を完了するには、DNS ゾーンに TXT レコードを追加します。Route 53 を使用する場合は、ドメイン名のホストゾーンを使用します。ドメイン名を検証すると、サブドメインも検証されます。たとえば、 を検証する場合example.com、追加の検証を実行beta.example.comせずに、リソース設定を alpha.example.comおよび に関連付けることができます。

を使用して TXT レコードを作成するには AWS マネジメントコンソール、「Amazon Route 53 コンソールを使用したレコードの作成」を参照してください。

AWS CLI Route 53 の を使用して TXT レコードを作成するには

1. 次のサンプルTXT-record.jsonファイルで change-resource-record-sets コマンドを使用します。

```
{
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "_11111aaaaaaaaa",
        "Type": "TXT",
        "ResourceRecords": [
          {
           "value": "vpc-lattice:1111aaaaaaa"
          }
        ]
      }
    }
  ]
}
```

2. 次の AWS CLI コマンドを使用して、前のステップの TXT レコードを Route 53 ホストゾーンに追加します。

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

hosted-zone-id を、アカウントのホストゾーンの Route 53 ホストゾーン ID に置き換えます。change-batch パラメータ値は、フォルダ (path/to/your) 内の JSON ファイル (TXT-record.json) を指します。

ドメイン名の検証ステータスを確認するには、VPC Lattice コンソールまたは get-domain-verification コマンドを使用できます。

ドメイン名を検証すると、削除されるまで検証されたままになります。DNS ゾーンから TXT レコードを削除すると、VPC Lattice は を削除verification-idし、ドメイン名を再検証する必要があります。DNS ゾーンの TXT レコードを削除すると、VPC Lattice はドメイン名の検証ステータスを に設定しますUNVERIFIED。これは、既存のリソースエンドポイント、サービスネットワークエンドポイント、またはサービスネットワーク VPC のリソース設定への関連付けには影響しません。ドメイン名を再検証するには、ドメイン名の検証プロセスを開始します。

VPC Lattice でリソース設定を作成する

リソース設定を作成します。

AWS マネジメントコンソール

コンソールを使用してライセンス設定を作成する

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [PrivateLink と Lattice] で [リソース設定] を選択します。
3. [リソース設定を作成] を選択します。
4. AWS アカウント内で一意の名前を入力します。リソース設定の作成後にこの名前を変更することはできません。
5. [設定タイプ] には、単一または子リソース用の [リソース] を選択するか、子リソースのグループ用の [リソースグループ] を選択します。
6. 以前に作成したリソースゲートウェイを選択するか、この時点でリソースゲートウェイを作成します。
7. (オプション) カスタムドメイン名を入力するには、次のいずれかを実行します。
   - Single タイプのリソース設定がある場合は、カスタムドメイン名を入力できます。リソースコンシューマーは、このドメイン名を使用してリソース設定にアクセスできます。
   - タイプグループと子のリソース設定がある場合は、まずグループリソース設定でグループドメインを指定する必要があります。次に、子リソース設定には、グループドメインのサブドメインであるカスタムドメインを含めることができます。
8. (オプション) 検証 ID を入力します。

   ドメイン名を検証する場合は、検証 ID を指定します。これにより、リソースコンシューマーは、ドメイン名を所有していることを知ることができます。
9. このリソース設定で表すリソースの識別子を選択します。
10. リソースの共有に使用するポート範囲を選択します。
11. [関連付けの設定] では、このリソース設定を共有可能なサービスネットワークに関連付けることができるかどうかを指定します。
12. [リソース設定を共有] で、このリソースにアクセスできるプリンシパルを識別するリソース共有を選択します。

13. (オプション) リソース設定との間でのリクエストと応答を監視したい場合は、[モニタリング] で [リソースアクセスログ] を有効にし、配信先を選択します。
14. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力してください。
15. [リソース設定を作成] を選択します。

AWS CLI

次の create-resource-configuration コマンドは、単一のリソース設定を作成し、カスタムドメイン名 に関連付けます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
```

次の create-resource-configuration コマンドは、グループリソース設定を作成し、カスタムドメイン名 に関連付けます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
```

次の create-resource-configuration コマンドは、子リソース設定を作成し、カスタムドメイン名に関連付けます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 のリソース設定の関連付けを管理する

アカウント内の および クライアントとリソース設定を共有するコンシューマーアカウントは、 タイプのリソースの VPC エンドポイントを直接使用するか、 タイプのサービスネットワークの VPC エンドポイントを使用してリソース設定にアクセスできます。その結果、リソース設定にはエンドポイントの関連付けとサービスネットワークの関連付けが含まれます。

サービスネットワークリソースの関連付けを管理する

サービスネットワークの関連付けを作成または削除します。

> **Note**
>
> サービスネットワークとリソース設定間の関連付けの作成中にアクセス拒否メッセージが表示された場合は、 AWS RAM ポリシーのバージョンを確認し、それがバージョン 2 であることを確認します。詳細については、 AWS RAM ユーザーガイドを参照してください。

コンソールを使用してサービスネットワークの関連付けを管理する

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [PrivateLink と Lattice] で [リソース設定] を選択します。
3. リソース設定の名前を選択して詳細ページを開きます。
4. [サービスネットワークの関連付け] タブを選択します。
5. [関連付けを作成] を選択します。
6. [VPC Lattice サービスネットワーク] からサービスネットワークを選択します。サービスネットワークを作成するには、[VPC Lattice ネットワークを作成] を選択します。
7. (オプション) タグを追加するには、[サービス関連付けのタグ] を展開して、[新しいタグを追加] を選択し、タグキーとタグ値を入力します。
8. (オプション) このサービスネットワークリソースの関連付けでプライベート DNS 名を有効にするには、プライベート DNS 名を有効にするを選択します。詳細については、「the section called “サービスネットワーク所有者のカスタムドメイン名”」を参照してください。
9. [Save changes] (変更の保存) をクリックします。
10. 関連付けを削除するには、関連付けのチェックボックスをオンにしてから、[アクション]、[削除] と選択します。確認を求められたら、**`confirm`**と入力し、[削除] を選択します。

を使用してサービスネットワークの関連付けを作成するには AWS CLI

create-service-network-resource-association コマンドを使用します。

を使用してサービスネットワークの関連付けを削除するには AWS CLI

delete-service-network-resource-association コマンドを使用します。

リソース VPC エンドポイントの関連付けを管理する

リソース設定にアクセスできるコンシューマーアカウントまたはアカウント内のクライアントは、リソース VPC エンドポイントを使用してリソース設定にアクセスできます。リソース設定にカスタムドメイン名がある場合は、プライベート DNS を有効にするを使用して、VPC Lattice がリソースエンドポイントまたはサービスネットワークエンドポイントのプライベートホストゾーンをプロビジョニングできるようにします。これにより、クライアントはドメイン名を直接カーリングしてリソース設定にアクセスできます。詳細については、「the section called “リソースコンシューマーのカスタムドメイン名”」を参照してください。

AWS マネジメントコンソール

1. エンドポイントの新しい関連付けを作成するには、左側のナビゲーションペインにある [PrivateLink と Lattice] にアクセスし、[エンドポイント] を選択します。
2. [エンドポイントを作成] を選択します。
3. VPC に接続するリソース設定を選択します。
4. VPC、サブネット、セキュリティグループを選択します。
5. (オプション) プライベート DNS を有効にして DNS オプションを設定するには、プライベート DNS 名を有効にするを選択します。
6. (オプション) VPC エンドポイントにタグ付けするには、[新しいタグを追加] を選択して、タグキーとタグ値を入力します。
7. エンドポイントの作成 を選択します。

AWS CLI

次の create-vpc-endpoint コマンドは、プライベート DNS を使用する VPC エンドポイントを作成します。プライベート DNS 設定は に設定VERIFIED_AND_SELECTEDされ、選択したドメインは example.comおよび ですexample.org。VPC Lattice は、検証済みドメインまたは example.comまたは に対してのみプライベートホストゾーンをプロビジョニングしますexample.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
```

を使用して VPC エンドポイントの関連付けを作成するには AWS CLI

create-vpc-endpoint コマンドを使用します。

を使用して VPC エンドポイントの関連付けを削除するには AWS CLI

delete-vpc-endpoint コマンドを使用します。

VPC Lattice のエンティティを共有する

Amazon VPC Lattice は AWS Resource Access Manager (AWS RAM) と統合され、サービス、リソース設定、サービスネットワークの共有を可能にします。 AWS RAM は、一部の VPC Lattice エンティティを他の AWS アカウント または と共有できるようにするサービスです AWS Organizations。では AWS RAM、リソース共有を作成して、所有しているエンティティを共有します。リソース共有は、共有するエンティティと、共有するコンシューマーを指定します。コンシューマーには以下が含まれます。

- の組織 AWS アカウント 内外に固有 AWS Organizations。
- AWS Organizationsの組織内の組織単位
- AWS Organizationsの組織全体。

詳細については AWS RAM、AWS RAM 「 ユーザーガイド」を参照してください。

内容

VPC Lattice エンティティを共有するための前提条件

- エンティティを共有するには、そのエンティティを で所有する必要があります AWS アカウント。つまり、エンティティはアカウントで割り当てまたはプロビジョニングする必要があります。自分と共有されているエンティティを共有することはできません。
- 組織または の組織単位とエンティティを共有するには AWS Organizations、 との共有を有効にする必要があります AWS Organizations。詳細については、「AWS RAM ユーザーガイド」の「AWS Organizations内でリソース共有を有効にする」を参照してください。

VPC Lattice エンティティを共有する

エンティティを共有するには、まず を使用してリソース共有を作成します AWS Resource Access Manager。リソース共有は、共有するエンティティ、共有先のコンシューマー、プリンシパルが実行できるアクションを指定します。

所有している VPC Lattice エンティティを他の と共有する場合 AWS アカウント、それらのアカウントがそれらのエンティティをアカウントのエンティティに関連付けることができます。共有エンティティに対して関連付けを作成すると、エンティティ所有者アカウントと関連付けを作成したアカウントで Amazon リソースネーム (ARN) が生成されます。したがって、エンティティ所有者と関連付けを作成したアカウントの両方が関連付けを削除できます。

の組織に属 AWS Organizations していて、組織内での共有が有効になっている場合、組織内のコンシューマーには共有エンティティへのアクセスが自動的に付与されます。それ以外の場合、コンシューマーはリソース共有への参加の招待を受け取り、招待を承諾すると共有エンティティへのアクセスが許可されます。

考慮事項

- サービスネットワーク、サービス、リソース設定の 3 種類の VPC Lattice エンティティを共有できます。
- VPC Lattice エンティティは任意の と共有できます AWS アカウント。
- VPC Lattice エンティティを個々の IAM ユーザーおよびロールと共有することはできません。
- VPC Lattice は、 サービス、リソース設定、サービスネットワークのカスタマー管理アクセス許可をサポートしています。

VPC Lattice コンソールを使用して所有しているエンティティを共有するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの VPC Lattice で、サービス、サービスネットワーク、またはリソース設定を選択します。
3. エンティティの名前を選択して詳細ページを開き、共有タブからサービスの共有、サービスネットワークの共有、またはリソース設定の共有を選択します。
4. AWS RAM リソース共有からリソース共有を選択します。リソース共有を作成するには、[RAM コンソールでリソース共有を作成] を選択します。
5. サービスの共有、サービスネットワークの共有、またはリソース設定の共有を選択します。

AWS RAM コンソールを使用して所有しているエンティティを共有するには

「AWS RAM ユーザーガイド」の「リソース共有の作成」で説明されている手順を使用します。

を使用して所有しているエンティティを共有するには AWS CLI

associate-resource-share コマンドを使用します。

VPC Lattice エンティティの共有を停止する

所有している VPC Lattice エンティティの共有を停止するには、リソース共有から削除する必要があります。既存の関連付けは、エンティティの共有を停止した後も保持されます。以前に共有したエンティティへの新しい関連付けは許可されません。エンティティ所有者または関連付け所有者のいずれかが関連付けを削除すると、両方のアカウントから削除されます。アカウント所有者がリソース共有を離れる場合は、リソース共有の所有者に、このリソースが共有されたアカウントのリストから自分のアカウントを削除するように依頼する必要があります。

VPC Lattice コンソールを使用して所有しているエンティティの共有を停止するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの VPC Lattice で、サービス、サービスネットワーク、またはリソース設定を選択します。
3. エンティティの名前を選択して、詳細ページを開きます。
4. [共有] タブでリソース共有のチェックボックスをオンにし、[削除] を選択します。

AWS RAM コンソールを使用して所有しているエンティティの共有を停止するには

「AWS RAM ユーザーガイド」の「Update a resource share」を参照してください。

を使用して所有しているエンティティの共有を停止するには AWS CLI

disassociate-resource-share コマンドを使用します。

責任と権限

共有 VPC Lattice エンティティを使用する場合、次の責任とアクセス許可が適用されます。

エンティティ所有者

- サービスネットワーク所有者は、コンシューマーが作成したサービスを変更できません。

- サービスネットワーク所有者は、コンシューマーが作成したサービスを削除できません。
- サービスネットワーク所有者は、サービスネットワークのすべてのサービスの関連付けを記述できます。
- サービスネットワーク所有者は、関連付けを作成したユーザーに関係なく、サービスネットワークに関連付けられているすべてのサービスの関連付けを解除できます。
- サービスネットワーク所有者は、サービスネットワークのすべての VPC の関連付けを記述できます。
- サービスネットワーク所有者は、コンシューマーがサービスネットワークに関連付けた VPC の関連付けを解除できます。
- サービスネットワーク所有者は、サービスネットワークのすべてのリソース設定の関連付けを記述できます。
- サービスネットワーク所有者は、関連付けを作成したユーザーに関係なく、サービスネットワークに関連付けられたリソース設定の関連付けを解除できます。
- サービスネットワーク所有者は、サービスネットワークのすべてのエンドポイントの関連付けを記述できます。
- サービスネットワーク所有者は、関連付けを作成したユーザーに関係なく、サービスネットワークに関連付けられたエンドポイントの関連付けを解除できます。
- サービス所有者は、サービスとのすべてのサービスネットワークの関連付けを記述できます。
- サービス所有者は、サービスが関連付けられているすべてのサービスネットワークからサービスの関連付けを解除できます。
- リソース設定所有者は、リソース設定とのすべてのネットワーク関連付けを記述できます。
- リソース設定所有者は、関連付けられている任意のサービスネットワークからリソース設定の関連付けを解除できます。
- VPC エンドポイント所有者は、関連付けられているサービスネットワークを記述できます。
- VPC エンドポイント所有者は、サービスネットワークからエンドポイントの関連付けを解除できます。
- 関連付けを作成したアカウントのみが、サービスネットワークと VPC 間の関連付けを更新できます。

エンティティコンシューマー

- コンシューマーは、作成していないサービスまたはリソース設定を削除することはできません。

- コンシューマーは、サービスネットワークに関連付けられたサービスまたはリソース設定のみの関連付けを解除できます。
- コンシューマーとネットワーク所有者は、サービスネットワークとサービスまたはリソース設定の間のすべての関連付けを記述できます。
- コンシューマーは、所有していないサービスまたはリソース設定のリソース設定情報を取得できません。
- コンシューマーは、すべてのサービス関連付けとリソース設定の関連付けを共有サービスネットワークと記述できます。
- コンシューマーは、サービスまたはリソース設定を共有サービスネットワークに関連付けることができます。
- コンシューマーは、共有サービスネットワークとのすべての VPC の関連付けを確認できます。
- コンシューマーは、共有サービスネットワークに VPC を関連付けることができます。
- コンシューマーは、自分がサービスネットワークに関連付けた VPC のみを解除できます。
- コンシューマーは、サービスネットワーク VPC エンドポイントを作成して、VPC を共有サービスネットワークに接続できます。
- コンシューマーは、VPC を共有サービスネットワークに接続するために作成したサービスネットワーク VPC エンドポイントのみを削除できます。
- 共有サービスのコンシューマーは、自分が所有していないサービスネットワークにサービスを関連付けることはできません。
- 共有サービスネットワークのコンシューマーは、自分が所有していない VPC やサービスを関連付けることはできません。
- 共有リソース設定のコンシューマーは、リソース設定を所有していないサービスネットワークに関連付けることはできません。
- 共有サービスネットワークのコンシューマーは、所有していない VPC、サービス、またはリソース設定を関連付けることはできません。
- コンシューマーは、共有されているサービス、サービスネットワーク、またはリソース設定を記述できます。
- コンシューマーは、両方が共有されている場合、2 つのエンティティを関連付けることはできません。

クロスアカウントイベント

エンティティ所有者とコンシューマーが共有エンティティに対してアクションを実行すると、それらのアクションはクロスアカウントイベントとして記録されます AWS CloudTrail。

`CreateServiceNetworkResourceAssociationBySharee`

エンティティコンシューマーが共有エンティティで CreateServiceNetworkResourceAssociation を呼び出すと、エンティティ所有者に送信されます。呼び出し元がリソース設定を所有している場合、イベントはサービスネットワークの所有者に送信されます。呼び出し元がサービスネットワークを所有している場合、イベントはリソース設定の所有者に送信されます。

`CreateServiceNetworkServiceAssociationBySharee`

エンティティコンシューマーが共有エンティティと CreateServiceNetworkServiceAssociation を呼び出すと、エンティティ所有者に送信されます。発信者がサービスを所有している場合、イベントはサービスネットワークの所有者に送信されます。発信者がサービスネットワークを所有している場合、イベントはサービスの所有者に送信されます。

`CreateServiceNetworkVpcAssociationBySharee`

エンティティコンシューマーが共有サービスネットワークで CreateServiceNetworkVpcAssociation を呼び出すと、エンティティ所有者に送信されます。

`DeleteServiceNetworkResourceAssociationByOwner`

エンティティ所有者が共有エンティティで DeleteServiceNetworkResourceAssociation を呼び出すと、関連付け所有者に送信されます。呼び出し元がリソース設定を所有している場合、イベントはサービスネットワーク関連付けの所有者に送信されます。呼び出し元がサービスネットワークを所有している場合、イベントはリソース関連付けの所有者に送信されます。

`DeleteServiceNetworkResourceAssociationBySharee`

エンティティコンシューマーが共有エンティティで DeleteServiceNetworkResourceAssociation を呼び出すと、エンティティ所有者に送信されます。呼び出し元がリソース設定を所有している場合、イベントはサービスネットワークの所有者に送信されます。呼び出し元がサービスネットワークを所有している場合、イベントはリソース設定の所有者に送信されます。

`DeleteServiceNetworkServiceAssociationByOwner`

エンティティ所有者が共有エンティティで DeleteServiceNetworkServiceAssociation を呼び出すと、関連付け所有者に送信されます。発信者がサービスを所有している場合、イベントはサービ

スネットワークの関連付けの所有者に送信されます。発信者がサービスネットワークを所有している場合、イベントはサービスの関連付けの所有者に送信されます。

`DeleteServiceNetworkServiceAssociationBySharee`

エンティティコンシューマーが共有エンティティと DeleteServiceNetworkServiceAssociation を呼び出すと、エンティティ所有者に送信されます。発信者がサービスを所有している場合、イベントはサービスネットワークの所有者に送信されます。発信者がサービスネットワークを所有している場合、イベントはサービスの所有者に送信されます。

`DeleteServiceNetworkVpcAssociationByOwner`

エンティティ所有者が共有サービスネットワークで DeleteServiceNetworkVpcAssociation を呼び出すと、関連付け所有者に送信されます。

`DeleteServiceNetworkVpcAssociationBySharee`

エンティティコンシューマーが共有サービスネットワークで DeleteServiceNetworkVpcAssociation を呼び出すと、エンティティ所有者に送信されます。

`GetServiceBySharee`

エンティティコンシューマーが共有サービスを使用して GetService を呼び出すと、エンティティ所有者に送信されます。

`GetServiceNetworkBySharee`

エンティティコンシューマーが共有サービスネットワークを使用して GetServiceNetwork を呼び出すと、エンティティ所有者に送信されます。

`GetServiceNetworkResourceAssociationBySharee`

エンティティコンシューマーが共有エンティティと GetServiceNetworkResourceAssociation を呼び出すと、エンティティ所有者に送信されます。呼び出し元がリソース設定を所有している場合、イベントはサービスネットワークの所有者に送信されます。呼び出し元がサービスネットワークを所有している場合、イベントはリソース設定の所有者に送信されます。

`GetServiceNetworkServiceAssociationBySharee`

エンティティコンシューマーが共有エンティティと GetServiceNetworkServiceAssociation を呼び出すと、エンティティ所有者に送信されます。発信者がサービスを所有している場合、イベントはサービスネットワークの所有者に送信されます。発信者がサービスネットワークを所有している場合、イベントはサービスの所有者に送信されます。

GetServiceNetworkVpcAssociationBySharee

エンティティコンシューマーが共有サービスネットワークで GetServiceNetworkVpcAssociation を呼び出すと、エンティティ所有者に送信されます。

以下は、CreateServiceNetworkServiceAssociationBySharee イベントのエントリ例です。


```
{
    "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"
}
```

の VPC Lattice Oracle Database@AWS

VPC Lattice は、 Oracle Database@AWS (ODB) の AWS マネージドサービス統合を強化し、ODB ネットワーク、 AWS VPCs、オンプレミス間の接続を簡素化します。この接続をサポートするために、VPC Lattice はユーザーに代わって次のエンティティをプロビジョニングします。

デフォルトのサービスネットワーク

デフォルトのサービスネットワークは命名規則を使用します `default-odb-network-randomHash`

デフォルトのサービスネットワークエンドポイント

この AWS リソースの名前はありません。

リソースゲートウェイ

リソースゲートウェイは命名規則を使用します `default-odb-network-randomHash`

VPC Lattice は、ODB ネットワークへの AWS マネージド統合と呼ばれるマネージドサービス統合をサポートします。デフォルトでは、Oracle Cloud Infrastructure (OCI) Managed Backup to Amazon S3 が有効になっています。Amazon S3 とゼロ ETL へのセルフマネージドアクセスを有効にすることを選択できます。

ODB ネットワークを作成したら、 AWS マネジメントコンソール または を使用してプロビジョニングされたリソースを表示できます AWS CLI。次のコマンド例では、ODB ネットワークのデフォルトのマネージド統合と、このサービスネットワークに必要なその他のリソースを一覧表示します。

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

考慮事項

VPC Lattice には、次の考慮事項が適用されます Oracle Database@AWS。

- デフォルトのサービスネットワーク、サービスネットワークエンドポイント、リソースゲートウェイ、または VPC Lattice によってプロビジョニングされた ODB マネージド統合を削除することはできません。これらのエンティティを削除するには、ODB ネットワークを削除するか、マネージド統合を無効にします。

- クライアントは、ODB ネットワーク内のマネージド統合にのみアクセスできます。VPCs 内など、ODB ネットワーク外のクライアントは、これらのマネージド統合を使用して S3 またはゼロ ETL にアクセスすることはできません。
- VPC Lattice によってプロビジョニングされた ODB ネットワーク外のマネージド統合に接続することはできません。
- Amazon S3 へのすべてのトラフィックはデフォルトのサービスネットワークエンドポイントを通過し、リソースにアクセスするための標準処理料金が適用されます。すべてのゼロ ETL トラフィックはリソースゲートウェイを経由し、共有するリソースの標準データ処理料金が適用されます。詳細については、「VPC Lattice の料金」を参照してください。
- Oracle Database@AWS マネージド統合には時間単位の料金はかかりません。
- VPC Lattice によってプロビジョニングされたリソースは、他のサービスネットワークと同様に管理できます。デフォルトのサービスネットワークを他の AWS アカウント または組織と共有し、新しいエンドポイント、VPC 関連付け、VPC Lattice サービスおよびリソースをデフォルトのネットワークに追加できます。
- VPC Lattice が Oracle Database@AWS リソースをプロビジョニングするには、次のアクセス許可が必要です。

  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"
       ]
      }
     }
   }
    ]
}
```

VPC Lattice を に使用するには Oracle Database@AWS、VPC Lattice のサービスネットワーク、サービスネットワークの関連付け、リソースゲートウェイに精通していることをお勧めします。

トピック

- the section called “Oracle Cloud Infrastructure (OCI) Managed Backup to Amazon S3”
- the section called “Amazon S3 アクセス”
- the section called “Amazon Redshift のゼロ ETL”
- the section called “VPC Lattice エンティティにアクセスして共有する”

Oracle Cloud Infrastructure (OCI) Managed Backup to Amazon S3

Oracle Database@AWS データベースを作成すると、VPC Lattice は というリソース設定を作成しますodb-managed-s3-backup-access。このリソース設定は、Amazon S3 へのデータベースの OCI マネージドバックアップを表し、OCI が所有する Amazon S3 バケットへの接続のみを有効にします。ODB ネットワークと S3 間のトラフィックが Amazon ネットワークから出ることはありません。

Amazon S3 アクセス

Amazon S3 への OCI マネージドバックアップに加えて、ODB ネットワークから Amazon S3 へのアクセスを可能にするマネージド統合を作成できます。Amazon S3 Access マネージド統合を有効にするように Oracle Database@AWS ネットワークを変更すると、VPC Lattice はデフォルトのサービスネットワークodb-s3-accessで というリソース設定をプロビジョニングします。この統合を使用して、セルフマネージドバックアップや復元など、独自のニーズに合わせて Amazon S3 にアクセスできます。認証ポリシーを指定することで、境界制御を確立できます。

考慮事項

Amazon S3 Access マネージド統合に関する考慮事項は次のとおりです。

- ODB ネットワークに対して作成できる Amazon S3 Access マネージド統合は 1 つだけです。
- このマネージド統合により、ODB ネットワークからのみ Amazon S3 にアクセスでき、デフォルトのサービスネットワークの他の VPC 関連付けやサービスネットワークエンドポイントからはアクセスできません。
- 異なる AWS リージョンの S3 バケットにはアクセスできません。

Amazon S3 Access マネージド統合を有効にする

次のコマンドを使用して、Amazon S3 Access マネージド統合を有効にします。

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

認証ポリシーによる安全なアクセス

ODB API を使用して認証ポリシーを定義することで、S3 バケットへのアクセスを保護できます。次のポリシー例では、特定の組織が所有する特定の S3 バケットへのアクセスを許可します。

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**
>
> `aws:SourceVpc`、`aws:SourceVpce`、および `aws:VpcSourceIp`条件キーは、ODB マネージド統合を使用する場合、S3 バケットポリシーではサポートされていません。

Amazon Redshift のゼロ ETL

VPC Lattice によってプロビジョニングされたサービスネットワークを使用して、ゼロ ETL を有効にできます。このマネージド型統合は、ODB ネットワークデータベースを Amazon Redshift に接

続して、異なるデータベース間でデータを分析できるようにします。 AWS Glue 統合 APIs、ODB APIs を使用してマネージド統合を有効にし、ネットワークパスを設定できます。詳細については、「Amazon Redshift とのゼロ ETL 統合」を参照してください。

考慮事項

マネージドゼロ ETL 統合に関する考慮事項は次のとおりです。

- マネージドゼロ ETL 統合を有効にすると、ゼロ ETL を使用して ODB ネットワークのインスタンスにのみアクセスできます。サービスネットワークに関連付けられた他のサービスとリソースは、ゼロ ETL から分離されます。

VPC Lattice エンティティにアクセスして共有する

VPCsを使用して、ODB ネットワークを VPC 内のサービス、リソース、その他のクライアントに接続することもできます。これらの接続オプションは、VPC Lattice によってプロビジョニングされたデフォルトのサービスネットワーク、リソースゲートウェイ、およびサービスネットワークエンドポイントを利用します。

VPC Lattice サービスとリソースにアクセスする

他のエンティティにアクセスするには、所有している、または共有されているサービスまたはリソースをデフォルトのサービスネットワークに関連付けます。ODB ネットワークのクライアントは、デフォルトのサービスネットワークエンドポイントを介してサービスまたはリソースにアクセスできます。

考慮事項

以下は、他の VPC Lattice エンティティに接続するための考慮事項です。

- 新しいサービスネットワークエンドポイント、VPC 関連付け、VPC Lattice リソースとサービスをサービスネットワークに追加できますが、ODB ネットワークに代わって VPC Lattice によってプロビジョニングされたリソースを変更することはできません。これらは Oracle Database@AWS APIsを通じて管理する必要があります。

VPC Lattice 経由で ODB ネットワークを共有する

ODB ネットワークリソースは、他の VPCs、アカウント、またはオンプレミスのクライアントと共有できます。開始するには、共有するリソースのリソース設定を作成します。リソース設定で

は、ODB ネットワークのデフォルトのリソースゲートウェイを使用する必要があります。その後、リソースをデフォルトのサービスネットワークに関連付けることができます。

他の VPCs のクライアント、またはサービスネットワークを共有 AWS アカウント したクライアントは、独自のサービスネットワークエンドポイントまたは VPC の関連付けを介してこれらのリソースにアクセスできます。詳細については、「the section called “関連付けを管理する”」を参照してください。

考慮事項

以下は、ODB ネットワークを共有するための考慮事項です。

- ODB ネットワークインスタンスは IP ベースのリソースとしてのみ共有することをお勧めします。
- VPC Lattice は、OCI の単一クライアントアクセスネーム (SCAN) リスナー DNS をサポートしていません。

Amazon VPC Lattice におけるセキュリティ

のクラウドセキュリティが最優先事項 AWS です。お客様は AWS 、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを活用できます。

ユーザーは、このインフラストラクチャでホストされるコンテンツに対する管理を維持する責任があります。責任共有モデルではこれをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ – AWS は、 で AWS サービスを実行するインフラストラクチャを保護する責任を担います AWS クラウド。 は、お客様が安全に使用できるサービス AWS も提供します。サードパーティーの監査者は、AWS コンプライアンスプログラムコンプライアンスプログラムの一環として、当社のセキュリティの有効性を定期的にテストおよび検証。Amazon VPC Lattice に適用されるコンプライアンスプログラムの詳細については、「コンプライアンスプログラムAWS による対象範囲内のサービスコンプライアンスプログラム」を参照してください。
- クラウドのセキュリティ – ユーザーにはこのインフラストラクチャでホストされているコンテンツに対する管理を行う責任があります。また、ユーザーは、データの機密性、会社の要件、適用される法律や規制など、その他の要因についても責任を負います。

このドキュメントは VPC Lattice を使用する際に責任共有モデルを適用する方法を理解するための一助となります。以下のトピックでは、VPC Lattice を設定して、セキュリティとコンプライアンスの目標を達成する方法を説明します。また、VPC Lattice AWS サービス、サービスネットワーク、リソース設定のモニタリングと保護に役立つ他の サービスの使用方法についても説明します。

内容

VPC Lattice サービスへのアクセスを管理する

VPC Lattice はデフォルトで安全です。これは、どのサービスとリソース設定でどの VPCs へのアクセスを提供するかを明確にする必要があるためです。VPC 関連付けまたはサービスネットワークタイプの VPC エンドポイントを介してサービスにアクセスできます。マルチアカウントシナリオでは、 AWS Resource Access Managerを使用して、アカウント境界間でサービス、リソース設定、サービスネットワークを共有できます。

VPC Lattice は複数のネットワークレイヤーでdefense-in-depth戦略を実装できるフレームワークを提供します。

- 第 1 レイヤー – サービスネットワークとのサービス、リソース、VPC、VPC エンドポイントの関連付け。VPC は、関連付けまたは VPC エンドポイントを介してサービスネットワークに接続できます。VPC がサービスネットワークに接続されていない場合、VPC 内のクライアントは、サービスネットワークに関連付けられているサービスおよびリソース設定にアクセスできません。
- 第 2 レイヤー – セキュリティグループやネットワーク ACL など、サービスネットワークに対するネットワークレベルでのオプションのセキュリティ保護。これらを使用することで、VPC 内のすべてのクライアントではなく、VPC 内の特定のクライアントグループへのアクセスを許可できます。
- 第 3 レイヤー – オプションの VPC Lattice 認証ポリシー。認証ポリシーはサービスネットワークと個々のサービスに適用できます。通常、サービスネットワークの認可ポリシーはネットワークまたはクラウド管理者によって運用され、粗粒度の認可が実装されます。例えば、 AWS Organizations 内の特定の組織からの認証されたリクエストのみを許可します。サービスレベルの認可ポリシーでは、通常、サービス所有者がきめ細かい制御を設定します。このような制御はサービスネットワークレベルで適用される粗粒度の認可よりも厳しい場合があります。

> **Note**
>
> サービスネットワークの認証ポリシーは、サービスネットワークのリソース設定には適用されません。

アクセスコントロール方法

- 認証ポリシー
- セキュリティグループ
- ネットワーク ACL

認証ポリシーを使用して VPC Lattice サービスへのアクセスを制御する

VPC Lattice 認証ポリシーは、指定したプリンシパルによるサービスのグループまたは特定のサービスへのアクセスを制御するために、サービスネットワークまたはサービスにアタッチする IAM ポリシードキュメントです。アクセスを制御する各サービスネットワークまたはサービスに認証ポリシーを 1 つアタッチできます。

> **Note**
>
> サービスネットワークの認証ポリシーは、サービスネットワークのリソース設定には適用されません。

認証ポリシーは IAM アイデンティティベースのポリシーとは異なります。IAM アイデンティティベースのポリシーは、IAM ユーザー、グループ、ロールにアタッチされ、実行できるアクションとリソースを定義します。認証ポリシーはサービスとサービスネットワークにアタッチされます。認可が正常に完了するためには、認可ポリシーとアイデンティティベースのポリシーの両方において、明示的な許可ステートメントが必要です。詳細については、「認可の仕組み」を参照してください。

AWS CLI および コンソールを使用して、サービスおよびサービスネットワークの認証ポリシーを表示、追加、更新、または削除できます。認証ポリシーを追加、更新、または削除すると、準備が完了するまでに数分かかる場合があります。を使用する場合は AWS CLI、正しいリージョンにいることを確認してください。プロファイルのデフォルトリージョンを変更するか、 コマンドで `--region`パラメータを使用できます。

内容

認証ポリシーの使用を開始するには、手順に沿ってサービスネットワークに適用する認証ポリシーを作成します。制限の厳しいアクセス許可を他のサービスには適用しない場合には、オプションで個別のサービスに認証ポリシーを設定できます。

認証ポリシーを使用してサービスネットワークへのアクセスを管理する

以下の AWS CLI タスクでは、認証ポリシーを使用してサービスネットワークへのアクセスを管理する方法を示します。コンソールでの手順については、「VPC Lattice のサービスネットワーク」を参照してください。

タスク

- 認証ポリシーをサービスネットワークに追加する
- サービスネットワークの認証タイプを変更する
- 認証ポリシーをサービスネットワークから削除する

認証ポリシーをサービスネットワークに追加する

このセクションのステップに従って、 を使用して以下 AWS CLI を行います。

- IAM を使用してサービスネットワークのアクセスコントロールを有効にします。
- 認証ポリシーをサービスネットワークに追加します。認証ポリシーを追加しない場合、すべてのトラフィックでアクセス拒否エラーが発生します。

アクセスコントロールを有効にし、認証ポリシーを新しいサービスネットワークに追加する方法

1. サービスネットワークでアクセスコントロールを有効にして認証ポリシーを使用できるようにするには、create-service-network コマンドを使用して `--auth-type` オプションを値 `AWS_IAM` と指定します。

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

成功すると、コマンドは以下のような出力を返します。

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

```
    "name": "Name"
}
```

2. put-auth-policy コマンドを使用して、認証ポリシーを追加するサービスネットワークの ID と追加する認証ポリシーを指定します。

   例えば、次のコマンドを使用して、ID *sn-0123456789abcdef0* でサービスネットワークの認証ポリシーを作成します。

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

   JSON を使用してポリシー定義を作成します。詳細については、「認証ポリシーの一般的な要素」を参照してください。

   成功すると、コマンドは以下のような出力を返します。

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

アクセスコントロールを有効にし、既存のサービスネットワークに認証ポリシーを追加する方法

1. サービスネットワークでアクセスコントロールを有効にして認証ポリシーを使用できるようにするには、update-service-network コマンドを使用して `--auth-type` オプションを値 `AWS_IAM` と指定します。

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

   成功すると、コマンドは以下のような出力を返します。

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

2. put-auth-policy コマンドを使用して、認証ポリシーを追加するサービスネットワークの ID と追加する認証ポリシーを指定します。

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

JSON を使用してポリシー定義を作成します。詳細については、「認証ポリシーの一般的な要素」を参照してください。

成功すると、コマンドは以下のような出力を返します。

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

サービスネットワークの認証タイプを変更する

サービスネットワークの認証ポリシーを無効にする方法

update-service-network コマンドを使用して --auth-type オプションを値 NONE と指定します。

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

後ほど認証ポリシーを再度有効にする必要がある場合は、--auth-type オプションを AWS_IAM と指定してこのコマンドを実行します。

認証ポリシーをサービスネットワークから削除する

認証ポリシーをサービスネットワークから削除する方法

delete-auth-policy コマンドを使用します。

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

サービスネットワークの認証タイプを NONE に変更する前に認証ポリシーを削除すると、リクエストはエラーとなります。

認証ポリシーを使用してサービスへのアクセスを管理する

以下の AWS CLI タスクでは、認証ポリシーを使用してサービスへのアクセスを管理する方法を示します。コンソールでの手順については、「VPC Lattice のサービス」を参照してください。

タスク

- 認証ポリシーをサービスに追加する
- サービスの認証タイプを変更する
- 認証ポリシーをサービスから削除する

認証ポリシーをサービスに追加する

を使用して以下のステップを実行します AWS CLI 。

- IAM を使用してサービスのアクセスコントロールを有効にします。
- 認証ポリシーをサービスに追加します。認証ポリシーを追加しない場合、すべてのトラフィックでアクセス拒否エラーが発生します。

アクセスコントロールを有効にし、認証ポリシーを新しいサービスに追加する方法

1. サービスでアクセスコントロールを有効にして認証ポリシーを使用できるようにするには、create-service コマンドを使用して `--auth-type` オプションを値 `AWS_IAM` と指定します。

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

成功すると、コマンドは以下のような出力を返します。

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

```
}
```

2. put-auth-policy コマンドを使用して、認証ポリシーを追加するサービスの ID と追加する認証ポリシーを指定します。

   例えば、次のコマンドを使用して、ID *svc-0123456789abcdef0* でサービスの認証ポリシーを作成します。

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

   JSON を使用してポリシー定義を作成します。詳細については、「認証ポリシーの一般的な要素」を参照してください。

   成功すると、コマンドは以下のような出力を返します。

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

アクセスコントロールを有効にし、既存のサービスに認証ポリシーを追加する方法

1. サービスでアクセスコントロールを有効にして認証ポリシーを使用できるようにするには、update-service コマンドを使用して `--auth-type` オプションを値 `AWS_IAM` と指定します。

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

   成功すると、コマンドは以下のような出力を返します。

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

2. put-auth-policy コマンドを使用して、認証ポリシーを追加するサービスの ID と追加する認証ポリシーを指定します。

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

JSON を使用してポリシー定義を作成します。詳細については、「認証ポリシーの一般的な要素」を参照してください。

成功すると、コマンドは以下のような出力を返します。

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

サービスの認証タイプを変更する

サービスの認証ポリシーを無効にする方法

update-service コマンドを使用して --auth-type オプションを値 NONE と指定します。

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

後ほど認証ポリシーを再度有効にする必要がある場合は、--auth-type オプションを AWS_IAM と指定してこのコマンドを実行します。

認証ポリシーをサービスから削除する

認証ポリシーをサービスから削除する方法

delete-auth-policy コマンドを使用します。

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

サービスの認証タイプを NONE に変更する前に認証ポリシーを削除すると、リクエストはエラーとなります。

サービスに認証されたリクエストを必要とする認証ポリシーを有効にする場合、そのサービスのすべてのリクエストには、Signature Version 4 (SigV4)を使用して計算された有効なリクエストの署名が含まれている必要があります。詳細については、「Amazon VPC Lattice の SIGv4 認証リクエスト」を参照してください。

認証ポリシーの一般的な要素

VPC Lattice 認証ポリシーは IAM ポリシーと同じ構文を使用して指定されます。詳細については、「IAM ユーザーガイド」の「アイデンティティベースおよびリソースベースのポリシー」を参照してください。

認証ポリシーには、次の要素が含まれます。

- プリンシパル – ステートメントのアクションとリソースへのアクセスが許可されているユーザーまたはアプリケーションを指します。認証ポリシーでは、プリンシパルはこのアクセス許可の被付与者である IAM エンティティを指します。プリンシパルは IAM エンティティとして認証され、サービスネットワークのサービスのように、特定のリソースまたはリソースのグループにリクエストを送信します。

  リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または AWS サービスを含めることができます。詳細については、「IAM ユーザーガイド」の「AWS JSON ポリシーの要素: プリンシパル」を参照してください。
- 効果 - 指定されたプリンシパルが特定のアクションをリクエストするときの効果を指定します。`Allow` または `Deny` のいずれかとなります。デフォルトでは、IAM を使用してサービスまたはサービスネットワークのアクセスコントロールを有効にした場合、プリンシパルにはサービスまたはサービスネットワークにリクエストをする権限がありません。
- アクション – アクセス許可を付与または拒否する特定の API アクション。VPC Lattice は`vpc-lattice-svcs`、プレフィックスを使用するアクションをサポートしています。詳細については、「サービス認可リファレンス」の「Amazon VPC Lattice Services で定義されるアクション」を参照してください。
- リソース – アクションによって影響を受けるサービスです。
- 条件 – 条件はオプションです。ポリシーが有効になるタイミングを制御するために使用できます。詳細については、「サービス認可リファレンス」の「Amazon VPC Lattice Services の条件キー」を参照してください。

認証ポリシーを作成し管理するときは、IAM Policy Generator を使用することもできます。

要件

JSON のポリシーには改行または空白行を含めないでください。

認証ポリシーのリソース形式

特定のリソースへのアクセスを制限するには、次の例のとおり <serviceARN>/<path> パターンと一致するスキーマを使用し Resource 要素をコーディングする認証ポリシーを作成します。

プロトコル	例
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/*" • "Resource": "arn:aws:vpc-lattice:us-west-2:1234567890:service/svc-0123456789abcdef0/*"

<serviceARN> には、次の Amazon リソースネーム (ARN)リソース形式を使用します。

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

例えば、次のようになります。

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

認証ポリシーで使用できる条件キー

アクセスは認証ポリシーの条件要素の条件キーによってさらに細かくコントロールできます。これらの条件キーはプロトコルと、リクエストが Signature Version 4 (SigV4) または匿名のどちらで署名されているかによって、評価の対象となります。条件キーは大文字と小文字が区別されます。

AWS には、 `aws:PrincipalOrgID`や などのアクセスを制御するために使用できるグローバル条件キーが用意されています`aws:SourceIp`。 AWS グローバル条件キーのリストを確認するには、「IAM ユーザーガイド」のAWS 「グローバル条件コンテキストキー」を参照してください。

次のストーリーでは、VPC Lattice 条件キーを一覧表示します。詳細については、「サービス認可リファレンス」の「Amazon VPC Lattice Services の条件キー」を参照してください。

条件キー	説明	例	匿名の (認証されていない) 発信者による利用可否	gRPC での利用可否
`vpc-lattice-svcs:Port`	リクエストが行われたサービスポートによりアクセスをフィルタリング	80	はい	はい
`vpc-lattice-svcs:RequestMethod`	リクエスト方法によりアクセスをフィルタリング	GET	はい	常に POST
`vpc-lattice-svcs:RequestPath`	リクエスト URL のパス部分でアクセスをフィルタリング	/path	はい	はい
`vpc-lattice-svcs:RequestHeader/` *`header-name`* `:` *`value`*	リクエストヘッダーのヘッダー名と値のペアによりアクセスをフィルタリング	content-type: application/json	はい	はい
`vpc-lattice-svcs:RequestQueryString/` *`key-name`*`:` *`value`*	リクエスト URL 内のクエリ文字列キーと値のペアによりアクセスをフィルタリング	quux: [corge, grault]	はい	なし

条件キー	説明	例	匿名の (認証されていない) 発信者による利用可否	gRPC での利用可否
vpc-lattice-svcs:ServiceNetworkArn	リクエストを受け取ったサービスのサービスネットワークの ARN によりアクセスをフィルタリング	arn:aws:vpc-lattice:us-west-2:123456789012:servicenetwork/sn-0123456789abcdef0	はい	はい
vpc-lattice-svcs:ServiceArn	リクエストを受け取ったサービスの ARN によりアクセスをフィルタリング	arn:aws:vpc-lattice:us-west-2:123456789012:service/svc-0123456789abcdef0	はい	はい
vpc-lattice-svcs:SourceVpc	リクエストが行われた VPC によりアクセスをフィルタリング	vpc-1a2b3c4d	はい	はい
vpc-lattice-svcs:SourceVpcOwnerAccount	リクエストが行われた所有アカウントの VPC によりアクセスをフィルタリング	123456789012	はい	はい

リソースタグ

タグは、ユーザーが割り当てるか、 AWS リソース AWS に割り当てるメタデータラベルです。各 タグは 2 つの部分で構成されます:

- タグキー (例： CostCenter、Environment、または Project)。タグキーでは、大文字と小文字が区別されます。
- タグ値として知られるオプションのフィールド (例： 111122223333 または Production)。タグ値を省略すると、空の文字列を使用した場合と同じになります。タグキーと同様に、タグ値でも大文字と小文字が区別されます。

タグ付けの詳細については、「タグを使用した AWS リソースへのアクセスの制御」を参照してください。

aws:ResourceTag/key AWS グローバル条件コンテキストキーを使用して、認証ポリシーでタグを使用できます。

次のポリシー例では、タグ を持つ サービスへのアクセスを許可しますEnvironment=Gamma。このポリシーでは、ハードコーディングサービス ARNs または IDs のないサービスを参照できます。

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

プリンシパルタグ

発信者の ID にアタッチされたタグに基づいて、サービスとリソースへのアクセスを制御できます。VPC Lattice は、 aws:PrincipalTag/context変数を使用したユーザー、ロール、またはセッションタグのプリンシパルタグに基づくアクセスコントロールをサポートします。詳細については、「IAM プリンシパルのアクセスの制御」を参照してください。

次のポリシー例では、タグ を持つ ID にのみアクセスを許可しますTeam=Payments。このポリシーにより、アカウント IDsやロール ARNs をハードコーディングせずにアクセスを制御できます。

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

匿名 (認証されていない) プリンシパル

匿名プリンシパルは、署名バージョン 4 (SigV4) で AWS リクエストに署名せず、サービスネットワークに接続されている VPC 内にある発信者です。匿名プリンシパルはサービスネットワークのサービスに対して認証されていないリクエストを認証ポリシーで許可されている場合には実行できます。

認証ポリシーの例

認証されたプリンシパルによるリクエストが必要な認証ポリシーの例には次のものがあります。

すべての例で、us-west-2 リージョンと架空のアカウント ID を使用しています。

例 1: 特定の AWS 組織によるサービスへのアクセスを制限する

次の認証ポリシーの例では、ポリシーが適用されるサービスネットワーク内のサービスにアクセスする権限を、認証されたすべてのリクエストに付与します。ただし、リクエストは、 条件で指定された AWS 組織に属するプリンシパルから発信される必要があります。

JSON

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

例 2: 特定の IAM ロールによるサービスへのアクセスを制限する

次の認証ポリシーの例では、Resource 要素で指定されたサービスに対して HTTP GET リクエストを行う権限を、IAM ロール rates-client を使用するすべての認証されたリクエストに付与します。Resource 要素のリソースはポリシーがアタッチされているサービスと同じです。

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"
            }
         }
      }
   ]
}
```

例 3: 特定の VPC の認証されたプリンシパルによるサービスへのアクセスを制限する

次の認証ポリシーの例では、VPC ID が *vpc-1a2b3c4d* の VPC のプリンシパルからの認証されたリクエストのみを許可します。

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"
            }
         }
```

```
      }
   ]
}
```

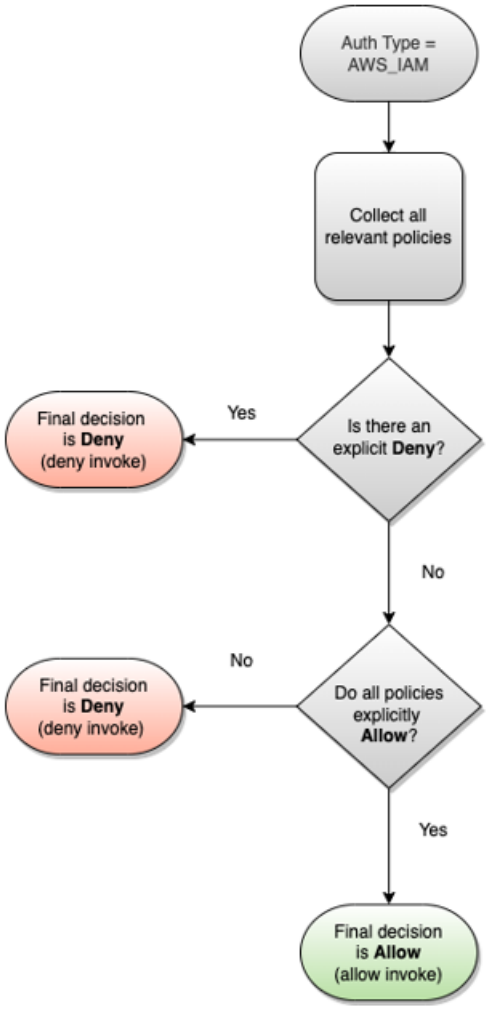
認可の仕組み

VPC Lattice サービスがリクエストを受信すると、 AWS エンフォースメントコードは関連するすべてのアクセス許可ポリシーを一緒に評価して、リクエストを承認または拒否するかどうかを決定します。リクエストコンテキストに適用されるすべての IAM アイデンティティベースのポリシーと認可ポリシーを認可時に評価します。デフォルトでは、認証タイプが AWS_IAM の場合、すべてのリクエストは暗黙的に拒否されます。関連するすべてのポリシーからの明示的な許可はデフォルトに優先します。

認可では次のことが行われます。

- 関連するすべての IAM アイデンティティベースのポリシーと認証ポリシーを収集します。
- 収集したポリシーのセットを評価します。
  - リクエスタ (IAM ユーザーまたはロールなど) が属するアカウントにおいて、オペレーションを実行する権限を持っていることを確認します。明示的な許可ステートメントがない場合、 AWS はリクエストを承認しません。
  - リクエストがサービスネットワークの認証ポリシーによって許可されていることを確認します。認証ポリシーが有効であっても明示的な許可ステートメントがない場合、 AWS はリクエストを許可しません。明示的な許可ステートメントがある場合、または認証タイプが NONE の場合、コードが継続します。
  - リクエストがサービスの認証ポリシーによって許可されていることを確認します。認証ポリシーが有効であっても明示的な許可ステートメントがない場合、 AWS はリクエストを許可しません。明示的な許可ステートメントがある場合、または認証タイプが NONE の場合、エンフォースメントコードにより許可の最終決定が返されます。
  - ポリシー内の明示的な拒否は、すべての許可に優先します。

次の図は認可の流れを示しています。リクエストが行われると、関連するポリシーによって特定のサービスへのリクエストアクセスが許可または拒否されます。



セキュリティグループを使用して VPC Lattice のトラフィックを制御する

AWS セキュリティグループは仮想ファイアウォールとして機能し、関連付けられているエンティティとの間のネットワークトラフィックを制御します。VPC Lattice を使用すると、セキュリティグループを作成し、VPC をサービスネットワークに接続する VPC 関連付けに割り当てて、サービスネットワークに追加のネットワークレベルのセキュリティ保護を適用できます。VPC エンドポイントを使用して VPC をサービスネットワークに接続する場合は、VPC エンドポイントにセキュリティグループを割り当てることもできます。同様に、VPC 内のリソースへのアクセスを有効にするために作成したリソースゲートウェイにセキュリティグループを割り当てることができます。

内容

- マネージドプレフィックスリスト
- 「セキュリティグループのルール」
- VPC の関連付けのセキュリティグループを管理する

マネージドプレフィックスリスト

VPC Lattice には、サービスネットワークの関連付けを使用して VPC をサービスネットワークに接続するときに、VPC Lattice ネットワーク経由でトラフィックをルーティングするために使用される IP アドレスを含むマネージドプレフィックスリストが用意されています。これらの IPsは、プライベートリンクローカル IPsまたはルーティング不可能なパブリック IPs。

セキュリティグループルールで VPC Lattice マネージドプレフィックスリストを参照できます。これにより、トラフィックはクライアントから VPC Lattice サービスネットワークを経由して VPC Lattice サービスターゲットに流れます。

例えば、EC2 インスタンスが米国西部 (オレゴン) リージョン (us-west-2) でターゲットとして登録されているとします。VPC Lattice マネージドプレフィックスリストからのインバウンド HTTPS アクセスを許可するルールをインスタンスのセキュリティグループに追加すると、このリージョンの VPC Lattice トラフィックがインスタンスに到達できるようになります。セキュリティグループから他のすべてのインバウンドルールを削除すると、VPC Lattice 以外のトラフィックがインスタンスに到達することを防げます。

VPC Lattice のマネージドプレフィックスリストの名前は次のとおりです。

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

詳細については、「Amazon VPC ユーザーガイド」の「AWSマネージドプレフィックスリスト」を参照してください。

Windows クライアントと macOS クライアント

VPC Lattice プレフィックスリストのアドレスは、リンクローカルアドレスとルーティング不可能なパブリックアドレスです。これらのクライアントから VPC Lattice に接続する場合は、マネージドプレフィックスリストの IP アドレスをクライアントのプライマリ IP アドレスに転送するように設定を更新する必要があります。以下は、Windows クライアントの設定を更新するコマンドの例です。169.254.171.0は、マネージドプレフィックスリストのアドレスの 1 つです。

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

macOS クライアントの設定を更新するコマンドの例を次に示します。 169.254.171.0はマネージドプレフィックスリストのアドレスの 1 つです。

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

静的ルートの作成を回避するには、VPC 内のサービスネットワークエンドポイントを使用して接続を確立することをお勧めします。詳細については、「the section called “サービスネットワーク VPC エンドポイントの関連付けを管理する”」を参照してください。

「セキュリティグループのルール」

VPC Lattice をセキュリティグループと一緒に使用してもしなくても、既存の VPC セキュリティグループの設定には影響しません。ただし、独自のセキュリティグループをいつでも追加できます。

主な考慮事項

- クライアントのセキュリティグループルールは、VPC Lattice へのアウトバウンドトラフィックを制御します。
- ターゲットのセキュリティグループルールは、ヘルスチェックトラフィックを含め、VPC Lattice からターゲットへのインバウンドトラフィックを制御します。
- サービスネットワークと VPC の関連付けに関するセキュリティグループルールは、VPC Lattice サービスネットワークにアクセスできるクライアントを制御します。
- リソースゲートウェイのセキュリティグループルールは、リソースゲートウェイからリソースへのアウトバウンドトラフィックを制御します。

リソースゲートウェイからデータベースリソースに流れるトラフィックの推奨アウトバウンドルール

トラフィックがリソースゲートウェイからリソースに流れるには、オープンポートのアウトバウンドルールと、リソースの承認されたリスナープロトコルを作成する必要があります。

目的地	プロトコル	ポート範囲	コメント
##### CIDR ##	TCP	3306	リソースゲートウェイからデータベースへのトラフィックを許可する

サービスネットワークと VPC の関連付けの推奨されるインバウンドルール

クライアント VPCs からサービスネットワークに関連付けられたサービスにトラフィックを流れるには、サービスのリスナーポートとリスナープロトコルのインバウンドルールを作成する必要があります。

ソース	プロトコル	ポート範囲	コメント
VPC CIDR	*listener*	*listener*	クライアントから VPC Lattice へのトラフィックを許可する

クライアントインスタンスから VPC Lattice に流れるトラフィックの推奨されるアウトバウンドルール

デフォルトで、セキュリティグループはすべてのアウトバウンドトラフィックを許可します。ただし、カスタムアウトバウンドルールがある場合は、クライアントインスタンスが VPC Lattice サービスネットワークに関連付けられているすべてのサービスに接続できるように、リスナーポートとプロトコルの VPC Lattice プレフィックスへのアウトバウンドトラフィックを許可する必要があります。VPC Lattice のプレフィックスリストの ID を参照することで、このトラフィックを許可できます。

目的地	プロトコル	ポート範囲	コメント
VPC Lattice ## ######### ID	*listener*	*listener*	クライアントから VPC Lattice へのトラフィックを許可する

VPC Lattice からターゲットインスタンスに流れるトラフィックの推奨されるインバウンドルール

トラフィックは VPC Lattice から流れるため、クライアントセキュリティグループをターゲットのセキュリティグループのソースとして使用することはできません。VPC Lattice のプレフィックスリストの ID を参照できます。

ソース	プロトコル	ポート範囲	コメント
VPC Lattice ## ######### ID	*target*	*target*	VPC Lattice からターゲットへのトラフィックを許可する
VPC Lattice ## ######### ID	*health check*	*health check*	VPC Lattice からターゲットへのヘルスチェックトラフィックを許可する

VPC の関連付けのセキュリティグループを管理する

を使用して、VPC のセキュリティグループ AWS CLI を表示、追加、または更新して、ネットワークの関連付けをサービスできます。を使用する場合 AWS CLI、コマンドはプロファイル用に AWS リージョン 設定された で実行されることに注意してください。別のリージョンでコマンドを実行する場合は、プロファイルのデフォルトのリージョンを変更するか、コマンドに `--region` パラメータを使用します。

開始する前に、サービスネットワークに追加する VPC と同じ VPC でセキュリティグループを作成していることを確認します。詳細については、「Amazon VPC ユーザーガイド」の「セキュリティグループを使用してリソースへのトラフィックを制御する」を参照してください。

コンソールを使用して VPC の関連付け作成時にセキュリティグループを追加する方法

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービスネットワーク] を選択します。
3. サービスネットワークの名前を選択して、その詳細ページを開きます。
4. [VPC の関連付け] タブで [VPC の関連付けを作成]、[Add VPC association] の順に選択します。
5. VPC と最大 5 つのセキュリティグループを選択します。
6. [Save changes] (変更の保存) をクリックします。

コンソールを使用して既存の VPC の関連付けのセキュリティグループを追加または更新する方法

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [VPC Lattice] で、[サービスネットワーク] を選択します。

3. サービスネットワークの名前を選択して、その詳細ページを開きます。
4. [VPC の関連付け] タブで関連付けのチェックボックスをオンにして、[アクション] 、[セキュリティグループの編集] の順に選択します。
5. 必要に応じて、セキュリティグループを追加または削除します。
6. [Save changes] (変更の保存) をクリックします。

を使用して VPC 関連付けを作成するときにセキュリティグループを追加するには AWS CLI

create-service-network-vpc-association コマンドを使用して、VPC の関連付けの VPC の ID と追加するセキュリティグループの ID を指定します。

```
aws vpc-lattice create-service-network-vpc-association \
    --service-network-identifier sn-0123456789abcdef0 \
    --vpc-identifier vpc-1a2b3c4d \
    --security-group-ids sg-7c2270198example
```

成功すると、コマンドは以下のような出力を返します。

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

を使用して既存の VPC 関連付けのセキュリティグループを追加または更新するには AWS CLI

update-service-network-vpc-association コマンドを使用して、サービスネットワークの ID とセキュリティグループの ID を指定します。このセキュリティグループは以前に関連付けられたセキュリティグループを上書きします。リストを更新するときに、1 つ以上のセキュリティグループを定義してください。

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

> **Warning**
>
> すべてのセキュリティグループを削除することはできません。まず VPC の関連付けを削除し、次にセキュリティグループなしで VPC の関連付けを再度作成する必要があります。VPC の関連付けを削除する際は慎重に行ってください。これはトラフィックがそのサービスネットワーク内のサービスに到達することを防いでいます。

ネットワーク ACL を使用して VPC Lattice へのトラフィックを制御する

ネットワークアクセスコントロールリスト (ACL) は、サブネットレベルで特定のインバウンドまたはアウトバウンドのトラフィックを許可または拒否します。デフォルトのネットワーク ACL では、すべてのインバウンドトラフィックとアウトバウンドトラフィックを許可します。サブネットのカスタムネットワーク ACLs を作成して、追加のセキュリティレイヤーを提供できます。詳細については、「Amazon VPC ユーザーガイド」の「ネットワーク ACL」を参照してください。

内容

クライアントサブネットACLs

クライアントサブネットACLs は、クライアントと VPC Lattice 間のトラフィックを許可する必要があります。VPC Lattice のマネージドプレフィックスリストから許可する IP アドレス範囲を取得できます。

インバウンドルールの例を次に示します。

ソース	プロトコル	ポート範囲	コメント
vpc_lattice_cidr_block	TCP	1025-65535	VPC Lattice からクライアントへのトラフィックを許可する

以下は、アウトバウンドルールの例です。

目的地	プロトコル	ポート範囲	コメント
vpc_latti ce_cidr_block	*listener*	*listener*	クライアントから VPC Lattice へのトラフィックを許可する

ターゲットサブネットのネットワーク ACLs

ターゲットサブネットACLs は、ターゲットポートとヘルスチェックポートの両方でターゲットと VPC Lattice 間のトラフィックを許可する必要があります。VPC Lattice のマネージドプレフィックスリストから許可する IP アドレス範囲を取得できます。

インバウンドルールの例を次に示します。

ソース	プロトコル	ポート範囲	コメント
vpc_latti ce_cidr_block	*target*	*target*	VPC Lattice からターゲットへのトラフィックを許可する
vpc_latti ce_cidr_block	*health check*	*health check*	VPC Lattice からターゲットへのヘルスチェックトラフィックを許可する

以下は、アウトバウンドルールの例です。

目的地	プロトコル	ポート範囲	コメント
vpc_latti ce_cidr_block	*target*	1024-65535	ターゲットから VPC Lattice へのトラフィックを許可する
vpc_latti ce_cidr_block	*health check*	1024-65535	ターゲットから VPC Lattice へのヘルス

目的地	プロトコル	ポート範囲	コメント
			チェックトラフィックを許可する

Amazon VPC Lattice の SIGv4 認証リクエスト

VPC Lattice は、クライアント認証に Signature Version 4 (SIGv4) または Signature Version 4A (SIGv4A) を使用します。詳細については、「IAM ユーザーガイド」の「API リクエストに対する AWS 署名バージョン 4」を参照してください。

考慮事項

- VPC Lattice は、SIGv4 または SIGv4A で署名されたリクエストの認証を試みます。認証なしではリクエストは失敗します。
- VPC Lattice ではペイロード署名をサポートしていません。`x-amz-content-sha256` ヘッダーの値を `"UNSIGNED-PAYLOAD"` に設定して送信する必要があります。

例

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

Python

この例では、署名付きリクエストを安全な接続経由でネットワークに登録されたサービスに送信します。requests を使用する場合、botocore パッケージにより認証プロセスが効率化されますが、必須ではありません。詳細については、Boto3 ドキュメントの「認証情報」を参照してください。

`botocore` および `awscrt`パッケージをインストールするには、次のコマンドを使用します。詳細については、AWS 「CRT Python」を参照してください。

```
pip install botocore awscrt
```

Lambda でクライアントアプリケーションを実行する場合は、Lambda レイヤーを使用して必要なモジュールをインストールするか、デプロイパッケージに含めます。

次の例では、プレースホルダー値を独自の値に置き換えます。

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

この例は、カスタムインターセプターを使用してリクエスト署名を実行する方法を示しています。AWS SDK for Java 2.x からのデフォルトの認証情報プロバイダークラスを使用して、正しい認証情報を取得します。特定の認証情報プロバイダーを使用する場合は、AWS SDK for Java 2.x から選択できます。では、HTTPS 経由の署名なしペイロードのみ AWS SDK for Java が許可されます。ただし、署名者を拡張することによって、HTTP 経由の未署名のペイロードをサポートできます。

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

この例では、 への追加の依存関係が必要です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

この例では、aws-crt NodeJS バインディングを使用して HTTPS 経由の署名付きリクエストを送信しています。

`aws-crt` パッケージをインストールするには、次のコマンドを使用します。

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

AWS_REGION 環境変数が存在する場合、この例では AWS_REGION で指定されたリージョンを使用しています。デフォルトのリージョンは us-east-1 です。

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

この例では、Go の Smithy コードジェネレーターと Go AWS プログラミング言語の SDK を使用して、リクエスト署名リクエストを処理します。この例では、Go バージョン 1.21 以降が必要です。

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)
}
```

Golang - GRPC

この例では、Go AWS プログラミング言語の SDK を使用して GRPC リクエストのリクエスト署名を処理します。これは、GRPC サンプルコードリポジトリのエコーサーバーで使用できます。

```
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 Lattice でのデータ保護

責任 AWS 共有モデル、Amazon VPC Lattice でのデータ保護に適用されます。このモデルで説明されているように、 AWS はすべての を実行するグローバルインフラストラクチャを保護する責任があります AWS クラウド。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。このコンテンツには、使用される AWS のサービス のセキュリティ設定と管理タスクが含まれます。データプライバシーの詳細については、「データプライバシーのよくある質問」を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿された「AWS 責任共有モデルおよび GDPR」のブログ記事を参照してください。

転送中の暗号化

VPC Lattice はコントロールプレーンとデータプレーンからなるフルマネージドサービスです。各プレーンはサービスにおいて異なる目的を担っています。コントロールプレーンには、リソースの作成、読み取り/説明、更新、削除、一覧表示 (CRUDL) に使用される管理 APIs (`CreateService`や`UpdateService`など) が用意されています。VPC Lattice コントロールプレーンへの通信は、転送中に TLS によって保護されます。データプレーンは、サービス間の相互接続を提供する VPC Lattice Invoke API です。TLS は、HTTPS または TLS を使用するときに VPC Lattice データプレーンへの通信を暗号化します。暗号スイートとプロトコルバージョンでは VPC Lattice が提供するデフォルトが使用され、設定はできません。詳細については、「VPC Lattice サービスの HTTPS リスナー」を参照してください。

保管中の暗号化

デフォルトでは、保管中のデータの暗号化により、機密データの保護に伴う運用上のオーバーヘッドと複雑さの軽減につながります。同時に、安全なアプリケーションを構築して、厳格な暗号化のコンプライアンスと規制要件に対応できます。

内容

- Amazon S3 マネージドキーを用いたサーバー側の暗号化 (SSE-S3)
- AWS KMS (SSE-KMS) に保存されている AWS KMS キーによるサーバー側の暗号化

Amazon S3 マネージドキーを用いたサーバー側の暗号化 (SSE-S3)

Amazon S3 マネージドキーによるサーバー側の暗号化 (SSE-S3) を使用すると、各オブジェクトは一意のキーで暗号化されます。追加の保護として、定期的にローテーションするルートキーを使用してキー自体を暗号化します。Amazon S3 のサーバー側の暗号化では、利用できるものの中でも最強のブロック暗号の 1 つである、256 ビットの高度暗号化規格 (AES-256) GCM を使用してデータを暗号化します。AES-GCM より前に暗号化されたオブジェクトについては、これらのオブジェクトを復号するために AES-CBC が引き続きサポートされています。詳細については、「Amazon S3 マネージド暗号化キーによるサーバー側の暗号化 (SSE-S3)」を参照してください。

VPC Lattice アクセスログの S3-managed暗号化キー (S3-S3) によるサーバー側の暗号化を有効にすると、S3 バケットに保存される前に、各アクセスログファイルが自動的に暗号化されます。詳細については、「Amazon CloudWatch ユーザーガイド」の「Amazon S3 に送信された口グ」を参照してください。

AWS KMS (SSE-KMS) に保存されている AWS KMS キーによるサーバー側の暗号化

AWS KMS キーによるサーバー側の暗号化 (SSE-KMS) は SSE-S3 と似ていますが、このサービスの使用には追加の利点と料金がかかります。 AWS KMS キーには、Amazon S3 内のオブジェクトへの不正アクセスに対する保護を強化する個別のアクセス許可があります。SSE-KMS には、 AWS KMS キーがいつ誰によって使用されたかを示す監査証跡も用意されています。詳細については、「AWS Key Management Service によるサーバー側の暗号化 (SSE-KMS) の使用」を参照してください。

内容

証明書のプライベートキーの暗号化と復号化

ACM 証明書とプライベートキーは、エイリアス aws/acm を持つ AWS マネージド KMS キーを使用して暗号化されます。このエイリアスを持つキー ID は、 AWS KMS コンソールの AWS マネージドキーで表示できます。

VPC Lattice は ACM リソースに直接アクセスしません。 AWS TLS Connection Manager を使用して証明書のプライベートキーを保護し、アクセスします。ACM 証明書を使用して VPC Lattice サービスを作成すると、VPC Lattice は証明書を AWS TLS 接続マネージャに関連付けます。これを行うには、プレフィックス aws/acm を使用して、マネージドキー AWS KMS に対する許可を に作成し

ます AWS 。権限はポリシーツールであり、TLS 接続マネージャに暗号化オペレーションでの KMS キーの使用を許可します。この権限により、被付与者のプリンシパル (TLS 接続マネージャ) は指定された権限オペレーションを KMS キーで呼び出し、証明書のプライベートキーを復号化できます。TLS 接続マネージャは証明書と復号化された (プレーンテキストの) プライベートキーを使用して、VPC Lattice サービスのクライアントとの安全な接続 (SSL/TLS セッション) を確立します。証明書と VPC Lattice サービスとの関連付けが解除されると、この許可は廃止されます。

KMS キーへのアクセスを削除する場合は、 または の `update-service` コマンドを使用して、サービスから証明書を置き換え AWS マネジメントコンソール または削除することをお勧めします AWS CLI。

VPC Lattice の暗号化コンテキスト

暗号化コンテキストは、プライベートキーの使用目的に関するコンテキスト情報を含むキーと値のペアのオプションセットです。 は、暗号化コンテキストを暗号化されたデータに AWS KMS バインドし、認証された暗号化をサポートする追加の認証済みデータとして使用します。

TLS キーを VPC Lattice と TLS 接続マネージャで使用すると、VPC Lattice サービスの名前が保管中のキーの暗号化に使用される暗号化コンテキストに含まれます。証明書とプライベートキーが使用されている VPC Lattice サービスを確認するには、次のセクションに示すように CloudTrail ログで暗号化コンテキストを表示するか、ACM コンソールの関連リソースタブを確認します。

データを復号化するには、そのリクエストに同じ暗号化コンテキストを含めます。VPC Lattice は、すべての AWS KMS 暗号化オペレーションで同じ暗号化コンテキストを使用します。キーは `aws:vpc-lattice:arn`で、値は VPC Lattice サービスの Amazon リソースネーム (ARN) です。

次の例では、`CreateGrant` のようなオペレーションの出力における暗号化コンテキストを示しています。

```
"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 Lattice の暗号化キーをモニタリングする

VPC Lattice サービスで AWS マネージドキーを使用する場合、 AWS CloudTrail を使用して VPC Lattice が送信するリクエストを追跡できます AWS KMS。

CreateGrant

ACM 証明書を VPC Lattice サービスに追加すると、TLS 接続マネージャが ACM 証明書に関連付けられたプライベートキーを復号できるようにする CreateGrant リクエストが自動で送信されます。

CreateGrant オペレーションは、CloudTrail、イベント履歴、CreateGrant でイベントとして表示できます。

以下は、 CreateGrantオペレーションの CloudTrail イベント履歴のイベントレコードの例です。


```
{
    "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"
}
```


上記のCreateGrant例では、被付与者プリンシパルは TLS Connection Manager であり、暗号化コンテキストには VPC Lattice サービス ARN があります。

ListGrants

KMS キー ID とアカウント ID を使用して ListGrants API を呼び出せます。呼び出すと、指定した KMS キーに対するすべての権限のリストが表示されます。詳細については、「ListGrants」を参照してください。

で次のListGrantsコマンドを使用して AWS CLI 、すべての許可の詳細を表示します。

```
aws kms list-grants –key-id your-kms-key-id
```

以下は出力の例です。

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

上記のListGrants例では、被付与者プリンシパルは TLS Connection Manager であり、暗号化コンテキストには VPC Lattice サービス ARN があります。

Decrypt

VPC Lattice は TLS 接続マネージャを使用してプライベートキーを復号化する Decrypt オペレーションを呼び出し、VPC Lattice サービスで TLS 接続を提供します。Decrypt オペレーションは、CloudTrail イベント履歴 Decrypt でイベントとして表示できます。

以下は、 Decryptオペレーションの CloudTrail イベント履歴のイベントレコードの例です。


```
{
    "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 Lattice のアイデンティティとアクセス管理

以下のセクションでは、VPC Lattice API アクションを実行できるユーザーを制御することで、 AWS Identity and Access Management (IAM) を使用して VPC Lattice リソースを保護する方法について説明します。

トピック

- Amazon VPC Lattice で IAM が機能する仕組み
- Amazon VPC Lattice API アクセス許可
- Amazon VPC Lattice のアイデンティティベースのポリシー
- Amazon VPC Lattice のサービスにリンクされたロールの使用
- AWS Amazon VPC Lattice の マネージドポリシー

Amazon VPC Lattice で IAM が機能する仕組み

IAM を使用して VPC Lattice へのアクセスを管理する前に、VPC Lattice で利用できる IAM の機能について説明します。

IAM 機能	VPC Lattice のサポート
アイデンティティベースのポリシー	あり
リソースベースのポリシー	はい
ポリシーアクション	あり
ポリシーリソース	あり
ポリシー条件キー	あり
ACL	なし
ABAC (ポリシー内のタグ)	あり

IAM 機能	VPC Lattice のサポート
一時的な認証情報	はい
サービスロール	いいえ
サービスリンクロール	はい

VPC Lattice およびその他の AWS のサービスがほとんどの IAM 機能と連携する方法の概要については、IAM ユーザーガイドのAWS 「IAM と連携する のサービス」を参照してください。

VPC Lattice のアイデンティティベースのポリシー

アイデンティティベースのポリシーのサポート: あり

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。JSON ポリシーで使用できるすべての要素について学ぶには、「IAM ユーザーガイド」の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

VPC Lattice 内のリソースベースのポリシー

リソースベースのポリシーのサポート: あり

リソースベースのポリシーは、 のリソースにアタッチする JSON ポリシードキュメントです AWS。リソースベースのポリシーをサポートする AWS サービスでは、サービス管理者はそれらを使用して、その AWS サービスの特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーで、プリンシパルを指定する必要があります。

VPC Lattice は認証ポリシーをサポートしています。これはサービスネットワーク内のサービスへのアクセスを制御できるリソースベースのポリシーです。詳細については、「認証ポリシーを使用して VPC Lattice サービスへのアクセスを制御する」を参照してください。

また、VPC Lattice は AWS Resource Access Managerとの統合で使用するリソースベースのアクセス許可ポリシーもサポートしています。これらのリソースベースのポリシーを使用して、サービス、リソース設定、サービスネットワークの他の AWS アカウントまたは組織への接続を管理するアクセス許可を付与できます。詳細については、「VPC Lattice のエンティティを共有する」を参照してください。

VPC Lattice のポリシーアクション

ポリシーアクションのサポート: あり

IAM ポリシーステートメントで、IAM をサポートするすべてのサービスからの任意の API アクションを指定できます。VPC Lattice の場合、API アクションの名前に `vpc-lattice:` のプレフィックスを使用します。例えば、`vpc-lattice:CreateService`、`vpc-lattice:CreateTargetGroup`、および `vpc-lattice:PutAuthPolicy` のようになります。

単一のステートメントで複数のアクションを指定するには、次のようにカンマで区切ります。

```
"Action": [ "vpc-lattice:action1", "vpc-lattice:action2" ]
```

ワイルドカードを使用して複数のアクションを指定することもできます。例えば、`Get` という単語で始まるアクション名すべてを次のように指定できます。

```
"Action": "vpc-lattice:Get*"
```

VPC Lattice API アクションの全リストを確認するには、「サービス認可リファレンス」の「Amazon VPC Lattice で定義されたアクション」を参照してください。

VPC Lattice のポリシーリソース

ポリシーリソースのサポート: あり

IAM ポリシーステートメントで、`Resource` 要素は、ステートメントがカバーするオブジェクトを指定します。VPC Lattice では、各 IAM ポリシーステートメントは ARN を使用して指定したリソースに適用されます。

それぞれの Amazon リソースネーム (ARN) 形式はリソースによって異なります。ARN を指定するには、*######*のテキストを、リソース固有の情報に置き換えます。

- アクセスログサブスクリプション:

```
"Resource": "arn:aws:vpc-lattice:region:account-id:accesslogsubscription/access-log-subscription-id"
```

- リスナー:

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

- リソースゲートウェイ

```
"Resource": "arn:aws:vpc-lattice:region:account-id:resourcegateway/resource-gateway-id"
```

- リソース設定

```
"Resource": "arn:aws:vpc-lattice:region:account-id:resourceconfiguration/resource-configuration-id"
```

- ルール:

```
"Resource": "arn:aws:vpc-lattice:region:account-id:service/service-id/
listener/listener-id/rule/rule-id"
```

- サービス:

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

- サービスネットワーク:

```
"Resource": "arn:aws:vpc-lattice:region:account-id:servicenetwork/service-network-id"
```

- サービスネットワークのサービスの関連付け:

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkserviceassociation/service-network-service-association-id"
```

- サービスネットワークリソース設定の関連付け

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkresourceassociation/service-network-resource-association-id"
```

- サービスネットワークの VPC の関連付け:

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkvpcassociation/service-network-vpc-association-id"
```

- ターゲットグループ:

```
"Resource": "arn:aws:vpc-lattice:region:account-id:targetgroup/target-group-id"
```

VPC Lattice のポリシー条件キー

サービス固有のポリシー条件キーのサポート: あり

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Condition` 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。イコールや未満などの条件演算子を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「グローバル条件コンテキストキー」を参照してください。

VPC Lattice 条件キーのリストを確認するには、「サービス認可リファレンス」の「Amazon VPC Lattice の条件キー」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートしています。 AWS グローバル条件キーの詳細については、「IAM ユーザーガイド」のAWS 「グローバル条件コンテキストキー」を参照してください。

VPC Lattice のアクセスコントロールリスト (ACL)

ACL のサポート: なし

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするためのアクセス許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

VPC Lattice での属性ベースのアクセス制御 (ABAC)

ABAC (ポリシー内のタグ) のサポート: あり

属性ベースのアクセス制御 (ABAC) は、タグと呼ばれる属性に基づいてアクセス許可を定義する認可戦略です。IAM エンティティと AWS リソースにタグをアタッチし、プリンシパルのタグがリソースのタグと一致するときにオペレーションを許可するように ABAC ポリシーを設計できます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値はありです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、「IAM ユーザーガイド」の「ABAC 認可でアクセス許可を定義する」を参照してください。ABAC をセットアップする手順を説明するチュートリアルについては、「IAM ユーザーガイド」の「属性ベースのアクセスコントロール (ABAC) を使用する」を参照してください。

VPC Lattice で一時的な認証情報を使用する

一時的な認証情報のサポート: あり

一時的な認証情報は AWS 、リソースへの短期的なアクセスを提供し、フェデレーションまたはスイッチロールの使用時に自動的に作成されます。長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成 AWS することをお勧めします。詳細については、「IAM ユーザーガイド」の「IAM の一時的な認証情報」および「AWS のサービス と IAM との連携」を参照してください。

VPC Lattice のサービスロール

サービスロールのサポート: なし

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの AWS のサービスに許可を委任するロールを作成するを参照してください。

> **Warning**
>
> サービスロールの許可を変更すると、VPC Lattice の機能が破損する可能性があります。VPC Lattice が指示する場合以外は、サービスロールを編集しないでください。

VPC Lattice のサービスにリンクされたロール

サービスリンクロールのサポート: あり

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスリンクロールのアクセス許可を表示できますが、編集することはできません。

VPC Lattice のサービスにリンクされたロールの作成または管理の詳細については、「Amazon VPC Lattice のサービスにリンクされたロールの使用」を参照してください。

Amazon VPC Lattice API アクセス許可

必要な VPC Lattice API アクションを呼び出すアクセス許可を IAM アイデンティティ (ユーザーやロールなど) に付与する必要があります。詳細については、「VPC Lattice のポリシーアクション」を参照してください。さらに、VPC Lattice アクションによっては、他の AWS APIs から特定のアクションを呼び出すアクセス許可を IAM ID に付与する必要があります。

API に必要なアクセス許可

API から次のアクションを呼び出す場合は、指定されたアクションを呼び出すアクセス許可を IAM ユーザーに付与する必要があります。

`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 (セキュリティグループが指定されている場合にのみ必要)

UpdateServiceNetworkVpcAssociation

- vpc-lattice:UpdateServiceNetworkVpcAssociation

- ec2:DescribeSecurityGroups (セキュリティグループが指定されている場合にのみ必要)

CreateTargetGroup

- vpc-lattice:CreateTargetGroup
- ec2:DescribeVpcs

RegisterTargets

- vpc-lattice:RegisterTargets
- ec2:DescribeInstances (ターゲットグループタイプが INSTANCE の場合のみ必要)
- ec2:DescribeVpcs (ターゲットグループタイプが INSTANCE または IP の場合のみ必要)
- ec2:DescribeSubnets (ターゲットグループタイプが INSTANCE または IP の場合のみ必要)
- lambda:GetFunction (ターゲットグループタイプが LAMBDA の場合のみ必要)
- lambda:AddPermission (ターゲットグループが指定された Lambda 関数を呼び出す権限をまだ持っていない場合にのみ必要)

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 Lattice のアイデンティティベースのポリシー

デフォルトでは、ユーザーおよびロールには VPC Lattice リソースを作成または変更するアクセス許可はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。

これらのサンプルの JSON ポリシードキュメントを使用して IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーを作成する (コンソール)」を参照してください。

VPC Lattice が定義するアクションとリソースタイプ (リソースタイプごとの ARN の形式を含む) の詳細については、「サービス認可リファレンス」の「Amazon VPC Lattice のアクション、リソース、および条件キー」を参照してください。

内容

ポリシーに関するベストプラクティス

アイデンティティベースのポリシーは、ユーザーのアカウントで誰かが VPC Lattice リソースの作成、アクセス、削除ができるどうかを決定します。これらのアクションでは、 AWS アカウントに費用が発生する場合があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行 – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与するAWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM ユーザーガイド の AWS マネージドポリシー または ジョブ機能のAWS マネージドポリシー を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイド の IAM でのポリシーとアクセス許可 を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションが などの特定の を通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます CloudFormation。詳細については、IAM ユーザーガイド の IAM JSON ポリシー要素:条件 を参照してください。

- IAM アクセスアナライザー を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM アクセスアナライザー は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイド の IAM Access Analyzer でポリシーを検証する を参照してください。
- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、MFA をオンにしてセキュリティを強化します。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイド の MFA を使用した安全な API アクセス を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイド の IAM でのセキュリティのベストプラクティス を参照してください。

フルアクセスに必要な追加のアクセス許可

VPC Lattice が統合されている他の AWS サービスおよび VPC Lattice 機能のスイート全体を使用するには、特定の追加のアクセス許可が必要です。このような権限は VPCLatticeFullAccess マネージドポリシーには含まれていません。それは混乱した代理権限昇格リスクがあるためです。

次のポリシーをロールにアタッチし、VPCLatticeFullAccess マネージドポリシーと合わせて使用する必要があります。

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*"
        }
    ]
}
```

このポリシーにより、次の追加のアクセス許可が付与されます。

- `iam:AttachRolePolicy`: 指定した IAM ロールに指定のマネージドポリシーをアタッチできます。
- `iam:PutRolePolicy`: 指定した IAM ロールに埋め込まれたインラインポリシードキュメントを追加または更新できます。
- `s3:PutBucketPolicy`: Amazon S3 バケットにバケットポリシーを適用できます。
- `firehose:TagDeliveryStream`: Firehose 配信ストリームのタグを追加または更新できます。

VPC Lattice のアイデンティティベースのポリシーの例

トピック

ポリシーの例: サービスネットワークへの VPC の関連付けを管理する

次の例は、このポリシーを持つユーザーにサービスネットワークとの VPC の関連付けを作成、更新、削除する権限を付与するポリシーを示しています。ただし、条件で指定された VPC とサービスネットワークのみに限ります。条件の指定に関する詳細については、VPC Lattice のポリシー条件キー を参照してください。

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"
            }
         }
      }
   ]
```

```
}
```

ポリシーの例: サービスネットワークへのサービス関連付けを作成する

VPC Lattice リソースへのアクセスを制御するために条件キーを使用していない場合は、代わりに Resource 要素内のリソースの ARN を指定してアクセスを制御できます。

次の例は、CreateServiceNetworkServiceAssociation API アクションで使用できるサービスとサービスネットワークの ARN を指定して、このポリシーを持つユーザーが作成できるサービスネットワークにのみサービスの関連付けを制限するポリシーを示しています。ARN 値の指定については、「VPC Lattice のポリシーリソース」を参照してください。

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"
         ]
      }
   ]
}
```

ポリシーの例: リソースにタグを追加する

次の例は、このポリシーを持つユーザーに対して、VPC Lattice リソースにタグを作成するアクセス許可を付与するポリシーを示しています。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "vpc-lattice:TagResource"
         ],
         "Resource": "arn:aws:vpc-lattice:us-west-2:123456789012:*/*"
      }
   ]
}
```

ポリシーの例: サービスにリンクされたロールを作成する

VPC Lattice では、 のユーザーが VPC Lattice リソースを初めて作成するときに、サービスにリンクされたロール AWS アカウント を作成するためのアクセス許可が必要です。サービスにリンクされたロールがまだ存在しない場合は、VPC Lattice によってアカウント内に作成されます。サービスにリンクされたロールは、VPC Lattice がユーザーに代わって他の を呼び出すことができるように、VPC Lattice AWS のサービス にアクセス許可を付与します。詳細については、「the section called “サービスにリンクされたロールの使用”」を参照してください。

この自動ロール作成を成功させるには、ユーザーには `iam:CreateServiceLinkedRole` アクションへのアクセス許可が必要です。

```
"Action": "iam:CreateServiceLinkedRole"
```

次の例は、このポリシーを持つユーザーに対して、VPC Lattice のサービスにリンクされたロールを作成するアクセス許可を付与するポリシーを示しています。

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"
               }
            }
         }
      ]
}
```

詳細についてはIAM ユーザーガイド の「サービスにリンクされた役割のアクセス許可」を参照してください。

Amazon VPC Lattice のサービスにリンクされたロールの使用

Amazon VPC Lattice は、 AWS のサービス ユーザーに代わって他の を呼び出すために必要なアクセス許可に、サービスにリンクされたロールを使用します。詳細については、「IAM ユーザーガイド」の「Service-linked roles」を参照してください。

VPC Lattice は、 という名前のサービスにリンクされたロールを使用します AWSServiceRoleForVpcLattice。

VPC Lattice のサービスにリンクされたロールによるアクセス許可

サービスにリンクされたロール AWSServiceRoleForVpcLattice は、次のサービスを信頼してロールを引き受けます。

- `vpc-lattice.amazonaws.com`

AWSVpcLatticeServiceRolePolicy という名前のロールアクセス許可ポリシーにより、VPC Lattice は `AWS/VpcLattice` 名前空間に CloudWatch メトリクスを公開できます。詳細については、AWS「 マネージドポリシーリファレンスAWSVpcLatticeServiceRolePolicy」の「」を参照してください。

サービスリンクロールの作成、編集、削除を IAM エンティティ (ユーザー、グループ、ロールなど) に許可するにはアクセス許可を設定する必要があります。詳細については、「the section called “ポリシーの例: サービスにリンクされたロールを作成する”」を参照してください。

VPC Lattice のサービスにリンクされたロールを作成する

サービスリンクロールを手動で作成する必要はありません。 AWS マネジメントコンソール、、AWS CLIまたは AWS API で VPC Lattice リソースを作成すると、VPC Lattice によってサービスにリンクされたロールが作成されます。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は同じ方法でアカウントにロールを再作成できます。VPC Lattice リソースを作成すると、VPC Lattice によって再度サービスにリンクされたロールが作成されます。

VPC Lattice のサービスにリンクされたロールを編集する

IAM を使用して、AWSServiceRoleForVpcLattice の説明を編集できます。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの編集」を参照してください。

VPC Lattice のサービスにリンクされたロールを削除する

Amazon VPC Lattice を使用する必要がなくなった場合は、AWSServiceRoleForVpcLattice を削除することをお勧めします。

このサービスにリンクされたロールを削除するには、 AWS アカウント内のすべての VPC Lattice リソースを削除する必要があります。

IAM コンソール、 AWS CLI、または AWS API を使用して、AWSServiceRoleForVpcLattice サービスにリンクされたロールを削除します。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの削除」を参照してください

サービスにリンクされたロールを削除した後、 AWS アカウントに VPC Lattice リソースを作成すると、VPC Lattice がそのロールを再度作成します。

VPC Lattice のサービスにリンクされたロールをサポートするリージョン

VPC Lattice では、このサービスが利用可能なすべてのリージョンで、サービスにリンクされたロールの使用がサポートされています。

AWS Amazon VPC Lattice の マネージドポリシー

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できるように、多くの一般的なユースケースにアクセス許可を付与するように設計されています。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケースに固有のカスタマー管理ポリシーを定義して、アクセス許可を絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS マネージドポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) に影響します。 AWS は、新しい が起動されるか、新しい API オペレーション AWS のサービス が既存のサービスで使用できるようになったときに、 AWS マネージドポリシーを更新する可能性が高くなります。

詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」を参照してください。

AWS マネージドポリシー: VPCLatticeFullAccess

このポリシーは Amazon VPC Lattice へのフルアクセスを提供し、他の依存サービスへのアクセスは制限します。これには次のことを実行する許可が含まれています。

- ACM — カスタムドメイン名の SSL/TLS 証明書 ARN を取得します。
- CloudWatch — アクセスログとモニタリングデータを表示します。
- CloudWatch Logs — アクセスログを設定し、CloudWatch Logs に送信します。
- Amazon EC2 – ネットワークインターフェイスを設定し、EC2 インスタンスと VPCsに関する情報を取得します。これは、リソース設定、リソースゲートウェイ、ターゲットグループの作成、VPC Lattice エンティティの関連付けの設定、ターゲットの登録に使用されます。
- Elastic Load Balancing — Application Load Balancer に関する情報を取得して、ターゲットとして登録します。
- Firehose – アクセスログの保存に使用される配信ストリームに関する情報を取得します。
- Lambda — Lambda 関数に関する情報を取得して、ターゲットとして登録します。
- Amazon RDS – RDS クラスターとインスタンスに関する情報を取得します。
- Amazon S3 — アクセスログを保存するために使用される S3 バケットに関する情報を取得します。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の「VPCLatticeFullAccess」を参照してください。

VPC Lattice が統合されている他の AWS サービスおよび VPC Lattice 機能のスイート全体を使用するには、特定の追加のアクセス許可が必要です。このような権限は `VPCLatticeFullAccess` マ

ネージドポリシーには含まれていません。それは混乱した代理権限昇格リスクがあるためです。詳細については、「フルアクセスに必要な追加のアクセス許可」を参照してください。

AWS マネージドポリシー: VPCLatticeReadOnlyAccess

このポリシーは Amazon VPC Lattice への読み取り専用アクセスを提供し、他の依存サービスへのアクセスは制限します。これには次のことを実行する許可が含まれています。

- ACM — カスタムドメイン名の SSL/TLS 証明書 ARN を取得します。
- CloudWatch — アクセスログとモニタリングデータを表示します。
- CloudWatch Logs — アクセスログサブスクリプションのログ配信情報を表示します。
- Amazon EC2 — EC2 インスタンスと VPC に関する情報を取得して、ターゲットグループを作成しターゲットを登録します。
- Elastic Load Balancing — Application Load Balancer に関する情報を取得します。
- Firehose – アクセスログ配信の配信ストリームに関する情報を取得します。
- Lambda — Lambda 関数に関する情報を表示します。
- Amazon RDS – RDS クラスターとインスタンスに関する情報を取得します。
- Amazon S3 — アクセスログ配信のための S3 バケットに関する情報を取得します。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の「VPCLatticeReadOnlyAccess」を参照してください。

AWS マネージドポリシー: VPCLatticeServicesInvokeAccess

このポリシーは Amazon VPC Lattice サービスを呼び出すためのアクセス許可を付与します。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の「VPCLatticeServicesInvokeAccess」を参照してください。

AWS マネージドポリシー: AWSVpcLatticeServiceRolePolicy

このポリシーは AWSServiceRoleForVpcLattice という名前のサービスにリンクされたロールにアタッチされ、VPC Lattice がユーザーに代わってアクションを実行できるようにします。このポリシーを IAM エンティティにアタッチすることはできません。詳細については、「Amazon VPC Lattice のサービスにリンクされたロールの使用」を参照してください。

このポリシーに対する許可を確認するには、「AWS マネージドポリシーリファレンス」の「AWSVpcLatticeServiceRolePolicy」を参照してください。

AWS マネージドポリシーへの VPC Lattice の更新

このサービスがこれらの変更の追跡を開始してからの VPC Lattice の AWS マネージドポリシーの更新に関する詳細を表示します。このページの変更に関する自動通知を受け取るには、VPC Lattice ユーザーガイドの RSS フィードにサブスクライブしてください。

変更	説明	日付
VPCLatticeFullAccess	VPC Lattice は、Amazon RDS クラスターとインスタンスを記述するための読み取り専用アクセス許可を追加します。	2024 年 12 月 1 日
VPCLatticeReadOnlyAccess	VPC Lattice は、Amazon RDS クラスターとインスタンスを記述するための読み取り専用アクセス許可を追加します。	2024 年 12 月 1 日
AWSVpcLatticeServiceRolePolicy	VPC Lattice は、VPC Lattice がリクエスタマネージドネットワークインターフェイスを作成できるようにするアクセス許可を追加します。	2024 年 12 月 1 日
VPCLatticeFullAccess	VPC Lattice に Amazon VPC Lattice へのフルアクセスと他の依存サービスへの制限付きアクセスを付与する新しいポリシーが追加されました。	2023 年 3 月 31 日
VPCLatticeReadOnlyAccess	VPC Lattice に Amazon VPC Lattice への読み取り専用アクセス権限と他の依存サービスへの制限付きアクセス権限を付与する新しいポリシーが追加されました。	2023 年 3 月 31 日
VPCLatticeServicesInvokeAccess	VPC Lattice に Amazon VPC Lattice サービスを呼び出すためのアクセス許可を付与する新しいポリシーが追加されました。	2023 年 3 月 31 日

変更	説明	日付
AWSVpcLatticeServiceRolePolicy	VPC Lattice にサービスにリンクされたロールへのアクセス許可が追加され、VPC Lattice が AWS/VpcLattice 名前空間で CloudWatch メトリクスを公開できるようになりました。AWSVpcLatticeServiceRolePolicy ポリシーに CloudWatch PutMetricData API アクションを呼び出すためのアクセス許可が含まれるようになりました。詳細については、「Amazon VPC Lattice のサービスにリンクされたロールの使用」を参照してください。	2022 年 12 月 5 日
VPC Lattice による変更の追跡に開始	VPC Lattice は AWS 、管理ポリシーの変更の追跡を開始しました。	2022 年 12 月 5 日

Amazon VPC Lattice のコンプライアンス検証

サードパーティーの監査者は、複数のコンプライアンスプログラムの一環として Amazon VPC Lattice のセキュリティと AWS コンプライアンスを評価します。

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、AWS のサービス「コンプライアンスプログラムによる対象範囲内」の」を参照し、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、AWS 「コンプライアンスプログラム」を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「Downloading Reports in AWS Artifact」を参照してください。

を使用する際のお客様のコンプライアンス責任 AWS のサービス は、お客様のデータの機密性、貴社のコンプライアンス目的、適用可能な法律および規制によって決まります。を使用する際のコンプライアンス責任の詳細については AWS のサービス、AWS 「 セキュリティドキュメント」を参照してください。

インターフェイスエンドポイント (AWS PrivateLink) を使用して Amazon VPC Lattice にアクセスする

VPC と Amazon VPC Lattice とのプライベート接続を確立するには、インターフェイス VPC エンドポイントを作成します。インターフェイスエンドポイントは、インターネットゲートウェイAWS PrivateLink、NAT デバイス、VPN 接続、または Direct Connect 接続なしで VPC Lattice APIs にプライベートにアクセスできるテクノロジーである を利用しています。VPC のインスタンスはパブリック IP アドレスがなくても VPC Lattice API と通信できます。

各インターフェイスエンドポイントはサブネット内の 1 つ以上のネットワークインターフェイスによって表されます。

インターフェイス VPC エンドポイントに関する考慮事項

VPC Lattice のインターフェイス VPC エンドポイントを設定する前に、「 AWS PrivateLink ガイド」の「Access AWS のサービス through AWS PrivateLink」を確認してください。

VPC Lattice では VPC からのすべての API アクションの呼び出しをサポートしています。

VPC Lattice 用のインターフェイス VPC エンドポイントを作成する

VPC Lattice サービスの VPC エンドポイントは、Amazon VPC コンソールまたは AWS Command Line Interface () を使用して作成できますAWS CLI。詳細については、「 AWS PrivateLink ガイド」の「インターフェイス VPC エンドポイントの作成」を参照してください。

次のサービス名を使用して、VPC Lattice の VPC エンドポイントを作成します。

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

エンドポイントのプライベート DNS を有効にすると、`vpc-lattice.us-east-1.amazonaws.com` などのリージョンのデフォルト DNS 名を使用して、VPC Lattice への API リクエストを実行できます。

Amazon VPC Lattice の耐障害性

AWS グローバルインフラストラクチャは、 AWS リージョン およびアベイラビリティーゾーンを中心に構築されています。

AWS リージョン は、複数の物理的に分離および分離されたアベイラビリティーゾーンを提供します。これらは、低レイテンシー、高スループット、および高度に冗長なネットワークで接続されます。

アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェールオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、フォールトトレランス、および拡張性が優れています。

AWS リージョン およびアベイラビリティーゾーンの詳細については、AWS 「 グローバルインフラストラクチャ」を参照してください。

Amazon VPC Lattice のインフラストラクチャセキュリティ

マネージドサービスである Amazon VPC Lattice は、 AWS グローバルネットワークセキュリティで保護されています。 AWS セキュリティサービスと がインフラストラクチャ AWS を保護する方法については、AWS 「 クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して環境を AWS 設計するには、「Security Pillar AWS Well-Architected Framework」の「Infrastructure Protection」を参照してください。

AWS 公開された API コールを使用して、ネットワーク経由で VPC Lattice にアクセスします。クライアントは以下をサポートする必要があります。

- Transport Layer Security (TLS)。TLS 1.2 が必須で、TLS 1.3 をお勧めします。
- DHE (楕円ディフィー・ヘルマン鍵共有) や ECDHE (楕円曲線ディフィー・ヘルマン鍵共有) などの完全前方秘匿性 (PFS) による暗号スイート。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

Amazon VPC Lattice をモニタリングする

このセクションの機能を使用して、Amazon VPC Lattice サービスネットワーク、サービス、ターゲットグループ、VPC 接続をモニタリングします。

内容

Amazon VPC Lattice の CloudWatch メトリクス

Amazon VPC Lattice は、ターゲットグループとサービスに関連するデータを Amazon CloudWatch に送信し、読み取り可能でほぼリアルタイムのメトリクスに加工します。これらのメトリクスは 15 か月間保持されるため、履歴情報にアクセスし、ウェブアプリケーションまたはサービスの動作をより的確に把握できます。また、特定のしきい値を監視するアラームを設定し、これらのしきい値に達したときに通知を送信したりアクションを実行したりできます。詳細については、「Amazon CloudWatch ユーザーガイド」を参照してください。

Amazon VPC Lattice は、 AWS アカウントのサービスにリンクされたロールを使用して Amazon CloudWatch にメトリクスを送信します。詳細については、「Amazon VPC Lattice のサービスにリンクされたロールの使用」を参照してください。

内容

Amazon CloudWatch メトリクスを表示する

Amazon CloudWatch コンソールまたは AWS CLIを使用して、ターゲットグループとサービスの Amazon CloudWatch メトリクスを表示できます。

CloudWatch コンソールを使用してメトリクスを表示するには

1. Amazon CloudWatch コンソール (https://console.aws.amazon.com/cloudwatch/) を開きます。

2. ナビゲーションペインで [Metrics (メトリクス)] を選択してください。
3. AWS/VpcLattice 名前空間を選択します。
4. (オプション) すべてのディメンションでメトリクスを表示するには、検索フィールドに名称を入力します。
5. (オプション) ディメンション別に検索するには、次のいずれかを選択します。
   - ターゲットグループについて報告されたメトリクスのみを表示するには、[ターゲットグループ] を選択します。1 つのターゲットグループのメトリクスを表示するには、検索フィールドにその名前を入力します。
   - サービスについて報告されたメトリクスのみを表示するには、[サービス] を選択します。1 つのサービスのメトリクスを表示するには、検索フィールドにその名前を入力します。

を使用してメトリクスを表示するには AWS CLI

次の CloudWatch list-metrics AWS CLI コマンドを使用して、使用可能なメトリクスを一覧表示します。

```
aws cloudwatch list-metrics --namespace AWS/VpcLattice
```

各メトリクスとそのディメンションの詳細については、「ターゲットグループのメトリクス」と「サービスメトリクス」を参照してください。

ターゲットグループのメトリクス

VPC Lattice は、ターゲットグループに関連するメトリクスを AWS/VpcLattice Amazon CloudWatch 名前空間に自動的に保存します。ターゲットグループの詳細については、「VPC Lattice のターゲットグループ」を参照してください。

ディメンション

ターゲットグループのメトリクスをフィルタリングするには、次のディメンションを使用します。

- AvailabilityZone
- TargetGroup

メトリクス	説明	TargetGroup プロトコル
TotalConnectionCount	合計接続数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS, TCP
ActiveConnectionCount	アクティブな接続数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。	HTTP, HTTPS, TCP

メトリクス	説明	TargetGroup プロトコル
	統計 • 最も有用な統計は Sum です。	
ConnectionErrorCount	接続障害の合計。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS, TCP

メトリクス	説明	TargetGroup プロトコル
HTTP1_ConnectionCount	HTTP/1.1 接続の合計数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS

メトリクス	説明	TargetGroup プロトコル
HTTP2_ConnectionCount	HTTP/2 接続の合計数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS

メトリクス	説明	TargetGroup プロトコル
ConnectionTimeoutCount	接続タイムアウトの合計数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS, TCP

メトリクス	説明	TargetGroup プロトコル
TotalReceivedConnectionBytes	受信した接続バイトの合計数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS, TCP

メトリクス	説明	TargetGroup プロトコル
TotalSentConnectionBytes	送信された接続バイトの合計数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS, TCP

メトリクス	説明	TargetGroup プロトコル
TotalRequestCount	リクエストの合計数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS

メトリクス	説明	TargetGroup プロトコル
ActiveRequestCount	アクティブなリクエストの合計数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS

メトリクス	説明	TargetGroup プロトコル
RequestTime	ミリ秒単位の最後のバイトへのリクエスト時間。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Average および pNN.NN (パーセンタイル) です。	HTTP, HTTPS

メトリクス	説明	TargetGroup プロトコル
HTTPCode_2XX_Count, HTTPCode_3XX_Count, HTTPCode_4XX_Count, HTTPCode_5XX_Count	HTTP レスポンスコードを集約します。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS

メトリクス	説明	TargetGroup プロトコル
TLSConnectionErrorCount	証明書検証の失敗を除く TLS 接続エラーの合計数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS, TCP

メトリクス	説明	TargetGroup プロトコル
TotalTLSConnectionHandshakeCount	成功した TLS 接続ハンドシェイクの合計数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。	HTTP, HTTPS, TCP

サービスメトリクス

VPC Lattice は、サービスに関連するメトリクスを AWS/VpcLattice Amazon CloudWatch 名前空間に自動的に保存します。サービスの詳細については、「VPC Lattice のサービス」を参照してください。

ディメンション

ターゲットグループのメトリクスをフィルタリングするには、次のディメンションを使用します。

- AvailabilityZone
- Service

メトリクス	説明
RequestTimeoutCount	レスポンスを待っている間にタイムアウトになったリクエストの合計数。 レポート条件 • リソースがトラフィックを受信してから常に報告されます (値がゼロかゼロ以外かにかかわらず) 。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。
TotalRequestCount	リクエストの合計数。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。
RequestTime	ミリ秒単位のリクエスト時間。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。

メトリクス	説明
	レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Average および pNN.NN (パーセンタイル) です。
HTTPCode_2XX_Count , HTTPCode_3XX_Count , HTTPCode_4XX_Count , HTTPCode_5XX_Count	HTTP レスポンスコードを集約します。 レポート条件 • リソースがトラフィックを受信した時点から常に (値がゼロであるかゼロ以外であるかに関係なく) 報告されます。 レポートの頻度 • 1 分に 1 回。 統計 • 最も有用な統計は Sum です。

Amazon VPC Lattice のアクセスログ

アクセスログは、VPC Lattice サービスとリソース設定に関する詳細情報をキャプチャします。これらのアクセスログを使用して、トラフィックパターンを分析し、ネットワーク内のすべてのサービスを監査できます。VPC Lattice サービスでは、 を公開VpcLatticeAccessLogsし、リソース設定では、個別に設定する必要がある VpcLatticeResourceAccessLogs を公開します。

アクセスログはオプションであり、デフォルトでは無効になっています。アクセスログを有効にした後は、いつでも無効にできます。

料金

アクセスログが公開されると料金が発生します。ユーザーに代わって AWS ネイティブに発行するログは、販売ログと呼ばれます。Vended Logs の料金の詳細については、「Amazon CloudWatch 料金表」を参照してください。[ログ]を選択すると、[Vended Logs] の下に価格が表示されます。

内容

アクセスログを有効にするために必要な IAM アクセス許可

アクセスログを有効にしてログを送信先に送信するには、使用している IAM ユーザー、グループ、またはロールにアタッチされたポリシーで次のアクションが必要です。

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": [
                "*"
            ]
        }
    ]
}
```

詳細については、[AWS Identity and Access Management ユーザーガイド]の「IAM ID アクセス許可の追加と削除」を参照してください。

IAM ユーザー、IAM グループ、または使用している IAM ロールにアタッチされているポリシーを更新したら、アクセスログの有効化 に進みます。

アクセスログの送信先

アクセスログは、次の宛先に送信できます。

Amazon CloudWatch Logs

- VPC Lattice は通常 2 分以内に CloudWatch Logs にログを配信します。ただし、実際のログ配信時間はベストエフォートベースであり、さらにレイテンシーが発生する可能性があることに注意してください。
- ロググループに特定の権限がない場合は、リソースポリシーが自動的に作成され、CloudWatch ロググループに追加されます。詳細については、「Amazon CloudWatch ユーザーガイド」の「CloudWatch Logs に送信されたログ」を参照してください。
- CloudWatch に送信されたアクセスログは、CloudWatch コンソールの [ロググループ] で確認できます。詳細については、「Amazon CloudWatch ユーザーガイド」の「CloudWatch Logs に送信されたログデータを表示する」を参照してください。

Amazon S3

- VPC Lattice は通常 6 分以内に Amazon S3 にログを配信します。ただし、実際のログ配信時間はベストエフォートベースであり、さらにレイテンシーが発生する可能性があることに注意してください。
- バケットに特定の権限がない場合は、バケットポリシーが自動的に作成され、Amazon S3 バケットに追加されます。詳細については、「Amazon CloudWatch ユーザーガイド」の「Amazon S3 に送信されたログ」を参照してください。

- Amazon S3 に送信されるアクセスログには、次の命名規則が使用されます。

```
[bucket]/[prefix]/AWSLogs/[accountId]/VpcLattice/AccessLogs/[region]/[YYYY/
MM/DD]/[resource-id]/[accountId]_VpcLatticeAccessLogs_[region]_[resource-
id]_YYYYMMDDTHHmmZ_[hash].json.gz
```

- Amazon S3 に送信される VpcLatticeResourceAccessLogs は、次の命名規則を使用します。

```
[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 Lattice は通常、2 分以内に Firehose にログを配信します。ただし、実際のログ配信時間はベストエフォートベースであり、さらにレイテンシーが発生する可能性があることに注意してください。
- アクセスログを Amazon Data Firehoseに送信する権限を VPC Lattice に付与するサービスにリンクされたロールが自動的に作成されます。この自動ロール作成が正常に行われるには、ユーザーが `iam:CreateServiceLinkedRole` アクションに対する許可を持っている必要があります。詳細については、「Amazon CloudWatch ユーザーガイド」の「Amazon Data Firehoseにログを送信する」を参照してください。
- Amazon Data Firehoseに送信されたログの表示の詳細については、「Amazon Data Firehose デベロッパーガイド」の「Amazon Kinesis Data Streams のモニタリング」を参照してください。

アクセスログの有効化

次の手順を実行して、アクセスログを取得し、選択した宛先に配信するように設定します。

内容

- コンソールを使用してアクセスログを有効にする
- を使用してアクセスログを有効にする AWS CLI

コンソールを使用してアクセスログを有効にする

作成時に、サービスネットワーク、サービス、またはリソース設定のアクセスログを有効にできます。次の手順で説明するように、サービスネットワーク、サービス、またはリソース設定を作成した後にアクセスログを有効にすることもできます。

コンソールを使用して基本サービスを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. サービスネットワーク、サービス、またはリソースの設定を選択します。
3. [アクション]、[ログ設定を編集] の順に選択します。
4. [アクセスログ] トグルスイッチをオンにします。
5. アクセスログの配信先を次のように追加します。
   - [CloudWatch ロググループ] を選択し、ロググループを選択します。ロググループを作成するには、[CloudWatch でロググループを作成する] を選択します。
   - [S3 バケット] を選択し、プレフィックスを含む S3 バケットパスを入力します。S3 バケットを検索するには、[S3 を参照] を選択します。
   - [Kinesis Data Firehose 配信ストリーム] を選択し、配信ストリームを選択します。配信ストリームを作成するには、[Kinesis で配信ストリームを作成] を選択します。
6. [Save changes] (変更の保存) をクリックします。

を使用してアクセスログを有効にする AWS CLI

CLI コマンド create-access-log-subscription を使用して、サービスネットワークまたはサービスのアクセスログを有効にします。

リクエストの追跡

VPC Lattice は、x-amzn-requestid ヘッダーによるオブザーバビリティとデバッグのために、クライアント、ターゲット、ログ間のリクエストの追跡と相関をサポートします。このヘッダーは、クライアントによって設定および送信することも、VPC Lattice によって生成することもでき、ターゲットに送信され、アクセスログでも使用できます。

デフォルトの 動作

- VPC Lattice は、リクエストごとにこのヘッダーを自動的に生成します。

- 値はランダムに生成された識別子です (デフォルトでは UUID スタイル) 。
- 生成された識別子は次のとおりです。
  - ダウンストリームターゲットに伝播されます。
  - レスポンスヘッダーでクライアントに返されます。
  - アクセスログへのログイン

例 (デフォルトのレスポンス)

以下に、VPC Lattice のデフォルト動作でクライアントに送信されるレスポンスの例を示します。このレスポンスは、valu eof x-amzn-requestid ヘッダーのランダムな値を生成します。

```
{
    "HTTP/1.1 200 OK
    x-amzn-requestid: a9f2c7a1-6b4f-4c79-9e87-ff5a1234a001"
}
```

値を設定するクライアント

- クライアントは、オプションで受信リクエストにこのヘッダーを設定して、自動的に生成された値を上書きできます。
- 考慮事項
  - ヘッダー値は UUID 形式に従う必要はありません。
  - ヘッダー値が 512 バイトを超える場合、VPC Lattice はそれを 512 に切り捨てます。

- 正常に上書きされると、指定されたヘッダー値は次のようになります。
  - レスポンスヘッダーに表示される
  - ターゲットに伝達される
  - アクセスログとメトリクスに表示される

例 (クライアントリクエストの上書き)

以下は、ヘッダー値を持つクライアントから送信されたリクエストの例です。

```
{
    "GET /my-service/endpoint HTTP/1.1
    Host: my-api.example.com
```

```
    x-amzn-requestid: trace-request-foobar"
}
```

例 (デフォルトのオーバーライドレスポンス)

以下は、オーバーライドされた値でクライアントに送信されるレスポンスの例です。

```
{
    "HTTP/1.1 200 OK
    x-amzn-requestid: trace-request-foobar"
}
```

アクセスログの内容

次の表は、アクセスログのエントリのフィールドを示しています。

フィールド	説明	形式
`callerPrincipalTags`	リクエストの PrincipalTags。	JSON
`hostHeader`	リクエストの権限ヘッダー。	string
`sslCipher`	クライアント TLS 接続を確立するために使用される暗号セットの OpenSSL 名。	string
`serviceNetworkArn`	サービスネットワーク ARN。	arn:aws:vpc-lattice:*region*:*account*:servicenetwork/*id*
`resolvedUser`	認証が有効な場合に認証が行われたときのユーザーの ARN。	null | ARN | "Anonymous" | "Unknown"
`authDeniedReason`	認証が有効な場合にアクセスが拒否される理由。	null | "Service" | "Network" | "Identity"
`requestMethod`	リクエストのメソッドヘッダー。	string

フィールド	説明	形式
targetGroupArn	ターゲットホストが属するターゲットホストグループ。	string
tlsVersion	TLS バージョン。	TLSv*x*
userAgent	ユーザーエージェントヘッダー。	string
serverNameIndication	[HTTPS のみ] ssl 接続ソケットに設定された Server Name Indication (SNI) の値。	string
destinationVpcId	送信先 VPC ID。	vpc-*xxxxxxxx*
sourceIpPort	送信元の IP アドレスとポート。	*ip*:*port*
targetIpPort	ターゲットの IP アドレスとポート。	*ip*:*port*
serviceArn	サービス ARN。	arn:aws:vpc-lattice:*region*:*account*:service/*id*
sourceVpcId	ソース VPC ID。	vpc-*xxxxxxxx*
requestPath	リクエストのパス。	LatticePath?:*path*
startTime	リクエストの開始時刻。	*YYYY*-*MM*-*DD*T*HH*:*MM*:*SS*Z
protocol	プロトコル。現在は HTTP/1.1 または HTTP/2。	string

フィールド	説明	形式
responseCode	HTTP レスポンスコード。最終ヘッダーのレスポンスコードのみが記録されます。詳細については、「アクセスログのトラブルシューティング」を参照してください。	integer
bytesReceived	受信した本文とヘッダーのバイト数。	integer
bytesSent	送信された本文とヘッダーのバイト数。	integer
duration	リクエストの開始時刻から最後のバイトが送信されるまでの合計期間 (ミリ秒単位)。	integer
requestToTargetDuration	リクエストの開始時刻から最後のバイトがターゲットに送信されるまでの合計期間 (ミリ秒単位)。	integer
responseFromTargetDuration	リクエストの最初のバイトがターゲットホストから読み取られてから、最後のバイトがクライアントに送信されるまでの合計期間 (ミリ秒単位)。	integer
grpcResponseCode	gRPC レスポンスコード。詳細については、「Status codes and their use in gRPC」を参照してください。このフィールドは、サービスが gRPC をサポートしている場合にのみ記録されます。	integer

フィールド	説明	形式
requestId	これは、x-amzn-requestid ヘッダーの値としてレスポンスに自動的に含まれる一意の識別子です。これにより、クライアント、ターゲット、ログ間のリクエストの相関関係が可能になり、オブザーバビリティとデバッグが可能になります。	string
callerPrincipal	認証されたプリンシパル。	string
callerX509SubjectCN	サブジェクト名 (CN)。	string
callerX509IssuerOU	発行者 (OU)。	string
callerX509SANNameCN	発行者の代替 (名前/CN)。	string
callerX509SANDNS	サブジェクト代替名 (DNS)。	string
callerX509SANURI	サブジェクト代替名 (URI)。	string
sourceVpcArn	リクエストが発生した VPC の ARN。	arn:aws:ec2:*region*:*account*:vpc/*id*

フィールド	説明	形式
failureReason	リクエストが失敗した理由を示します。取り得る値には以下のものがあります。 • TargetConnectionError - リクエストがターゲットグループのターゲットに接続できませんでした。 • TargetProtocolError - ターゲットが有効なデータで応答しませんでした。これは、ターゲットに無効な TLS レコードがあるか、無効なターゲットグループプロトコルを使用していることを示している可能性があります。 • TargetDataTimeout - アイドルタイムアウトに達しました。 • TargetConnectionClosed - ターゲットは、レスポンスを完了する前に接続を閉じました。 • ClientConnectionClosed - クライアントは、完全なレスポンスを受信する前に接続を閉じました。 • ClientRateLimited - クライアントが接続制限を超え、VPC Lattice がレートを制限しました。	string

フィールド	説明	形式
	• ClientAccessDenied - VPC Lattice がリソースへのアクセスを拒否しました。VPC Lattice がアクセスを拒否した理由の詳細については、 authDeniedReason を使用します。 • ClientProtocolError - クライアントは、理解されていないデータを送信しました。これは、クライアントが無効な TLS レコードまたは無効なプロトコルを使用したことを示している可能性があります。 • ConnectionDurationExceeded - 接続が接続期間の上限に達しました。 • InternalError - リクエストの処理中に内部エラーが発生しました。	

例

ログエントリの例を示します。

```
{
    "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"
}
```

リソースアクセスログの内容

次の表に、リソースアクセスログエントリのフィールドを示します。

フィールド	説明	形式
serviceNetworkArn	サービスネットワーク ARN。	arn:*partition* vpc-lattice:*region*:*account*:servicenetwork/*id*
serviceNetworkResourceAssociationId	サービスネットワークリソース ID。	*snra-xxx*
vpcEndpointId	リソースへのアクセスに使用されたエンドポイント ID。	string
sourceVpcArn	接続が開始されたソース VPC ARN または VPC。	string

フィールド	説明	形式
resourceConfigurationArn	アクセスされたリソース設定の ARN。	string
protocol	リソース設定との通信に使用されるプロトコル。現在、tcp のみがサポートされています。	string
sourceIpPort	接続を開始したソースの IP アドレスとポート。	*ip*:*port*
destinationIpPort	接続が開始された IP アドレスとポート。これは SN-E/SN-A の IP になります。	*ip*:*port*
gatewayIpPort	リソースゲートウェイがリソースにアクセスするために使用する IP アドレスとポート。	*ip*:*port*
resourceIpPort	リソースの IP アドレスとポート。	*ip*:*port*

例

ログエントリの例を示します。

```
{
    "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"
}
```

アクセスログのトラブルシューティング

このセクションでは、アクセスログに表示される可能性のある HTTP エラーコードについて説明します。

エラーコード	考えられる原因
HTTP 400: Bad Request	• クライアントが HTTP 仕様を満たさない不正な形式のリクエストを送信した。 • リクエストヘッダーが全体で 60,000 を超えているか、ヘッダーが 100 を超えている。 • クライアントが、リクエスト本文全体を送信する前に接続を閉じた。
HTTP 403: Forbidden	認証はサービスに対して設定されているが、受信リクエストは認証も承認もされていない。
HTTP 404: Non Existent Service	存在しないサービスに接続しようとしているか、適切なサービスネットワークに登録されていない。
HTTP 500: 内部サーバーエラー	VPC Lattice でターゲットに接続できないなどのエラーが発生した。
HTTP 502: Bad Gateway	VPC Lattice でエラーが発生した。

Amazon VPC Lattice の CloudTrail ログ

Amazon VPC Lattice は、ユーザーAWS CloudTrail、ロール、または によって実行されたアクションを記録するサービスである と統合されています AWS のサービス。CloudTrail は、VPC Lattice のすべての API コールをイベントとしてキャプチャします。キャプチャされた呼び出しには、VPC Lattice コンソールからの呼び出しと、VPC Lattice API オペレーションへのコード呼び出しが含まれます。CloudTrail で収集された情報を使用して、VPC Lattice に対するリクエスト、リクエスト元の IP アドレス、リクエスト日時などの詳細を確認できます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- ルートユーザーまたはユーザー認証情報のどちらを使用してリクエストが送信されたか。
- リクエストが IAM Identity Center ユーザーに代わって行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。
- リクエストが、別の AWS のサービスによって送信されたかどうか。

CloudTrail は、アカウントを作成する AWS アカウント と でアクティブになり、CloudTrail イベント履歴に自動的にアクセスできます。CloudTrail の [イベント履歴] では、 AWS リージョンで過去 90 日間に記録された管理イベントの表示、検索、およびダウンロードが可能で、変更不可能な記録を確認できます。詳細については、「AWS CloudTrail ユーザーガイド」の「CloudTrail イベント履歴の使用」を参照してください。[イベント履歴] の閲覧には CloudTrail の料金はかかりません。

AWS アカウント 過去 90 日間のイベントの継続的な記録については、証跡または CloudTrail Lake イベントデータストアを作成します。

CloudTrail 証跡

証跡により、CloudTrail はログファイルを Amazon S3 バケットに配信できます。を使用して作成されたすべての証跡 AWS マネジメントコンソール はマルチリージョンです。 AWS CLIを使用する際は、単一リージョンまたは複数リージョンの証跡を作成できます。アカウント AWS リージョン 内のすべての でアクティビティをキャプチャするため、マルチリージョン証跡を作成することをお勧めします。単一リージョンの証跡を作成する場合、証跡の AWS リージョンに記録されたイベントのみを表示できます。証跡の詳細については、「AWS CloudTrail ユーザーガイド」の「AWS アカウントの証跡の作成」および「組織の証跡の作成」を参照してください。

証跡を作成すると、進行中の管理イベントのコピーを 1 つ無料で CloudTrail から Amazon S3 バケットに配信できますが、Amazon S3 ストレージには料金がかかります。CloudTrail の料金の詳細については、「AWS CloudTrail の料金」を参照してください。Amazon S3 の料金に関する詳細については、「Amazon S3 の料金」を参照してください。

CloudTrail Lake イベントデータストア

[CloudTrail Lake] を使用すると、イベントに対して SQL ベースのクエリを実行できます。CloudTrail Lake は、行ベースの JSON 形式の既存のイベントを Apache ORC 形式に変換します。ORC は、データを高速に取得するために最適化された単票ストレージ形式です。イベントは、イベントデータストアに集約されます。イベントデータストアは、高度なイベントセレク

タを適用することによって選択する条件に基づいた、イベントのイミュータブルなコレクションです。どのイベントが存続し、クエリに使用できるかは、イベントデータストアに適用するセレクタが制御します。CloudTrail Lake の詳細については、 AWS CloudTrail ユーザーガイドのAWS CloudTrail「Lake の使用」を参照してください。

CloudTrail Lake のイベントデータストアとクエリにはコストがかかります。イベントデータストアを作成する際に、イベントデータストアに使用する料金オプションを選択します。料金オプションによって、イベントの取り込みと保存にかかる料金、および、そのイベントデータストアのデフォルトと最長の保持期間が決まります。CloudTrail の料金の詳細については、「AWS CloudTrail の料金」を参照してください。

その他のアクションをモニタリングするには、アクセスログを使用します。詳細については、「アクセスログ」を参照してください。

CloudTrail での VPC Lattice 管理イベント

管理イベントは、 のリソースで実行される管理オペレーションに関する情報を提供します AWS アカウント。これらのイベントは、コントロールプレーンオペレーションとも呼ばれます。CloudTrail は、デフォルトで管理イベントをログ記録します。

Amazon VPC Lattice は、VPC Lattice コントロールプレーンオペレーションを管理イベントとしてログに記録します。VPC Lattice が CloudTrail にログ記録する Amazon VPC Lattice コントロールプレーンオペレーションのリストについては、「Amazon VPC Lattice API リファレンス」を参照してください。

VPC Lattice イベントの例

各イベントは任意の送信元からの単一のリクエストを表し、リクエストされた API オペレーション、オペレーションの日時、リクエストパラメータなどに関する情報を含みます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、イベントは特定の順序で表示されません。

次の例は、CreateService オペレーションの CloudTrail イベントを示しています。

```
{
  "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"
}
```


次の例は、DeleteService オペレーションの CloudTrail イベントを示しています。

```
{
  "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 レコードの内容については、「AWS CloudTrail ユーザーガイド」の「CloudTrail record contents」を参照してください。

Amazon VPC Lattice のクォータ

AWS アカウント には、各 の制限と呼ばれるデフォルトのクォータがあります AWS のサービス。特に明記していない限り、クォータはリージョン固有です。一部のクォータの増加を要求できますが、他のクォータは増加できません。

VPC Lattice のクォータを表示するには、Service Quotas コンソールを開きます。ナビゲーションペインで [AWS のサービス] を選択し、[VPC Lattice] を選択します。

クォータの引き上げをリクエストするには、「Service Quotas ユーザーガイド」の「クォータ引き上げリクエスト を参照してください。

AWS アカウント には、VPC Lattice に関連する次のクォータがあります。

名前	デフォルト	引き上げ可能	説明
認可ポリシーのサイズ	サポートされている各リージョン: 10 KB	不可	認可ポリシー内の JSON ファイルの最大サイズ。
グループリソース設定あたりの子リソース設定数	サポートされている各リージョン: 60	あり	グループリソース設定内の子リソース設定の最大数。追加の容量と制限の引き上げについては、AWS サポートにお問い合わせください。
AWS リージョンあたりのドメイン検証	サポートされている各リージョン : 5	あり	アカウントごとに作成できるドメイン検証の最大数。追加の容量と制限の引き上げについては、AWS サポートにお問い合わせください。

名前	デフォルト	引き上げ可能	説明
サービスあたりのリスナー	サポートされている各リージョン: 2	あり	サービスのために作成できるリスナーの最大数。追加の容量と制限の引き上げについては、 AWS サポートにお問い合わせください。
サービスネットワークあたりのリソース設定数	サポートされている各リージョン: 500	あり	サービスネットワークに関連付けられたリソース設定の最大数。追加の容量と制限の引き上げについては、 AWS サポートにお問い合わせください。
AWS リージョンあたりのリソース設定	サポートされている各リージョン: 2,000	あり	AWS アカウントが AWS リージョンごとに持つことができるリソース設定の最大数。追加の容量と制限の引き上げについては、 AWS サポートにお問い合わせください。
VPC あたりのリソースゲートウェイ数	サポートされている各リージョン: 500	あり	VPC 内のリソースゲートウェイの最大数。追加の容量と制限の引き上げについては、 AWS サポートにお問い合わせください。

名前	デフォルト	引き上げ可能	説明
リスナーあたりのルール	サポートされている各リージョン: 10	あり	サービスリスナーのために定義できるルールの最大数。追加の容量と制限の引き上げについては、AWS サポートにお問い合わせください。
関連付けあたりのセキュリティグループ	サポートされている各リージョン : 5	不可	VPC とサービスネットワーク間の関連付けに追加できるセキュリティグループの最大数。
サービスネットワークあたりのサービスの関連付け	サポートされている各リージョン: 500	あり	1 つのサービスネットワークに関連付けることができるサービスの最大数。追加の容量と制限の引き上げについては、AWS サポートにお問い合わせください。
リージョンあたりのサービスネットワーク	サポートされている各リージョン: 50	可能	リージョンあたりのサービスネットワークの最大数。追加の容量と制限の引き上げについては、AWS サポートにお問い合わせください。

名前	デフォルト	引き上げ可能	説明
リージョンあたりのサービス	サポートされている各リージョン: 2,000	あり	リージョンあたりのサービスの最大数。追加の容量と制限の引き上げについては、 AWS サポートにお問い合わせください。
リージョンあたりのターゲットグループ	サポートされている各リージョン: 500	あり	リージョンあたりのターゲットグループの最大数。追加の容量と制限の引き上げについては、 AWS サポートにお問い合わせください。
サービスあたりのターゲットグループ	サポートされている各リージョン: 10	あり	サービスに関連付けることができるターゲットグループの最大数。追加の容量と制限の引き上げについては、 AWS サポートにお問い合わせください。
ターゲットグループあたりのターゲット	サポートされている各リージョン: 1,000	あり	1 つのターゲットグループに関連付けることができるターゲットの最大数。追加の容量と制限の引き上げについては、 AWS サポートにお問い合わせください。

名前	デフォルト	引き上げ可能	説明
サービスネットワークあたりの VPC の関連付け	サポートされている各リージョン: 500	あり	1 つのサービスネットワークに関連付けることができる VPC の最大数。追加の容量と制限の引き上げについては、AWS サポートにお問い合わせください。
サービスネットワークあたりのサービスネットワークタイプの VPC エンドポイント数	サポートされている各リージョン: 200	あり	サービスネットワークに関連付けられているサービスネットワークエンドポイントの最大数。追加の容量と制限の引き上げについては、 AWS サポートにお問い合わせください。

VPC Lattice では、次のアベイラビリティーゾーンはサポートされていません: use1-az3、usw1-az2、apne1-az3、、、apne2-az2euc1-az2euw1-az4cac1-az3、、ilc1-az2。

以下の制限も適用されます。

制限	値	説明
アベイラビリティーゾーンごとのサービスあたりの帯域幅	10 Gbps	アベイラビリティーゾーンごとにサービスごとに割り当てられる最大帯域幅。
接続あたりの最大送信単位 (MTU)	8500 バイト	サービスが受け入れることができる最大のデータパケットのサイズ。

制限	値	説明
アベイラビリティーゾーンあたりのサービスごとの 1 秒あたりのリクエスト数	10,000	HTTP サービスの場合、これはアベイラビリティーゾーンあたりのサービスごとの 1 秒あたりのリクエストの最大数です。
VPC Lattice サービスの接続あたりの接続アイドル時間	1 分	アクティブなリクエストがない (HTTP および GRPC の場合) 、または VPC Lattice サービスのアクティブなデータ転送がない (TLS-PASS THROUGH の場合) 接続をアイドル状態にできる最大時間。HTTP およびアプリケーションレベルのキープアライブを使用して、このアイドルタイムアウトを最大接続有効期間まで延長できます。
VPC Lattice サービスの接続あたりの最大接続有効期間	10 分	VPC Lattice サービスのクライアントとサーバー間の接続を開くことができる最大時間。
VPC Lattice リソースの接続あたりの最大接続有効期間	NA	VPC Lattice は、リソースのライフタイム接続制限を課しません。クライアントとサーバーは、VPC Lattice リソースのアイドルタイムアウトである 350 秒を認識しながら、ライフタイム接続期間を決定します。
VPC Lattice リソースの接続あたりの接続アイドル時間	350 秒	TCP キープアライブを使用して、このアイドルタイムアウトを延長できます。

制限	値	説明
VPC あたりのサービスネットワーク	1 つのサービスネットワーク	VPC は、関連付けを通じて 1 つのサービスネットワークにのみ接続できます。VPC を複数のサービスネットワークに接続するには、サービスネットワークタイプの VPC エンドポイントを使用できます。

Amazon VPC Lattice ユーザーガイドのドキュメント履歴

次の表は VPC Lattice のドキュメントリリースをまとめたものです。

変更	説明	日付
リソースゲートウェイの設定可能な IP アドレスを追加	VPC Lattice は、リソースゲートウェイの設定可能な IP アドレスをサポートするようになりました。	2025 年 10 月 7 日
の VPC Lattice を追加 Oracle Database@AWS	VPC Lattice for が Oracle Database@AWS リリースされました。	2025 年 6 月 26 日
管理エンドポイントのデュアルスタックサポートを追加	VPC Lattice は、すべての VPC Lattice 管理 APIs のデュアルスタック (IPv4 および IPv6) エンドポイントをサポートするようになりました。	2025 年 4 月 30 日
リソースの共有とアクセス	VPC Lattice は、VPC とアカウントの境界間でのリソースの共有とアクセスをサポートするようになりました。これには、VPCLatticeReadOnlyAccess ポリシーと VPCLatticeFullAccess ポリシーの更新が含まれます。	2024 年 12 月 1 日
TLS パススルー	VPC Lattice は TLS パススルーをサポートするようになりました。これにより、end-to-end認証のためにアプリケーションで TLS 終了を実行できます。	2024 年 5 月 14 日

Lambda イベント構造のバージョン	VPC Lattice は新しいバージョンの Lambda イベント構造をサポートするようになりました。	2023 年 9 月 7 日
共有 VPC のサポート	参加者は VPC Lattice ターゲットグループを共有 VPC に作成できます。	2023 年 7 月 5 日
一般提供リリース	一般提供 (GA) 向けの VPC Lattice ユーザーガイドをリリースしました。	2023 年 3 月 31 日
VPC Lattice が AWS 管理ポリシーの変更を報告するようになりました	管理ポリシーへの変更は、「セキュリティ」章の「VPC Lattice AWS の管理ポリシー」で報告されます。	2023 年 3 月 29 日
Application Load Balancer のターゲットタイプのサポート	VPC Lattice で Application Load Balancer タイプのターゲットグループの作成をサポートするようになりました。	2023 年 3 月 29 日
すべてのインスタンスタイプのサポート	VPC Lattice ですべてのインスタンスタイプがサポートされるようになりました。	2023 年 3 月 27 日
IPv6 サポート	VPC Lattice で IPv4 と IPv6 のターゲットグループの両方がサポートされるようになりました。	2023 年 3 月 27 日
ヘルスチェック用の HTTP2 プロトコルバージョン	ターゲットグループのプロトコルバージョンが HTTP2 の場合、ヘルスチェックがサポートされるようになりました。	2023 年 3 月 27 日

リスナールールの固定レスポンスアクション	VPC Lattice サービスのリスナーは、転送アクションに加えて固定レスポンスアクションをサポートするようになりました。	2023 年 3 月 27 日
カスタムドメイン名のサポート	VPC Lattice サービスのカスタムドメイン名を設定できるようになりました。	2023 年 2 月 14 日
BYOC (独自の証明書の持ち込み) のサポート	VPC Lattice ではカスタムドメイン名に ACM の独自の SSL/TLS 証明書を使用できます。	2023 年 2 月 14 日
VPC Lattice によるサポートされていないインスタンスタイプの更新済みリストの報告を開始	3 つのインスタンスがサポートされていないインスタンスのリストに追加されています。	2023 年 1 月 26 日
VPC Lattice が AWS 管理ポリシーの変更を報告するようになりました	2022 年 12 月 5 日より、マネージドポリシーの変更は「セキュリティ」の章の「VPC Lattice のAWS マネージドポリシー」トピックで報告されます。リストされている最初の変更は CloudWatch モニタリングに必要な許可の追加です。	2022 年 12 月 5 日
初回リリース	VPC Lattice ユーザーガイドの初回リリース。	2022 年 12 月 5 日

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。