ユーザーガイド

Amazon VPC Lattice

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon VPC Lattice: ユーザーガイド

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは、お客様に混乱を招く可能性がある態様、または Amazon の信用を傷つけたり、失わせたりする態様において、Amazon のものではない製品またはサービスに関連して使用してはなりません。Amazon が所有しない他の商標はすべてそれぞれの所有者に帰属します。所有者は必ずしも Amazon との提携や関連があるわけではありません。また、Amazon の支援を受けているとはかぎりません。

Table of Contents

Amazon VPC Lattice とは 1
主要コンポーネント 1
役割と責任 3
機能 4
VPC Lattice の仕組み 5
VPC Lattice へのアクセス 8
料金 8
セットアップ 9
Sign up for AWS 9
IAM ユーザーを作成する 9
サービスネットワーク 11
サービスネットワークを作成する 12
関連付けを管理する 14
サービスの関連付けを管理する 14
VPC の関連付けを管理する 15
アクセス設定を編集する 16
モニタリングの詳細を編集する 17
タグの管理 18
サービスネットワークを削除する 19
サービス 20
ステップ 1: VPC Lattice サービスを作成する 21
ステップ 2: ルーティングを定義する 22
ステップ 3: ネットワークの関連付けを作成する 23
ステップ 4: 確認して作成する 24
関連付けを管理する 24
アクセス設定を編集する 25
モニタリングの詳細を編集する 26
タグの管理 27
カスタムドメイン名を設定する 28
カスタムドメイン名をサービスに関連付ける 30
BYOC 32
証明書のプライベートキーを保護する 34
サービスを削除する 34
ターゲットグループ 36

ターゲットグループの作成 ... 37
ターゲットグループの作成 ... 37
共有サブネット ... 39
ターゲットの登録 ... 40
インスタンス ID ... 41
IP アドレス ... 41
Lambda 関数 ... 42
アプリケーション ロード バランサー ... 42
ヘルスチェックを設定する ... 43
ヘルスチェックの設定 ... 44
ターゲットのヘルスステータスをチェックする ... 45
ヘルスチェックの設定を変更する ... 46
ルーティング設定 ... 47
ルーティングアルゴリズム ... 47
[Target type (ターゲットタイプ)] ... 48
IP アドレスタイプ ... 49
HTTP ターゲット ... 49
x-forwarded ヘッダー ... 49
発信者 ID ヘッダー ... 50
ターゲットとしての Lambda 関数 ... 51
Lambda 関数の準備 ... 51
Lambda 関数のターゲットグループの作成 ... 42
VPC Lattice サービスからのイベントを受け取る ... 53
VPC Lattice サービスへのレスポンス ... 56
複数値ヘッダー ... 57
Lambda 関数の登録解除 ... 57
ターゲットとしての Application Load Balancer ... 58
前提条件 ... 58
ステップ 1: ALB タイプのターゲットグループを作成する ... 59
ステップ 2: Application Load Balancer をターゲットとして登録する ... 60
プロトコルバージョン ... 60
タグの更新 ... 62
ターゲットグループの削除 ... 63
リスナー ... 64
リスナーの設定 ... 64
リスナーの作成 ... 65

HTTP リスナー 65
前提条件 66
HTTP リスナーを追加する 66
HTTPS リスナー 67
セキュリティポリシー 68
ALPN ポリシー 69
HTTPS リスナーの追加 69
リスナールール 71
デフォルトのルール 71
ルールの優先順位 71
ルールアクション 71
ルールの条件 72
ルールの追加 73
ルールを更新する 74
ルールの削除 74
リスナーの更新 75
リスナーの削除 75
VPC Lattice リソースを共有する 77
前提条件 77
リソースを共有する 78
リソースの共有を停止する 79
責任と権限 79
リソース所有者 79
リソースコンシューマー 80
クロスアカウントイベント 81
セキュリティ 84
サービスへのアクセスを管理する 85
認証ポリシー 85
セキュリティグループ 100
ネットワーク ACL 105
認証済みリクエスト 107
データ保護 115
転送中の暗号化 116
保管中の暗号化 116
ID およびアクセス管理 122
Amazon VPC Lattice で IAM が機能する仕組み 123

API アクセス許可 131
アイデンティティベースのポリシー 132
サービスリンクロールの使用 138
AWS 管理ポリシー 140
コンプライアンス検証 143
AWS PrivateLink 144
インターフェイス VPC エンドポイントに関する考慮事項 145
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 145
耐障害性 145
インフラストラクチャセキュリティ 146
モニタリング 147
CloudWatch メトリクス 147
Amazon CloudWatch メトリクスを表示する 147
ターゲットグループのメトリクス 148
サービスメトリクス 162
アクセスログ 166
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 167
アクセスログの送信先 168
アクセスログの有効化 169
アクセスログの内容 170
アクセスログのトラブルシューティング 174
CloudTrail ログ 174
VPC Lattice のログファイルエントリを理解する 175
クォータ 178
ドキュメント履歴 182
..... clxxxiv

Amazon VPC Lattice とは

Amazon VPC Lattice は、アプリケーションのサービスの接続、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、マイクロサービスと呼ばれる、複数の小型でモジュラー型のサービスで構成されることがあります。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を接続すると、ネットワークが複雑になり、課題が生じることもあります。例えば、デベロッパーが異なるチームに分散している場合、複数のアカウントや VPC にわたってマイクロサービスを構築してデプロイすることになります。

VPC Lattice では、マイクロサービスをサービスと呼んでいます。これは VPC Lattice ドキュメントに記載されている表現です。

コンテンツ

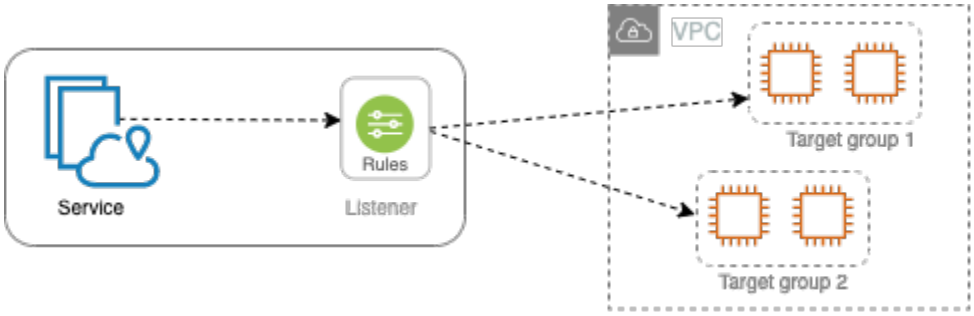
- 主要コンポーネント
- 役割と責任
- 機能
- VPC Lattice の仕組み
- VPC Lattice へのアクセス
- 料金

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) 内で、EC2 インスタンスまたは ECS コンテナで実行できるほか、Lambda 関数として実行することもできます。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。ターゲットには、EC2 インスタンス、IP アドレス、Lambda 関数、Application Load Balancer、Kubernetes ポッドを指定できます。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。

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。1 つのサービスには、HTTP プロトコルおよび HTTPS プロトコルと 1 ~ 65535 のポート番号を使用する、最大 2 つのリスナーを設定できます。

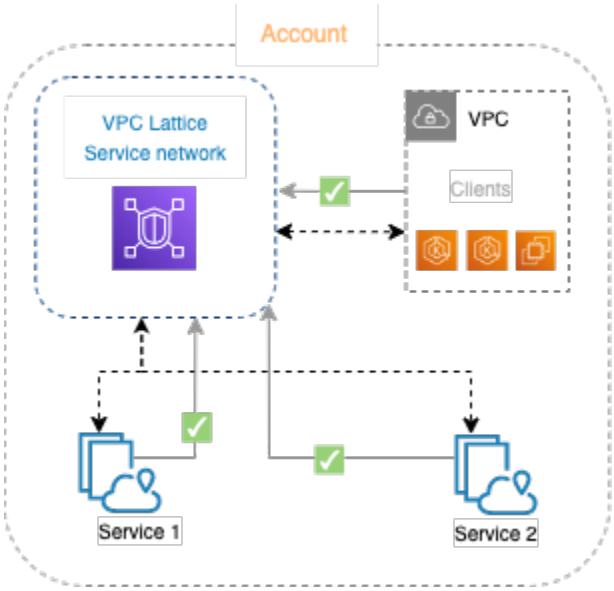
ルール

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。

サービスネットワーク

サービスの集合の論理的な境界です。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 にデプロイされたリソースです。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

AWS Resource Access Manager () を通じて所有または共有しているすべての VPC Lattice サービスの中央レジストリAWS RAM。

認証ポリシー

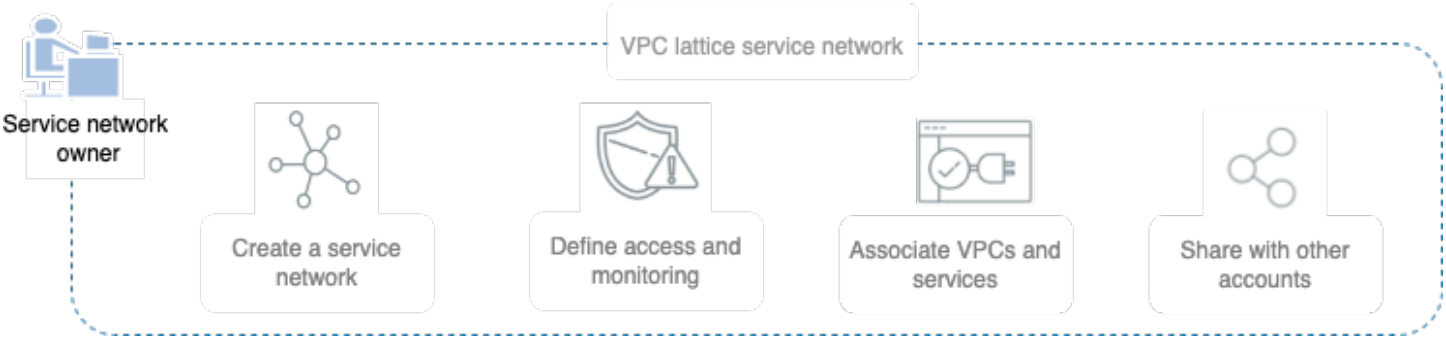
サービスへのアクセスを定義するために使用できる、きめ細やかな認証ポリシーです。個別の認証ポリシーを個々のサービスまたはサービスネットワークにアタッチできます。例えば、EC2 インスタンスの Auto Scaling グループで実行されている支払いサービスと、 AWS Lambdaで実行されている請求サービスとのやり取りを定めたポリシーを作成できます。

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承認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、サービスネットワーク所有者はサービスを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 — 通常、サービス所有者は組織における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。

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、VPC Lattice エンドポイントを介して client-to-service および service-to-service トラフィックを送信します。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

C lient-to-service 接続は、 AWS ネットワークインフラストラクチャ内の VPC Lattice データプレーンを使用して確立されます。VPC をサービスネットワークに関連付けると、VPC 内のクラ

イアントは、必要なアクセス権を持っている場合、サービスネットワーク内のサービスに接続できます。

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、VPC Lattice はサービス所有者のアカウントでメトリクスを公開します。また、ログ記録を有効にするオプションもあります。クライアントも同じサービスネットワークに関連付けられている場合、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ているすべてのサービスのログを受け取ります。サービス所有者は、サービスにリクエストを送信したすべてのクライアントのログを受け取ります。

VPC Lattice は、 CloudWatch ロググループ、Firehose 配信ストリーム、S3 バケットといった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つツールと連携します。

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と VPC の関連付けです。VPC とサービスとの関連付けがなければ、クライアントはサービ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 の仕組み

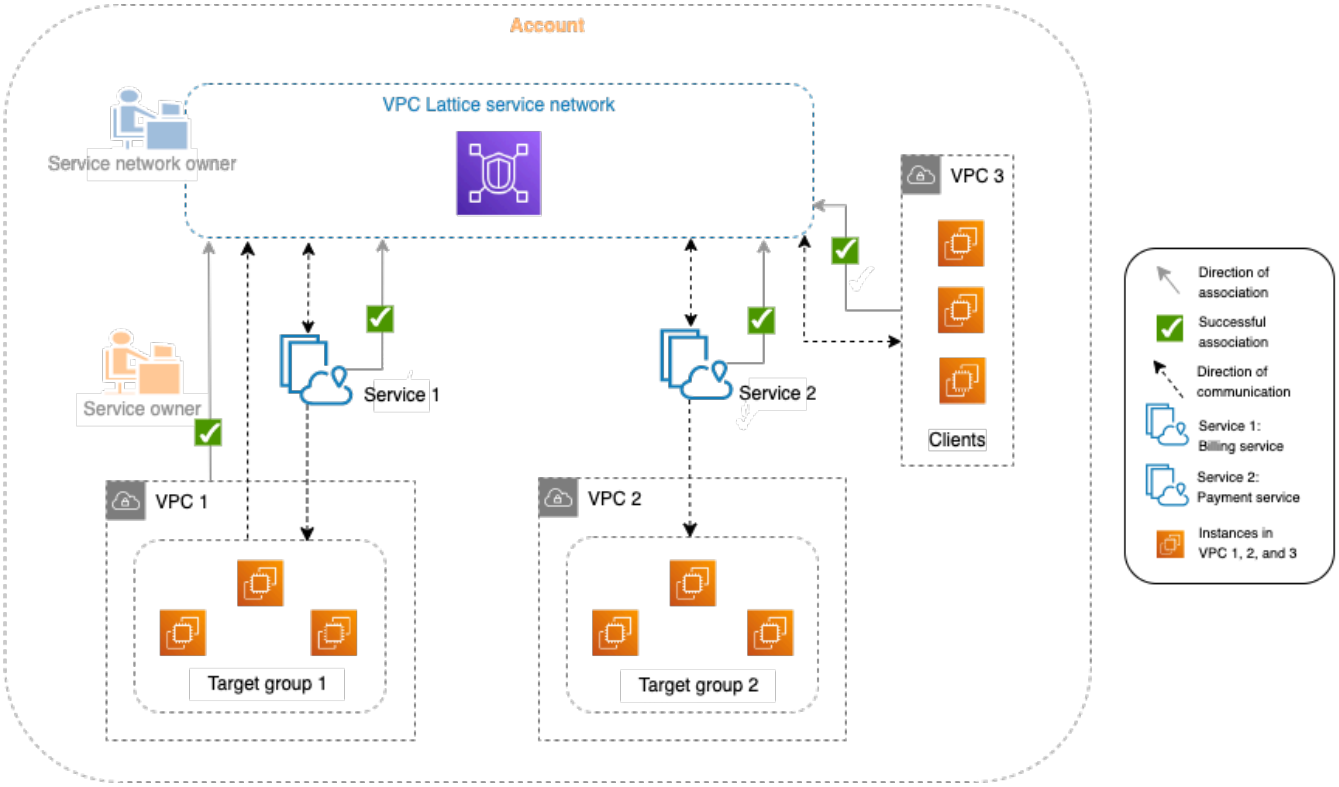
VPC Lattice は、VPC 内のすべてのサービスを簡単かつ効果的に検出、保護、接続、モニタリングできるように設計されています。VPC Lattice 内のそれぞれのコンポーネントは、サービスネットワークとの関連付けとアクセス設定に基づいて、サービスネットワーク内で単方向または双方向に通信します。アクセス設定は、この通信に必要な認証ポリシーおよび承認ポリシーで構成されます。

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。

- サービスネットワークに関連付けられているサービスは、VPC がその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からリクエストを受け取ることができます。
- クライアントは、同じサービスネットワークに関連付けられている VPC 内にある場合に限り、サービスネットワークに関連付けられたサービ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拒否されます。

- 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他の VPC のクライアントにリクエストを送信できません。
- サービスネットワークに関連付けられている VPC 内のサービスのターゲットはクライアントでもあるため、サービスネットワークに関連付けられている他のサービ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 VPC 内のサービスのターゲットはクライアントではないため、サービスネットワークに関連付けられている他のサービ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。両方のサービスと 3 つの VPC はすべて、サービスネットワークと同じアカウントで作成されています。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。

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。その際には、サービスが実行されるターゲットリソース (インスタンスなど) 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。

   上の図の、サービスのターゲットグループを指す点線の矢印は、各サービスからそれぞれのターゲットグループに流れるトラフィックを表します。点線の矢印は、サービスとターゲットグループ間の通信の方向を表します。

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。サービスネットワークにサービスを関連付けると、そのサービスは、そのサービスネットワークに関連付けられている VPC 内の他のサービスやクライアントから検出できるようになります。

   サービスとサービスネットワーク間にある双方向の点線の矢印は、関連付けによる双方向の通信を表します。サービスネットワークからサービスへの点線の矢印は、クライアントからのリクエストを受け取るサービスを表します。逆方向の点線の矢印、つまりサービスからサービスネットワークに向かう点線の矢印は、サービスネットワークを通じてクライアントのリクエストに応答するサービスを表します。

5. VPC 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が、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で表示されます。これらの関連付けにより、これらの VPC 内のターゲットがクライアントになり、関連付けられたサービスにリクエストを送信できるようになります。VPC 3 とサービスネットワーク間にある双方向の点線の

矢印は、関連付けの結果としての VPC 3 のクライアント (インスタンスなど) とサービスネットワーク間の双方向の通信を表します。同様に、ターゲットグループ 1 からサービスネットワークを指す点線の矢印は、サービスネットワークに関連付けられた他のサービスにリクエストを行うクライアントを表します。

VPC 2 には関連付けを表す矢印やチェックマークがありません。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。

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 Management Console - 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 を含むさまざまな AWS のサービス用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。
- AWS 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。

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。詳細については、「Amazon VPC Lattice の料金」を参照してください。

Amazon VPC Lattice をセットアップする

VPC Lattice を初めてセットアップして起動するには、このセクションのタスクを完了します。

タスク

- Sign up for AWS
- IAM ユーザーを作成する

Sign up for AWS

Amazon Web Services にサインアップすると、VPC Lattice を含む AWS 内のすべてのサービスを使用するために、AWS アカウント が自動的に登録されます。料金は、使用するサービスの料金のみが請求されます。

AWS アカウント が既にある場合は、次のタスクに進んでください。AWS アカウント をお持ちでない場合は、次に説明する手順にしたがってアカウントを作成してください。

AWS アカウント をお持ちでない場合は、以下の手順を実行してアカウントを作成してください。

AWS アカウント にサインアップするには

1. https://portal.aws.amazon.com/billing/signup を開きます。
2. オンラインの手順に従います。

   サインアップ手順の一環として、通話呼び出しを受け取り、電話のキーパッドを用いて検証コードを入力するように求められます。

   AWS アカウント にサインアップすると、AWS アカウントのルートユーザー が作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティのベストプラクティスとして、管理ユーザーに管理アクセスを割り当て、ルートユーザーのみを使用してルートユーザーアクセスが必要なタスクを実行してください。

IAM ユーザーを作成する

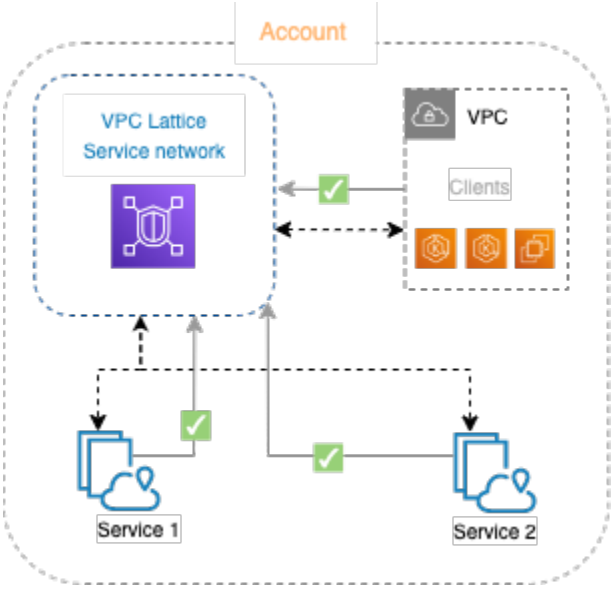
管理者ユーザーを作成するには、以下のいずれかのオプションを選択します。

管理者を管理する方法を 1 つ選択します	To	By	以下の操作も可能
IAM Identity Center 内 (推奨)	短期の認証情報を使用して AWS にアクセスします。 これはセキュリティのベストプラクティスと一致しています。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。	AWS IAM Identity Center ユーザーガイドの「開始方法」の手順に従います。	AWS Command Line Interface ユーザーガイドの「AWS IAM Identity Center を使用するための AWS CLI の設定」に従って、プログラムによるアクセスを設定します。
IAM 内 (非推奨)	長期認証情報を使用して AWS にアクセスする。	IAM ユーザーガイドの「最初の IAM 管理者のユーザーおよびグループの作成」の手順に従います。	IAM ユーザーガイドの「IAM ユーザーのアクセスキーの管理」に従って、プログラムによるアクセスを設定します。

VPC Lattice のサービスネットワーク

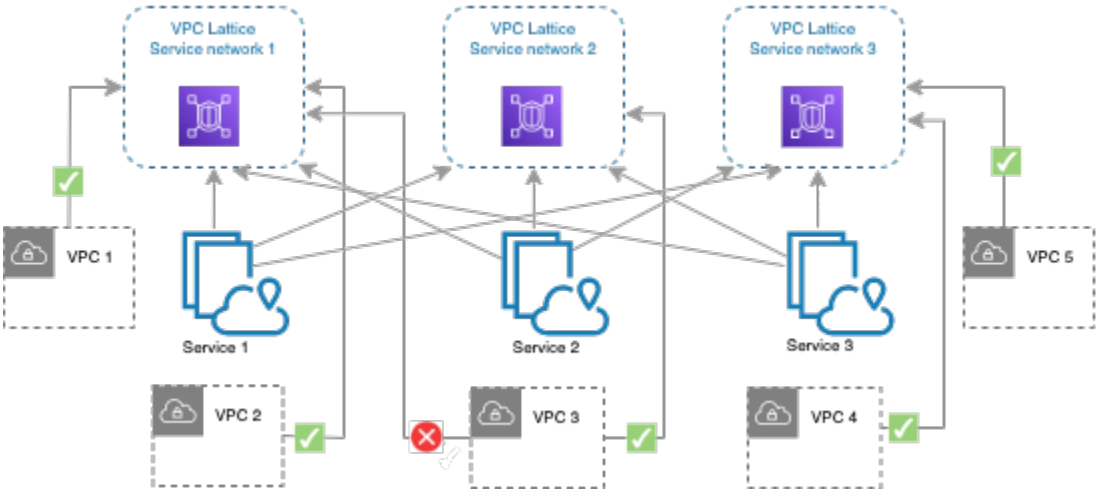
サービスネットワークは、サービスの集合の論理的な境界です。ネットワークに関連付けらたサービスは、検出、接続、アクセシビリティ、オブザーバビリティの認証を受けることができます。ネットワーク内のサービスにリクエストを行うには、サービスまたはクライアントが、サービスネットワークに関連付けられた VPC 内にある必要があります。

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。また、複数の VPC を 1 つのサービスネットワークに関連付けることができます。ただし、各 VPC は 1 つのサービスネットワークにのみ関連付けることができます。

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。ただし、図の赤い x マークは、各 VPC を 1 つのサービスネットワークに対して 1 つ以上関連付けることができないことを示しています。



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

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 (https://console.aws.amazon.com/vpc/) を開きます。
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) 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、サービスネットワークを作成した後に VPC を関連付ける方法については、「the section called “VPC の関連付けを管理する”」を参照してください。

7. [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。
8. (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、 でロググループを作成する CloudWatch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。
9. (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。
10. [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コマンドを実行します。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。

サービスネットワーク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) がサービスにリクエストを送信できるようになります。VPC をサービスネットワークに関連付けると、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と通信できるようになります。

コンテンツ

- サービスの関連付けを管理する
- VPC の関連付けを管理する

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 の仕組み」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 (https://console.aws.amazon.com/vpc/) を開きます。
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. [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 の関連付けを管理する

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 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拒否されます。

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。ただし、VPC 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。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 の仕組み」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 (https://console.aws.amazon.com/vpc/) を開きます。
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) 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。

e. [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 の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承認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。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、粒度の粗い認証を実装できます。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 (https://console.aws.amazon.com/vpc/) を開きます。

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. [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コマンドを実行します。

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、ロググループ、Firehose 配信ストリーム、S3 バケットのリソースに CloudWatch 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 (https://console.aws.amazon.com/vpc/) を開きます。
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:
   - 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、 でロググループを作成する CloudWatch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. [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コマンドを実行します。

を使用してログの送信先を更新するには AWS CLI

update-access-log-subscriptionコマンドを実行します。

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コマンドを実行します。

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 (https://console.aws.amazon.com/vpc/) を開きます。

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。

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には、まず、サービスネットワークとサービスまたは VPC 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

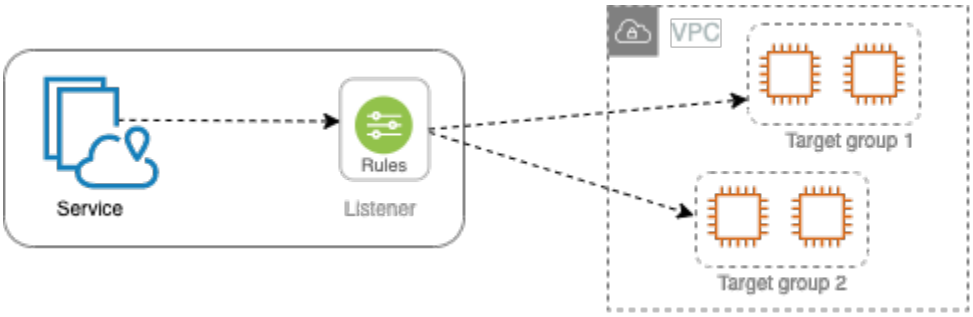
1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインの [VPC Lattice] で、[サービスネットワーク] を選択します。
3. サービスネットワークのチェックボックスをオンにし、[アクション]、[サービスネットワークを削除] の順に選択します。
4. 確認を求められたら、「**`confirm`**」と入力してから、[削除] を選択します。

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コマンドを実行します。

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。ターゲットには、EC2 インスタンス、IP アドレス、サーバーレス Lambda 関数、Application Load Balancer、または Kubernetes ポッドを指定できます。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)
- 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 (https://console.aws.amazon.com/vpc/) を開きます。
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 ~ 63 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. (オプション) サービスのカスタムドメイン名を指定するには、[カスタムドメイン設定を指定] を選択し、カスタムドメイン名を入力します。オプションで、一致する証明書をここで選択できます。それ以外の場合は、サービスの HTTPS リスナーを作成するときに、一致する証明書を選択できます。
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、 でロググループを作成する CloudWatch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) 他の アカウントとサービスを共有するには、 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 : ポート] では、[HTTP] または [HTTPS] 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。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 つだけ指定した場合、トラフィックの 100% 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 (https://console.aws.amazon.com/vpc/) を開きます。
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. [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承認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、粒度の粗い認証を実装できます。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 (https://console.aws.amazon.com/vpc/) を開きます。
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. [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コマンドを実行します。

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、ロググループ、Firehose 配信ストリーム、S3 バケットのリソースに CloudWatch 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 (https://console.aws.amazon.com/vpc/) を開きます。
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:

- 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、 でロググループを作成する CloudWatch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. [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コマンドを実行します。

を使用してログの送信先を更新するには AWS CLI

update-access-log-subscriptionコマンドを実行します。

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コマンドを実行します。

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。サービスに既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。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 (https://console.aws.amazon.com/vpc/) を開きます。
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「`service-name-service_id.partition_id.vpc-lattice-svcs.region.on.aws`」などのサービスの一意の完全修飾ドメイン名 (FQDN) を生成します。ただし、この VPC Lattice が生成したドメイン名はユーザーが覚えにくいものです。

カスタムドメイン名は、ユーザーに提供できる、よりシンプルで直感的な URL 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。ただし、これは CNAME レコードでカスタムドメイン名を VPC Lattice が生成したドメイン名にマッピングして、クエリをサービスにルーティングした場合に限ります。詳細については、「カスタムドメイン名をサービスに関連付ける」を参照してください。

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。

を使用してサービスのカスタムドメイン名を設定するには AWS Management Console

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
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。

を使用してサービスのカスタムドメイン名を設定するには AWS CLI

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。

AWS Certificate Manager (ACM) に独自の SSL/TLS 証明書がない場合は、カスタムドメイン名を設定する前に証明書を作成またはインポートできます。ただし、証明書はカスタムドメイン名を使用して HTTPS リクエストを処理する場合のみ必要です。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。

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ための CNAME 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、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、カスタムドメイン名 (例: `parking.example.com`) が VPC Lattice の自動生成ドメイン名 (例: `my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws`) にマッピングされるように CNAME レコードを作成します。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。詳細については、「Amazon Route 53 デベロッパーガイド」の「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。さらに、以下の手順を参照してホストゾーンと CNAME レコードを作成し、カスタムドメイン名を VPC Lattice エンドポイントにマッピングすることもできます。

Amazon Route 53 コンソールを使用して CNAME レコードで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には

1. Route 53 コンソール (https://console.aws.amazon.com/route53/) を開きます。
2. ナビゲーションペインで、[ホストゾーン] を選択し、次に [ホストゾーンの作成] を選択します。
3. [ドメイン名] には、VPC Lattice サービスへのトラフィックのルーティングに使用するホストゾーンの名前を選択します。例えば、カスタムドメイン名が `parking.example.com` (http://parking.example.com/) の場合、ホストゾーンのドメイン名は `example.com` (http://example.com/) となり、apex ドメイン名とも呼ばれます。次に、このホストゾーンの CNAME

レコードを作成し、トラフィックを VPC Lattice サービスにルーティングできます。注意: 作成後にホストゾーンの名前を変更することはできません。

4. [タイプ] には、必要に応じて [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] を選択します。
5. [リージョン] を選択し、このホストゾーンに関連付ける VPC の [VPC ID] を選択します。
6. 必要に応じてタグを追加し、[ホストゾーンの作成] を選択します。作成後、ホストゾーンは [ホストゾーン] に表示されます。
7. 作成したホストゾーンに CNAME レコードを作成するには、ホストゾーンを選択し、[レコードを作成] を選択します。
8. [レコードを作成] で次の値を指定します。

   a. [レコード名] には、カスタムドメイン名として使用する名前を入力します。`parking.example.com` (http://acme.example.com/) をカスタムドメイン名として使用する場合は、「`parking*`」と入力します。つまり、サブドメイン名 `parking` は入力しますが、ホストゾーンのドメイン名 `example.com` (http://example.com/) は入力しません。

   b. [レコードタイプ] には、[CNAME] を選択します。

   c. [エイリアス] はオフのままにしておきます。

   d. [値] には、VPC Latticeが生成したサービスのドメイン名を入力します (例: `my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws`)。この自動生成ドメイン名は、サービスページの VPC Lattice コンソールで確認できます。を使用する場合 AWS CLI、 `create-service`または `list-services` コマンドの出力はこの自動生成されたドメイン名を返します。

   e. [TTL (秒)] には、デフォルト値の 300 をそのまま使用します。

   f. [ルーティングポリシー] では、該当するルーティングポリシーを選択します。詳細については、「Amazon Route 53 デベロッパーガイド」の「ルーティングポリシーの選択」を参照してください。

9. [レコードを作成] を選択します。

通常、変更は 60 秒以内にすべての Route 53 サーバーに伝播されます。伝播が完了すると、カスタムドメイン名を使用してトラフィックをサービスにルーティングできるようになります。

を使用してホストゾーンにエイリアスレコードを作成するには AWS CLI

1. `get-service` コマンドを実行して、VPC Lattice が生成したサービスのドメイン名 (例: `my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws`) とホストゾーン ID を取得します。
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 hosted-zone-id-for-your-service-domain --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-service-domain.com alias",
    "Changes": [
        {
            "Action": "CREATE",
            "ResourceRecordSet": {
                "Name": "my-custom-domain-name.com",
                "Type": "alias-record-type",
                "AliasTarget": {
                    "HostedZoneId": "hosted-zone-id-for-your-service-domain",
                    "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 KMSKMS キーと呼ばれる を使用して別のキーを作成し、エイリアス aws/acm . AWS KMS uses このキー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「AWS Certificate Managerでのデータ保護」を参照してください。

VPC Lattice は、 にのみアクセスできるサービスである AWS TLS Connection Manager を使用して AWS のサービス、証明書のプライベートキーを保護して使用します。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 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 で許可を作成します。この権限により、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。

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 (https://console.aws.amazon.com/vpc/) を開きます。
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] を選択します。

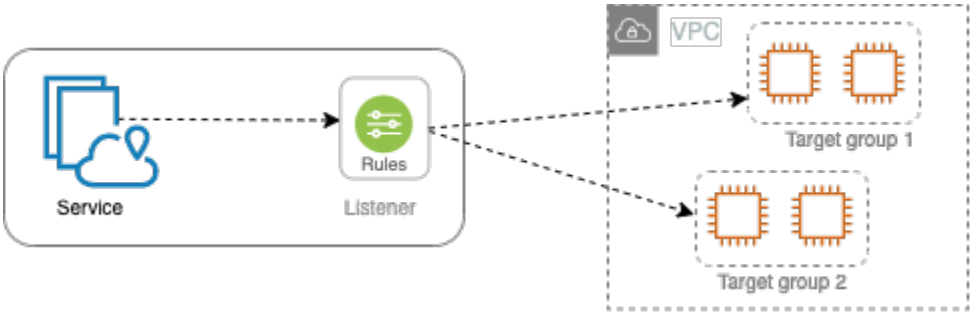
を使用してサービスを削除するには AWS CLI

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。ターゲットには、EC2 インスタンス、IP アドレス、Lambda 関数、Application Load Balancer、Kubernetes ポッドを指定できます。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。

コンテンツ

- VPC Lattice ターゲットグループを作成する
- 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
- ターゲットグループの削除

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 (https://console.aws.amazon.com/vpc/) を開きます。
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必要があります。また、最大 32 文字で、英数字またはハイフンのみを使用し、先頭と末尾にハイフンを使用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。

を使用してターゲットグループを作成するには AWS CLI

create-target-group コマンドを使用してターゲットグループを作成し、register-targets コマンドを使用してターゲットを追加します。

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 リソースを共有する」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。

コンテンツ

- 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
- IP アドレスによるターゲットの登録または登録解除
- 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 (https://console.aws.amazon.com/vpc/) を開きます。
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 (https://console.aws.amazon.com/vpc/) を開きます。
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 (https://console.aws.amazon.com/vpc/) を開きます。
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 (https://console.aws.amazon.com/vpc/) を開きます。
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正常性」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、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HealthCheckIntervalSeconds秒ごとに送信します。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連続した失敗を超えると、サービスはターゲットをサービス停止状態にします。ヘルスチェックがHealthyThresholdCount連続した成功を超えると、サービスはターゲットを実行中に戻します。

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

設定	説明
	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。
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 (https://console.aws.amazon.com/vpc/) を開きます。
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ヘルスステータスをチェックするには 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 Lambda 関数を開始し、異常なターゲットに関する詳細を送信します。

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 (https://console.aws.amazon.com/vpc/) を開きます。
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. [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コマンドを実行します。

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
- 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。

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 `ALB` である場合、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 タスクをターゲットとして登録するには、ALB ターゲットタイプを使用して Amazon ECS サービスの Application Load Balancer を登録します。詳細については、Amazon Elastic Container Service デベロッパーガイド の Service load balancing 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。

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、RFC 2616 の Message Headers 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-for-port`

送信先ポート

`x-forwarded-for-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-amz-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。
- `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-amz-lattice-network`

VPC。形式は次のとおりです。

```
source-vpc=arn:aws:ec2:region:account:vpc/id
```

`x-amz-lattice-target`

ターゲット。形式は次のとおりです。

```
service=arn;service-network=arn;target-group=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。

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。詳細については、「 AWS Lambda デベロッパーガイド」の「Amazon VPC Lattice AWS Lambda で を使用する」を参照してください。

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 Management Console 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「AWS Lambda 関数のバージョニングとエイリアス」および「エイリアスを使用したトラフィックの移行」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 (https://console.aws.amazon.com/vpc/) を開きます。
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。

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

create-target-group および 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 サービスは、 `Connection`や などの hop-by-hop ヘッダーを尊重しません`Transfer-Encoding`。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,
    "statusDescription": "200 OK",
    "headers": {
        "Set-cookie": "cookies",
        "Content-Type": "application/json"
    },
    "body": "Hello from Lambda (optional)"
```

```
}
```

複数値ヘッダー

デフォルトでは、VPC Lattice は、複数の値を含むヘッダーが含まれている、または同じヘッダーが複数回含まれているクライアントからのリクエストまたは Lambda 関数からのレスポンスをサポートします。また、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。

リクエストヘッダーの場合、複数のパラメータが同じ名前を共有していると、VPC Lattice は両方の値をターゲットに渡します。以下は、2 つの異なるヘッダーの名前が `header 1` になっている例です。

```
header1 = foo
header1 = bar
```

その後、VPC Lattice は両方の値をターゲットに送信します。

```
"header1": ["foo", "bar"]
```

クエリ文字列の場合、複数のパラメータが同じ名前を共有していると、最後の値が優先されます。つまり、パラメータが同じキー名を共有している場合、パラメータは 1 つの値に `_not_ coalesced` されます。

以下は、`foo` と `bar` が同じ名前 `QS1` のパラメータの値になっている場合の例です。

```
http://www.example.com?&QS1=foo&QS1=bar
```

その後、VPC Lattice は最後の値をターゲットに送信します。

```
"QS1": "bar"
```

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 (https://console.aws.amazon.com/vpc/) を開きます。
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 関数を登録解除するには 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 認証と承認などの 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 のユーザーガイド」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 のユーザーガイド」の「X-Forwarded-For」を参照してください。

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ません。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正常性」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 (https://console.aws.amazon.com/vpc/) を開きます。
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. [プロトコル] で **HTTP** または **HTTPS** を選択します。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。
- タグの名前または値に `aws:` プレフィックスは使用しないでください。このプレフィックスは AWS 用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 (https://console.aws.amazon.com/vpc/) を開きます。
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。

ターゲットグループの削除

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

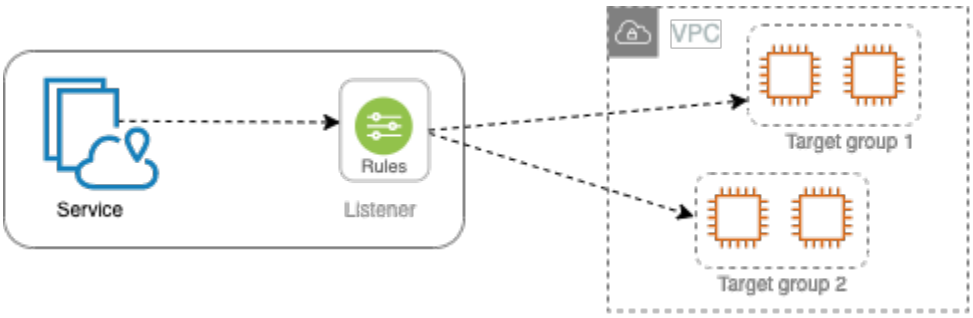
1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[ターゲットグループ] を選択します。
3. ターゲットグループのチェックボックスを選択し、[アクション]、[削除] を選択します。
4. 確認を求められたら、**confirm**と入力し、[削除] を選択します。

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コマンドを実行します。

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、1 つ以上の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。リスナーに対して定義したルールにより、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。



コンテンツ

- リスナーの設定
- リスナーの作成
- VPC Lattice サービスの HTTP リスナー
- VPC Lattice サービスの HTTPS リスナー
- VPC Lattice サービスのリスナールール
- リスナーの更新
- リスナーの削除

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
- 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。

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。

VPC ラティスはサポートしていません。 WebSockets

リスナーの作成

VPC Lattice サービスのリスナーを作成できます。リスナーを作成するときは、名前、デフォルトのアクション、プロトコルを指定する必要があります。リスナーにはデフォルトのルールが含まれています。リスナーのルールを追加で作成することもできます。

コンソールを使用してリスナーを作成する場合

- the section called “HTTP リスナーを追加する ”
- the section called “HTTPS リスナーの追加”
- the section called “ルールの追加”

を使用してリスナーを作成するには AWS CLI

create-listener コマンドと create-rule コマンドを使用します。

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。サービスの HTTPS リスナー作成については、「HTTPS リスナー」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 (https://console.aws.amazon.com/vpc/) を開きます。
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。

HTTPS リスナーを作成できます。このリスナーは TLS バージョン 1.2 を使用して VPC Lattice との HTTPS 接続を直接終了できます。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。

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。

このページの情報はサービスの HTTPS リスナー作成の際に参照してください。サービスの HTTP リスナー作成については、「HTTP 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
- ALPN ポリシー
- HTTPS リスナーの追加

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採用しています。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 はこの優先順位に基づき、TLSv1.2 プロトコルと次の SSL/TLS 暗号を使用します。

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

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 (https://console.aws.amazon.com/vpc/) を開きます。

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。

コンテンツ

- デフォルトのルール
- ルールの優先順位
- ルールアクション
- ルールの条件
- ルールの追加
- ルールを更新する
- ルールの削除

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 レスポンスコードを返せます。

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 (https://console.aws.amazon.com/vpc/) を開きます。
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。

   ヘッダーマッチまたはメソッドマッチ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. [変更の保存] を選択します。

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 (https://console.aws.amazon.com/vpc/) を開きます。
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 (https://console.aws.amazon.com/vpc/) を開きます。
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. [変更の保存] を選択します。

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。

リスナーの更新

リスナーを作成後に、デフォルトアクションのターゲットグループを変更できます。また、ターゲットグループをデフォルトアクションに追加し、ターゲットグループに重み付けを設定できます。リスナー名、リスナープロトコル、リスナーポートは更新できません。

コンソールを使用してリスナーを更新する場合

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
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. [デフォルトアクション] では、必要に応じてターゲットグループまたは重み付けを更新できます。
6. ターゲットグループを追加するには、[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。
7. リスナールールも追加、編集、削除ができます。詳細については、「リスナールール」を参照してください。
8. 更新内容を確認して [変更を保存] をクリックします。

を使用してリスナーのデフォルトアクションを更新するには AWS CLI

update-listener コマンドを使用します。

リスナーの削除

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 (https://console.aws.amazon.com/vpc/) を開きます。
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。

VPC Lattice リソースを共有する

Amazon VPC Lattice は AWS Resource Access Manager (AWS RAM) と統合してリソース共有を可能にします。AWS RAM は、VPC Lattice リソースの一部を他の AWS アカウント と、または AWS Organizations を通じて共有できるようにするサービスです。AWS RAM を使用したリソース共有。これにより、自身が所有するリソースを共有できます。リソース共有は、共有するリソースと、それらを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。

- AWS Organizations の組織内外の特定の AWS アカウント。
- AWS Organizations の組織内の組織単位。
- AWS Organizations の組織全体。

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

目次

- VPC Lattice リソースを共有するための前提条件
- VPC Lattice リソースを共有する
- VPC Lattice リソースの共有を停止する
- 責任と権限
- クロスアカウントイベント

VPC Lattice リソースを共有するための前提条件

- リソースを共有するには、AWS アカウント でそのリソースを所有する必要があります。つまり、自分のアカウントにそのリソースが割り当てられているか、プロビジョニングされている必要があります。自分が共有先になっているリソースを共有することはできません。
- 自分の組織または AWS Organizations の組織単位とリソースを共有するには、AWS Organizations との共有を有効にする必要があります。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 リソースを共有する

リソースを共有するには、AWS Resource Access Manager を使用してリソース共有を作成することから始めます。リソース共有では、共有対象のリソース、リソースを共有するコンシューマー、プリンシパルが実行できるアクションを指定します。

自分が所有する VPC Lattice リソースを他の AWS アカウント と共有すると、それらのアカウントのリソースを自分のアカウントのリソースと関連付けられるようになります。共有リソースに対して関連付けを作成すると、リソース所有者アカウントに Amazon リソースネーム (ARN) が生成され、さらに関連付けを作成したアカウントにも ARN が生成されます。これにより、リソース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。

自分が AWS Organizations の組織に属していて、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リソースに対するアクセス許可が自動的に付与されます。これに該当しない場合、コンシューマーはリソースへの参加の招待を受け取り、その招待を受け入れた後で、共有リソースに対するアクセス許可が付与されます。

考慮事項

- サービスネットワークとサービスの 2 種類の VPC Lattice リソースを共有できます。
- VPC Lattice リソースは任意の AWS アカウント と共有できます。
- VPC Lattice リソースを個々の IAM ユーザーやロールと共有することはできません。
- VPC Lattice は、サービスネットワークとサービスの両方について、お客様が管理する権限をサポートします。

VPC Lattice コンソールを使用して自分が所有するリソースを共有するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] または [サービスネットワーク] を選択します。
3. リソースの名前を選択してその詳細ページを開き、[共有] タブから [サービスを共有] または [サービスネットワークを共有] を選択します。
4. [リソース共有] から AWS RAM 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. [サービスを共有] または [サービスネットワークを共有] を選択します。

AWS RAM コンソールを使用して自分が所有するリソースを共有するには

「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。

AWS CLI を使用して自分が所有するリソースを共有するには

associate-resource-share コマンドを使用します。

VPC Lattice リソースの共有を停止する

自分が所有する VPC Lattice リソース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。リソースの共有を停止しても、既存の関連付けは維持されます。以前に共有していたリソースへの新しい関連付けは許可されません。リソース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、その関連付けは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終了する場合は、リソース共有の所有者に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自分が所有するリソースの共有を停止するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインの [VPC Lattice] で、[サービス] または [サービスネットワーク] を選択します。
3. リソースの名前を選択して、その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自分が所有するリソースの共有を停止するには

「AWS RAM ユーザーガイド」の「リソース共有の更新」を参照してください。

AWS CLI を使用して自分が所有するリソースの共有を停止するには

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

責任と権限

共有 VPC Lattice リソースを使用する場合、次の責任と権限が適用されます。

リソース所有者

- 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。
- サービス所有者は、サービスとのすべての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。
- 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。

リソースコンシューマー

- コンシューマーは、自分が作成していないサービスを削除できません。
- 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サービスのみを解除できます。
- 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、自分が所有していないサービスのサービス情報を取得できません。
- 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サービスの関連付けを記述できます。
- コンシューマーは、サービス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。
- 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。
- コンシューマーは、共有されているサービスまたはサービスネットワークを記述できます。

- コンシューマーは、2 つのリソースを共有している場合、両方のリソース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

リソース所有者とコンシューマーが共有リソース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 AWS CloudTrail のクロスアカウントイベントとして記録されます。

`CreateServiceNetworkServiceAssociationBySharee`

リソースコンシューマーが共有リソースで CreateServiceNetworkServiceAssociation を呼び出すと、リソース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`

リソース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、リソース所有者に送信されます。

`DeleteServiceNetworkServiceAssociationByOwner`

リソース所有者が共有リソースで 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`

リソースコンシューマーが共有リソースで DeleteServiceNetworkServiceAssociation を呼び出すと、リソース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`

リソース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`

リソース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、リソース所有者に送信されます。

`GetServiceBySharee`

リソースコンシューマーが共有サービスで GetService を呼び出すと、リソース所有者に送信されます。

`GetServiceNetworkBySharee`

リソースコンシューマーが共有サービスネットワークで GetServiceNetwork を呼び出すと、リソース所有者に送信されます。

`GetServiceNetworkServiceAssociationBySharee`

リソースコンシューマーが共有リソースで GetServiceNetworkServiceAssociation を呼び出すと、リソース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`

リソース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、リソース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"
}
```

Amazon VPC Lattice におけるセキュリティ

AWS クラウドセキュリティは最優先事項です。 AWS お客様は、最もセキュリティに敏感な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の恩恵を受けることができます。

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。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して説明しています。

- クラウドのセキュリティ — AWS AWS で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があります。 AWS クラウド AWS また、安全に使用できるサービスも提供します。第三者監査人は、AWS、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しています。Amazon VPC Lattice に適用されるコンプライアンスプログラムについては、「AWS コンプライアンスプログラム別の対象サービス」「」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、お客様は、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 リソースの監視と保護に役立つ他のサービスの使用方法についても学びます。

コンテンツ

- サービスへのアクセスを管理する
- Amazon VPC Lattice でのデータ保護
- Amazon VPC Lattice のアイデンティティとアクセス管理
- Amazon VPC Lattice のコンプライアンス検証
- インターフェースエンドポイント () を使用して VPC Lattice にアクセスする PrivateLink
- Amazon VPC Lattice の耐障害性
- Amazon VPC Lattice のインフラストラクチャセキュリティ

サービスへのアクセスを管理する

VPC Lattice ではどのサービスにどの VPC でのアクセスを付与するかを明示する必要があるため、デフォルトで安全性が確保されます。マルチアカウントのシナリオでは、AWS Resource Access Manager を使ってアカウントの境界を越えてリソースを共有できます。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 – サービスと VPC のサービスネットワークとの関連付け。VPC または特定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ない場合、VPC 内のクライアントはサービス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により、VPC 内のすべてのリソースではなく、VPC 内の特定のリソース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証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、粗粒度の認証が実装されます。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証ポリシーでは、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証よりも厳しい場合があります。

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。

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。認証

が正常に完了するためには、認証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、「認証の仕組み」を参照してください。

AWS CLI およびコンソールを使用して、サービスとサービスネットワークの認証ポリシーを表示、追加、更新、削除できます。を使用する場合 AWS CLI、 AWS リージョン コマンドはプロファイルに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。

コンテンツ

- 認証ポリシーの一般的な要素
- 認証ポリシーのリソース形式
- 認証ポリシーで使用できる条件キー
- 匿名 (認証されていない) プリンシパル
- 認証ポリシーの例
- 認証の仕組み

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

AWS CLI 以下のタスクは、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ています。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

このセクションの手順に従って、次の操作を行います AWS CLI 。

- 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コマンドを実行します。

```
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

AWS CLI 以下のタスクは、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ています。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

次の手順に従って、 AWS CLI を使用して次の操作を行います。

- 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コマンドを実行します。

```
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、「署名バージョン 4 で認証されたリクエストの例」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、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、 AWS または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。明示的な `Allow` はデフォルトに優先します。
- アクション – VPC Lattice は 1 つのアクション `vpc-lattice-svcs:Invoke` をサポートします。このアクセス許可により、指定されたプリンシパルは `Resources` 要素で指定されたリソースに対してリクエストを実行できます。
- 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。これを使用してポリシーを有効にするタイミングをコントロールします。

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。

認証ポリシーのリソース例

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"`

プロトコル	例
	• "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:

```
"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。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。

要件

条件キーは大文字と小文字が区別されます。

認証ポリシーの条件キー

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	Yes
`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	Yes	常に POST
`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	Yes
`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]	Yes	No
`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	Yes
`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-2:123456	はい	Yes

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
		789012:service/svc-0123456789abcdef0		
`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	Yes
`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	Yes

AWS また、`aws:PrincipalOrgID`グローバル条件キーなど、アクセスを制御するために使用できる追加の条件キーも用意されています。 AWS すべての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。

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) AWS で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 内にいる呼び出し元です。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: 特定の組織によるサービスへのアクセスを制限する AWS

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、 AWS リクエストは条件で指定された組織に所属するプリンシパルからのものでなければなりません。

```
{
   "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。

```
{
   "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。

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

認証の仕組み

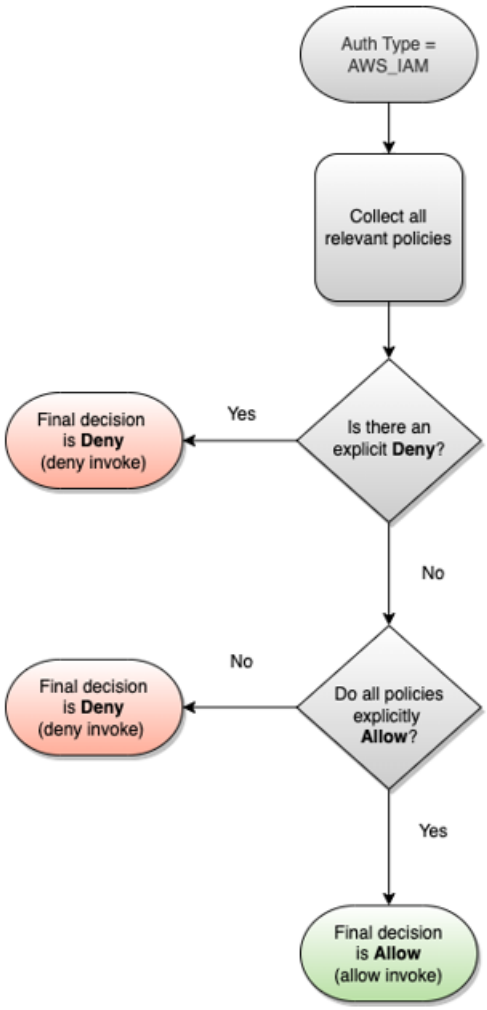
VPC Lattice サービスがリクエストを受け取ると、 AWS エンフォースメントコードは関連するすべての権限ポリシーをまとめて評価し、リクエストを承認するか拒否するかを決定します。リクエストコンテキストに適用されるすべての IAM アイデンティティベースのポリシーと認証ポリシーを認証

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。

認証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 リクエストは承認されません。
  - 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になっていても、明示的な許可ステートメントがない場合は、 AWS リクエストを承認しません。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になっていても、明示的な許可ステートメントがない場合、 AWS リクエストは承認されません。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。

次の図は認証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、関連するリソースに出入りするネットワークトラフィックを制御します。VPC Lattice を使うと、セキュリティグループを作成して、VPC をサービスネットワークにリンクする VPC の関連付けにセキュリティグループを割り当て、サービスネットワークにおいてネットワークレベルのセキュリティ保護を強化できます。

コンテンツ

- 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 は VPC Lattice ネットワーク上のトラフィックのルーティングに使用される IP アドレスを含む、マネージドプレフィックスリストを提供します。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の提供」を参照してください。

Windows 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です。Windows クライアントから VPC Lattice に接続する場合は、Windows クライアントの設定を更新して、VPC Lattice が使用するリンクローカル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。この例では 169.254.171.0 が VPC Lattice が使用するリンクローカルアドレスとなります。

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

「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む 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。

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 VPC からサービスネットワークに関連するサービスにトラフィックが流れるようにするには、サービスのリスナーポートとリスナープロトコルに関するインバウンドルールを作成する必要があります。

インバウンド

送信元	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す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。

アウトバウンド

送信先	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。

インバウンド

送信元	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 上のセキュリティグループとサービスネットワークの関連付けを表示、追加、または更新できます。 AWS CLI を使用する場合 AWS CLI、 AWS リージョン コマンドはプロファイルに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 (https://console.aws.amazon.com/vpc/) を開きます。
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. [変更の保存] を選択します。

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 (https://console.aws.amazon.com/vpc/) を開きます。
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. [変更の保存] を選択します。

を使用して VPC アソシエーション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 アソシエーションの 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 アソシエーション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。サブネット用のカスタムネットワーク ACL を作成して、セキュリティをさらに強化できます。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。

コンテンツ

- クライアントサブネット用のネットワーク ACL
- ターゲットサブネットのネットワーク ACL

クライアントサブネット用のネットワーク ACL

クライアントサブネットのネットワーク ACL 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 の管理対象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

送信元	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

アウトバウンド

送信先	プロトコル	ポート範囲	コメント
vpc_Lattice_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 ACL

ターゲットサブネットのネットワーク ACL 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 の管理対象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

送信元	プロトコル	ポート範囲	コメント
vpc_lattice_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_lattice_cidr_block	*health check*	*health check*	VPC Lattice からターゲットへのヘルス

送信元	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

アウトバウンド

送信先	プロトコル	ポート範囲	コメント
vpc_lattice_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_Lattice_cidr_block	*health check*	1024-65535	ターゲットから VPC Lattice へのヘルスチェックトラフィックを許可する

署名バージョン 4 で認証されたリクエストの例

VPC ラティスはクライアント認証に署名バージョン 4 (SigV4) または署名バージョン 4A (SigV4A) を使用します。詳細については、IAM ユーザーガイドの「AWS API リクエストへの署名」を参照してください。

考慮事項

- VPC ラティスは SigV4 または SigV4A で署名されたすべてのリクエストを認証しようとします。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 で Intercepter を使用する場合
- Java で Intercepter を使用しない場合
- Node.js

Python

この例では、署名されたリクエストを安全な接続を介してネットワークに登録されている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。

botocoreawscrtおよびパッケージをインストールするには、以下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
```

次の例では、プレースホルダーの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
from botocore.credentials import Credentials
import botocore.session

if __name__ == '__main__':
    session = botocore.session.Session()
    signer = crt.auth.CrtS3SigV4Auth(session.get_credentials(), 'vpc-lattice-svcs',
 'us-west-2')
    endpoint = 'https://user-02222f67d3a427111.1234abc.vpc-lattice-svcs.us-
west-2.on.aws/create'
    data = "some-data-here"
    headers = {'Content-Type': 'application/json'}
    request = AWSRequest(method='POST', url=endpoint, data=data, headers=headers)
    request.context["payload_signing_enabled"] = False # payload signing is not
 supported
    signer.add_auth(request)

    prepped = request.prepare()

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

SIGv4A

```
from botocore import crt
```

```
import requests
from botocore.awsrequest import AWSRequest
from botocore.credentials import Credentials
import botocore.session

if __name__ == '__main__':
    session = botocore.session.Session()
    signer = crt.auth.CrtS3SigV4AsymAuth(session.get_credentials(), 'vpc-lattice-
svcs', 'us-west-2')
    endpoint = 'https://user-02222f67d3a427111.1234abc.vpc-lattice-svcs.us-
west-2.on.aws/create'
    data = "some-data-here"
    headers = {'Content-Type': 'application/json'}
    request = AWSRequest(method='POST', url=endpoint, data=data, headers=headers)
    request.context["payload_signing_enabled"] = False # payload signing is not
 supported
    signer.add_auth(request)

    prepped = request.prepare()

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

Java で Intercepter を使用する場合

この例では Amazon Request Signing Interceptor を使って、リクエスト署名を処理しています。

```
import com.amazonaws.http.AwsRequestSigningApacheInterceptor;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.auth.signer.Aws4UnsignedPayloadSigner;
import software.amazon.awssdk.regions.Region;

import java.nio.charset.StandardCharsets;

import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.ByteArrayEntity;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;

public class App {
    public static void main(String[] args) {
      var interceptor = new AwsRequestSigningApacheInterceptor(
          "vpc-lattice-svcs",
```

```
            Aws4UnsignedPayloadSigner.create(), // requires HTTPS
            DefaultCredentialsProvider.create(),
            Region.US_WEST_2.id()
            );
        CloseableHttpClient client = HttpClients.custom()
          .addInterceptorLast(interceptor)
          .build();

        var httpPost = new HttpPost("https://user-02222f67d3a427111.1234abc.vpc-lattice-
svcs.us-west-2.on.aws/create");
        httpPost.addHeader("content-type", "application/json");

        var body = """
        {
          "name": "Jane Doe",
          "job": "Engineer"
        }
        """;
        httpPost.setEntity(new ByteArrayEntity(body.getBytes(StandardCharsets.UTF_8)));

        try (var response = client.execute(httpPost)) {
          System.out.println(new
 String(response.getEntity().getContent().readAllBytes()));
        } catch (Exception e) {
          throw new RuntimeException(e);
        }
      }
}
```

Java で Intercepter を使用しない場合

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。は HTTPS AWS SDK for Java 経由の署名なしペイロードのみを許可します。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。

```
import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.auth.signer.Aws4UnsignedPayloadSigner;
import software.amazon.awssdk.auth.signer.AwsSignerExecutionAttribute;
import software.amazon.awssdk.core.interceptor.ExecutionAttributes;
import software.amazon.awssdk.http.SdkHttpFullRequest;
import software.amazon.awssdk.http.SdkHttpMethod;
import software.amazon.awssdk.regions.Region;

import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.ByteArrayEntity;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;

public class App {

    public static void main(String[] args) {
        var signer = Aws4UnsignedPayloadSigner.create(); // requires HTTPS

        Map<String, String> headers = new HashMap<>();
        headers.put("content-type", "application/json");
        var body = """
        {
            "name": "Jane Doe",
            "job": "Engineer"
        }
        """;

        String endpoint = "https://user-02222f67d3a427111.1234abc.vpc-lattice-svcs.us-
west-2.on.aws/create";

        var sdkRequest = SdkHttpFullRequest.builder().method(SdkHttpMethod.POST);

        sdkRequest.host("user-02222f67d3a427111.1234abc.vpc-lattice-svcs.us-
west-2.on.aws");
        sdkRequest.protocol("HTTPS");
        sdkRequest.encodedPath("/create");
        sdkRequest.contentStreamProvider(() -> new
 ByteArrayInputStream(body.getBytes(StandardCharsets.UTF_8)));


        for (Map.Entry<String, String> header : headers.entrySet()) {
            sdkRequest.putHeader(header.getKey(), header.getValue());
        }
```

```
        ExecutionAttributes attributes = ExecutionAttributes.builder()
                .put(AwsSignerExecutionAttribute.AWS_CREDENTIALS,
 DefaultCredentialsProvider.create().resolveCredentials())
                .put(AwsSignerExecutionAttribute.SERVICE_SIGNING_NAME, "vpc-lattice-
svcs")
                .put(AwsSignerExecutionAttribute.SIGNING_REGION, Region.US_WEST_2)
                .build();

        SdkHttpFullRequest prepRequest = signer.sign(sdkRequest.build(), attributes);

        HttpPost httpPost = new HttpPost(endpoint);
        for (Map.Entry<String, List<String>> header : prepRequest.headers().entrySet())
 {
            if (header.getKey().equalsIgnoreCase("host")) { continue; }
            for(var value : header.getValue()) {
                httpPost.addHeader(header.getKey(), value);
            }
        }

        CloseableHttpClient client = HttpClients.custom().build();

        httpPost.setEntity(new ByteArrayEntity(body.getBytes(StandardCharsets.UTF_8)));

        try (var response = client.execute(httpPost)){
            System.out.println(new
 String(response.getEntity().getContent().readAllBytes()));
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
```

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').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: '/',
      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').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: '/',
      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()
  }
)
```

Amazon VPC Lattice でのデータ保護

AWS https://aws.amazon.com/compliance/shared-responsibility-model/、Amazon VPC Lattice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、すべてを稼働させ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 のセキュリティ構成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は `CreateService` や `UpdateService` などのリソースの作成、読み取り/表示、更新、削除、リスト (CRUDL) に使用する管理 API を提供します。VPC Lattice のコントロールプレーンへの通信は転送時に 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可能にする VPC Lattice の呼び出し API です。TLS は VPC Lattice のデータプレーンへの通信も暗号化します。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。

コンテンツ

- 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) AWS KMS に保存されている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。追加の保護措置として、定期的にローテーションされるルートキーを使ってキーそのものを暗号化します。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 アクセスログ用の S3 バケットの Amazon S3 マネージド暗号化キー (SSE-S3) によるサーバー側の暗号化を有効にすると、各アクセスログファイルは S3 AWS バケットに保存される前に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 に送信されるログ」を参照してください。

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

AWS KMS 鍵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 キーの使用には、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権限があります。また、SSE-KMS では 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入手できます。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。

コンテンツ

- 証明書のプライベートキーの暗号化と復号化
- VPC Lattice の暗号化コンテキスト
- VPC Lattice の暗号化キーをモニタリングする

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 aws/acm AWS というエイリアスのマネージド KMS キーで暗号化されます。このエイリアスの付いたキー ID は、コンソールの管理対象キーの下に表示されます。AWS KMS AWS

VPC Lattice は ACM リソースに直接アクセスしません。 AWS TLS 接続マネージャーを使用して証明書の秘密鍵を保護し、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。そのためには、aws/acm AWS KMS AWS というプレフィックスを付けたマネージドキーに対するグラントを作成します。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、またはを使用するコマンドを使用して、サービスの証明書を交換または削除することをお勧めします。 AWS Management Console `update-service` AWS CLI

VPC Lattice の暗号化コンテキスト

暗号化コンテキストは、秘密鍵の使用目的に関する追加の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。 AWS KMS 暗号化コンテキストを暗号化データにバインドし、認証済み暗号化をサポートする追加の認証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どの VPC Lattice サービスに使用されているかは、 CloudTrail 次のセクションに示すように口グの暗号化コンテキストを確認するか、ACM コンソールの [関連リソース] タブで確認できます。

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

>> イベント履歴>> `CreateGrant` CloudTrail で操作をイベントとして表示できます。**`CreateGrant`**

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

```
{
    "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` の例では、被付与者のプリンシパルは TLS 接続マネージャで、暗号化コンテキストに VPC Lattice サービス ARN が含まれていることがわかります。

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、を参照してください[ListGrants](#)。

`ListGrants`で次のコマンドを実行すると AWS CLI 、すべての権限の詳細が表示されます。

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

出力は次の例のようになります。

```
{
    "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` の例では、被付与者のプリンシパルは TLS 接続マネージャで、暗号化コンテキストに VPC Lattice サービス ARN が含まれていることがわかります。

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。>> イベント履歴 CloudTrail >> **`Decrypt`** で操作をイベントとして表示できます`Decrypt`。

以下は、 CloudTrail `Decrypt`オペレーションのイベント履歴にある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、 AWS Identity and Access Management (IAM) を使用して 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 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
- VPC Lattice API 許可

- Amazon VPC Lattice のアイデンティティベースのポリシー
- VPC Lattice のサービスにリンクされたロールを使用する
- AWS VPC ラティスの管理ポリシー

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。

Amazon VPC Lattice で使用できる IAM の機能

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	Yes
リソースベースのポリシー	Yes
ポリシーアクション	Yes
ポリシーリソース	はい
ポリシー条件キー	Yes
ACL	No
ABAC (ポリシー内のタグ)	はい
一時的な認証情報	Yes
サービスロール	いいえ
サービスリンクロール	Yes

VPC Lattice AWS やその他のサービスがほとんどの IAM 機能でどのように機能するかについての概要については、IAM ユーザーガイドの「IAM AWS と連携するサービス」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ポリシーをサポートする	Yes

アイデンティティベースポリシーは、IAM ユーザー、ユーザーのグループ、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 ポリシーの作成」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。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。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	Yes

リソースベースのポリシーは、リソースにアタッチする JSON ポリシードキュメントです。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーが添付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は、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。このようなリソースベースのポリシーを使用すれば、他の AWS アカウントまたは組織に使用許可を付与して、リソースの共有ができます。詳細については、「VPC Lattice リソースを共有する」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に対するサポート	Yes

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

ポリシーリソースに対するサポート	Yes

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"
```

- 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	はい

IAM ポリシーで VPC Lattice リソースへのアクセスをコントロールする条件を指定できます。ポリシーステートメントは、条件が true の場合にのみ有効です。

VPC Lattice では、VPC Lattice API アクションを実行できるユーザーを決定するためにアイデンティティベースのポリシーで使用できる次のサービス定義条件キー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 の条件キー」を参照してください。

アイデンティティベースのポリシーのサービス定義条件キー

条件キー	説明	サポートされているアクション
`vpc-lattice:AuthType`	リクエストの認証タイプ (`AWS_IAM` または `NONE`) によりアクセスをフィルタリング	• CreateService • CreateServiceNetwork • UpdateService • UpdateServiceNetwork
`vpc-lattice:Protocol`	リクエストのプロトコル (`HTTP` または `HTTPS`) によりアクセスをフィルタリング	• CreateListener
`vpc-lattice:SecurityGroupIds`	リクエストのセキュリティグループ ID によりアクセスをフィルタリング	• CreateServiceNetworkVpcAssociation • UpdateServiceNetworkVpcAssociation
`vpc-lattice:ServiceArn`	リクエストのサービスの ARN によりアクセスをフィルタリング	• CreateServiceNetworkServiceAssociation • DeleteServiceNetworkServiceAssociation • GetServiceNetworkServiceAssociation • ListServiceNetworkServiceAssociations
`vpc-lattice:ServiceNetworkArn`	リクエスト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	• CreateServiceNetworkServiceAssociation • CreateServiceNetworkVpcAssociation • DeleteServiceNetworkVpcAssociation • GetServiceNetworkServiceAssociation

条件キー	説明	サポートされているアクション
		• GetServiceNetworkVpcAssociation • ListServiceNetworkServiceAssociations • ListServiceNetworkVpcAssociations • UpdateServiceNetworkVpcAssociation
`vpc-lattice:TargetGroupArns`	リクエストのターゲットグループの ARN によりアクセスをフィルタリング	• CreateListener • CreateRule • UpdateListener • UpdateRule
`vpc-lattice:VpcId`	リクエストの仮想プライベートクラウド (VPC) の ID によりアクセスをフィルタリング	• CreateServiceNetworkVpcAssociation • CreateTargetGroup • DeleteServiceNetworkVpcAssociation • GetServiceNetworkVpcAssociation • ListServiceNetworkVpcAssociations • UpdateServiceNetworkVpcAssociation

AWS 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ます。 AWS グローバル条件キーについては、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	No

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許可を持つかをコントロールします。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 のサポート (ポリシー内のタグ)	はい

属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義するアクセス許可戦略です。では AWS、これらの属性はタグと呼ばれます。IAM エンティティ (ユーザーまたはロール) AWS や多くのリソースにタグを付けることができます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。次に、プリンシパルのタグがアクセスを試行するリソースのタグと一致したときにオペレーションを許可するよう、ABAC ポリシーを設計します。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

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 Yes です。サービスが一部のリソースタイプに対してのみ 3 つの条件キーすべてをサポートする場合、値は Partial です。

ABAC の詳細については、IAM ユーザーガイドの「ABAC とは? 」を参照してください。ABAC をセットアップするステップ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「属性に基づく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	Yes

AWS のサービス 一時的な認証情報を使用してサインインすると機能しないものもあります。 AWS のサービス 一時的な認証情報で機能するものなど、追加情報については、『IAM ユーザーガイド』の「IAM と連携する」を参照してくださいAWS のサービス 。

ユーザー名とパスワード以外の方法でにサインインすると、 AWS Management Console 一時的な認証情報が使用されることになります。たとえば、会社のシングルサインオン (SSO) AWS リンクを使用してアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自

動的に作成されます。ロールの切り替えに関する詳細については、 IAM ユーザーガイド の「ロールへの切り替え (コンソール)」を参照してください。

または API を使用して一時的な認証情報を手動で作成できます。 AWS CLI AWS その後、その一時的な認証情報を使用してアクセスできます AWS。 AWS 長期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することをおすすめします。詳細については、「IAM の一時的セキュリティ認証情報」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	いいえ

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 のサービスにアクセス許可を委任するロールの作成」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	Yes

サービスにリンクされたロールは、にリンクされている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。 AWS アカウント サービスにリンクされたロールはに表示され、そのサービスが所有します。IAM 管理者は、サービスリンクロールの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、「VPC Lattice のサービスにリンクされたロールを使用する」を参照してください。

VPC Lattice API 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、一部の VPC Lattice アクションでは、他の API から特定のアクションを呼び出すための権限を IAM ID に付与する必要があります。 AWS

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。

`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。また、、 AWS Command Line Interface (AWS CLI) AWS Management Console、API を使用してタスクを実行することもできません。 AWS 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。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き受けることができます。

これらサンプルの 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 ポリシーの作成」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。

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。

トピック

- ポリシーのベストプラクティス
- フルアクセスに必要な追加のアクセス許可
- VPC Lattice のアイデンティティベースのポリシーの例

ポリシーの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を実行すると、 AWS アカウントに料金が発生する可能性があります。アイデンティティベースの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 管理ポリシーから始めて、最小権限の権限に移行する — ユーザーとワークロードへのアクセス権限の付与を開始するには、AWS 多くの一般的なユースケースで権限を付与する管理ポリシーを使用してください。これらのポリシーは、で利用できます。 AWS アカウント AWS 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、権限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「AWS ジョブ機能の管理ポリシー」を参照してください。
- 最小特権を適用する – IAM ポリシーで許可を設定するときは、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。例えば、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。サービスアクションがなどの特定の用途で使用された場合は AWS のサービス、条件を使用して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 AWS CloudFormation。詳細については、「IAM ユーザーガイド」の [IAM JSON policy elements: Condition] (IAM JSON ポリシー要素：条件) を参照してください。
- IAM Access Analyzer 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 Access Analyzer 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 ポリシーの検証」を参照してください。
- 多要素認証 (MFA) が必要 — IAM ユーザーまたは root 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、セキュリティを強化するために MFA をオンにしてください。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 保護 API アクセスの設定」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 AWS が統合されている他のサービスや VPC Lattice の全機能を使用するには、特定の追加権限が必要です。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。

```
{
    "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

- サービスネットワークとの VPC の関連付けを管理する
- サービスネットワークとのサービスの関連付けを作成する
- リソースにタグを追加
- サービスにリンクされたロールの作成

サービスネットワークとの 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。

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

サービスネットワークとのサービスの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。

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

リソースにタグを追加

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。

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "vpc-lattice:TagResource"
         ],
         "Resource": "arn:aws:vpc-lattice:us-west-2:123456789012:*/*"
      }
   ]
}
```

サービスにリンクされたロールの作成

VPC Lattice では、社内のユーザーが VPC Lattice リソースを初めて作成するときに、 AWS アカウント サービスにリンクされたロールを作成する権限が必要です。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 にアクセス権限を付与し、 AWS のサービス ユーザーに代わって他のユーザーを呼び出せるようにします。

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。

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

VPC Lattice のサービスにリンクされたロールを使用する

Amazon VPC Lattice は、 AWS のサービス ユーザーに代わって他のユーザーを呼び出すために必要なアクセス権限として、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「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。

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

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

AWSServiceRoleForVpcLatticeサービスにリンクされたロールは、以下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権限ポリシーにより、VPC Lattice CloudWatch は名前空間にメトリクスを公開できます。`AWS/VpcLattice`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricData",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "AWS/VpcLattice"
                }
            }
        }
    ]
}
```

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、許可を設定する必要があります。詳細については、「IAM User Guide」(IAM ユーザーガイド) の「Service-linked role permissions」(サービスリンクロールのアクセス権限) 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 Management Console、、または AWS API で VPC Lattice リソースを作成すると AWS CLI、VPC Lattice が自動的にサービスにリンクされたロールを作成します。

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 の使用に関する説明は編集できます。AWSServiceRoleForVpcLattice詳細については、「IAM ユーザーガイド」の「サービスリンクロールの編集」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、削除することをお勧めします。AWSServiceRoleForVpcLattice

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、、または AWS API を使用して AWS CLI、サービスにリンクされたロールを削除します。AWSServiceRoleForVpcLattice詳細については、IAM ユーザーガイドの「サービスリンクロールの削除」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 VPC ラティスの管理ポリシー

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のポリシーです。 AWS AWS 管理ポリシーは、ユーザー、グループ、ロールにアクセス権限を割り当てることができるように、多くの一般的な使用事例にアクセス許可を与えるように設計されています。

AWS 管理ポリシーでは、 AWS すべての顧客が使用できるようになっているため、特定のユースケースでは最小権限のアクセス権限が付与され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 カスタマーマネージドポリシーを定義して、許可をさらに減らすことをお勧めします。

AWS 管理ポリシーで定義されている権限は変更できません。 AWS 管理ポリシーで定義されている権限を更新すると AWS 、その更新は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 AWS 管理ポリシーが更新される可能性が最も高いのは、新しい API 操作が既存のサービスで開始されたときや、新しい API AWS のサービス 操作が使用可能になったときです。

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 管理ポリシー:VPC LatticeFullAccess

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 ログ — アクセスログを設定して CloudWatch Logs に送信します。
- 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 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。

このポリシーの権限を確認するには、『AWS 管理ポリシーリファレンス』LatticeFullAccessの「VPC」を参照してください。

VPC Lattice AWS が統合されている他のサービスや VPC Lattice の全機能を使用するには、特定の追加権限が必要です。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 管理ポリシー:VPC LatticeReadOnlyAccess

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 ログ — 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 — アクセスログ配信用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 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。

このポリシーの権限を確認するには、『AWS 管理ポリシーリファレンス』LatticeReadOnlyAccessの「VPC」を参照してください。

AWS 管理ポリシー:VPC LatticeServicesInvokeAccess

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。

このポリシーの権限を確認するには、『AWS 管理ポリシーリファレンス』LatticeServicesInvokeAccessの「VPC」を参照してください。

AWS 管理ポリシー: AWSVpcLatticeServiceRolePolicy

このポリシーは、VPC Lattice AWSServiceRoleForVpcLatticeがユーザーに代わってアクションを実行できるようにするという、サービスにリンクされたロールにアタッチされます。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、「VPC Lattice のサービスにリンクされたロールを使用する」を参照してください。

このポリシーの権限を確認するには、『管理ポリシーリファレンス』AWSVpcLatticeServiceRolePolicyAWS のを参照してください。

VPC Lattice AWS による管理ポリシーの更新

このサービスが変更の追跡を開始して以降の VPC Lattice AWS の管理ポリシーの更新に関する詳細をご覧ください。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
VPC LatticeFullAccess	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
VPC LatticeReadOnlyAccess	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
VPC LatticeServicesInvokeAccess	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 はサービスにリンクされたロールにアクセス権限を追加して、VPC Lattice が名前空間にメトリクスを公開できるようにします。 CloudWatch `AWS/VpcLattice` `AWSVpcLatticeServiceRolePolicy` このポリシーには API アクションを呼び出す権限が含まれています。 CloudWatch PutMetricData詳細については、「VPC Lattice のサービスにリンクされたロールを使用する」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 AWS は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

第三者監査人は、 AWS 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 対象範囲:コンプライアンスプログラム別」の「)」を参照して、関心のあるコンプライアンスプログラムを選択してください。一般的な情報については、「AWS」を参照してください。

サードパーティの監査レポートはを使用してダウンロードできます AWS Artifact。詳細については、の「レポートのダウンロード」の「 AWS Artifact」を参照してください AWS Artifact。

AWS のサービス を使用する際のコンプライアンス責任は、データの機密性、会社のコンプライアンス目標、および適用される法律と規制によって決まります。 AWS コンプライアンスに役立つ以下のリソースを提供しています。

- セキュリティとコンプライアンスのクイックスタートガイド — これらの導入ガイドでは、アーキテクチャ上の考慮事項について説明し、 AWS セキュリティとコンプライアンスに重点を置いたベースライン環境をデプロイする手順を説明しています。
- Amazon Web Services での HIPAA セキュリティとコンプライアンスのためのアーキテクチャ — このホワイトペーパーでは、企業が HIPAA 対応アプリケーションを作成する方法について説明しています。 AWS

> **Note**
>
> すべての企業が AWS のサービス HIPAA に適格というわけではありません。詳細については、「HIPAA 対応サービスのリファレンス」を参照してください。

- AWS — この一連のワークブックとガイドは、お客様の業界や地域に当てはまる場合があります。
- AWS カスタマー・コンプライアンス・ガイド — コンプライアンスの観点から見た責任分担モデルを理解してください。このガイドは、 AWS のサービス セキュリティを確保するためのベストプラクティスをまとめたもので、複数のフレームワーク (米国標準技術研究所 (NIST)、ペイメントカード業界セキュリティ標準審議会 (PCI)、国際標準化機構 (ISO) など) にわたるセキュリティ管理へのガイダンスをまとめています。
- AWS Config 開発者ガイドのルールによるリソースの評価 — AWS Config このサービスでは、リソース構成が社内慣行、業界ガイドライン、規制にどの程度準拠しているかを評価します。
- AWS Security Hub— AWS のサービス これにより、内部のセキュリティ状態を包括的に把握できます。 AWS Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、「Security Hub のコントロールリファレンス」を参照してください。
- AWS Audit Manager— AWS のサービス これにより、 AWS 使用状況を継続的に監査して、リスクの管理や規制や業界標準への準拠を簡素化できます。

インターフェースエンドポイント () を使用して VPC Lattice にアクセスする PrivateLink

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。インターフェー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、または接続なしで VPC Lattice API にプライベートにアク

セスできるようにするテクノロジーを利用しています。 AWS Direct Connect 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、Amazon VPC ユーザーガイドの「インターフェイス VPC AWS エンドポイントを使用してサービスにアクセスする」を必ず確認してください。

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 サービス用の VPC エンドポイントは、Amazon VPC コンソールまたは () のいずれかを使用して作成できます。 AWS Command Line Interface AWS CLI詳細については、 Amazon VPC ユーザーガイド のインターフェイス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。

詳細については、Amazon VPC ユーザーガイドの「インターフェイス VPC AWS エンドポイントを使用してサービスにアクセスする」を参照してください。

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、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

AWS リージョン 物理的に分離された複数のアベイラビリティーゾーンを提供し、低レイテンシー、高スループット、冗長性の高いネットワークで接続します。

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イ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が高く、フォールトトレラントで、スケーラブルです。

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、「グローバルインフラストラクチャ」を参照してください。AWS

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として、Amazon VPC AWS Latticeはグローバルネットワークセキュリティによって保護されています。 AWS AWS セキュリティサービスとインフラストラクチャを保護する方法については、「AWS Cloud Security」を参照してください。 AWS 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設計するには、「Security Pillar AWS Well-Architected Framework におけるインフラストラクチャ保護」を参照してください。

AWS 公開されている API 呼び出しを使用して、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、できれば TLS 1.3 が必要です。
- DHE (Ephemeral Diffie-Hellman) や ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) などの Perfect Forward Secrecy (PFS) を使用した暗号スイート。これらのモードは、Java 7 以降など、ほとんどの最新システムでサポートされています。

また、リクエストには、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、AWS Security Token ServiceAWS STS を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

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。

目次

- VPC Lattice の CloudWatch メトリクス
- VPC Lattice のアクセスログ
- VPC Lattice の CloudTrail ログ

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 アカウント内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、「VPC Lattice のサービスにリンクされたロールを使用する」を参照してください。

目次

- Amazon CloudWatch メトリクスを表示する
- ターゲットグループのメトリクス
- サービスメトリクス

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) を選択します。
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。

AWS CLI を使ってメトリクスを表示するには

使用可能なメトリクスを表示するには、次の CloudWatch list-metrics AWS CLI コマンドを使用します。

```
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。

ターゲットグループの `HTTP code` と `RequestTime` のメトリクスをモニタリングする場合は、これらのメトリクスをアベイラビリティーゾーン (AZ) でフィルタリングして、ターゲットグループがどの AZ に属しているかを判断できます。

メトリクス	説明
`TotalConnectionCount`	合計接続数。 レポート条件

メトリクス	説明
	• 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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 統計 • 最も有用な統計は `Sum` です。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
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。 [Dimensions] (ディメンション): • 名前: TargetGroup 、値: ターゲットグループの名前。 • 名前: AvailabilityZone 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `TargetGroup` 、値: ターゲットグループの名前。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

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。

サービスの `HTTP code` と `RequestTime` のメトリクスをモニタリングする場合は、これらのメトリクスをアベイラビリティーゾーン (AZ) でフィルタリングして、サービスがどの AZ に属しているかを判断できます。

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。

メトリクス	説明
	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。 [Dimensions] (ディメンション): • 名前: `Service`、値: サービスの ID。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `Service`、値: サービスの ID。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 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。 [Dimensions] (ディメンション): • 名前: `Service`、値: サービスの ID。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

メトリクス	説明
`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。 [Dimensions] (ディメンション): • 名前: `Service`、値: サービスの ID。 • 名前: `AvailabilityZone` 、値: ターゲットグループが属する AZ。

VPC Lattice のアクセスログ

アクセスログには、VPC Lattice サービスに関する詳細情報が記録されます。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。

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 ネイティブに公開されるログは、公開ログと呼ばれます。提供されるログの料金の詳細については、「Amazon CloudWatch 料金表」、「ログ」を選択し、「提供されるログ」で料金表を確認してください。

コンテンツ

- 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
- アクセスログの送信先
- アクセスログの有効化
- アクセスログの内容
- アクセスログのトラブルシューティング

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。

```
{
    "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 分以内にログを CloudWatch Logs に配信します。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。
- CloudWatch ロググループに特定のアクセス許可がない場合、リソースポリシーが自動的に作成され、ロググループに追加されます。詳細については、「Amazon ユーザーガイド」の「ログに送信される CloudWatch ログ」を参照してください。 CloudWatch
- に送信されるアクセスログは、 CloudWatch コンソールの CloudWatch ロググループにあります。詳細については、「Amazon CloudWatch ユーザーガイド」の「ログに送信された CloudWatch 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 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。 CloudWatch
- 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 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「 に送信されたログ Amazon Data Firehose」を参照してください。
- に送信されたログの表示の詳細については Amazon Data Firehose、「 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。

コンテンツ

- 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

サービスネットワークまたはサービスのアクセスログは、作成時に有効にできます。次の手順で説明するように、サービスネットワークまたはサービスの作成後に、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 (https://console.aws.amazon.com/vpc/) を開きます。
2. サービスネットワークまたはサービスを選択します。
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、 でロググループを作成する CloudWatch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. [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 コマンドを使用してcreate-access-log-subscription、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。

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
`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
`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*

例

ログエントリの例を示します。

```
{
    "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
}
```

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。

VPC Lattice の CloudTrail ログ

AWS CloudTrail は、ユーザー、ロール、AWS サービスによって実行されたアクションを記録する AWS のサービスです。CloudTrail は VPC Lattice に対する API コールをイベントとしてキャプ

チャします。CloudTrail は、アカウント作成時に AWS アカウント で有効になります。VPC Lattice でアクティビティが発生すると、そのアクティビティは、他の AWS のサービスのイベントとともに、CloudTrail イベントとして [イベント履歴] に記録されます。キャプチャされたコールには、VPC Lattice コンソールからのコールと、VPC Lattice API オペレーションへのコードコールが含まれます。CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

CloudTrail のログファイルには、単一か複数のログエントリがあります。イベントはあらゆるソースからの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメータなどの情報が含まれます。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、特定の順序では表示されません。証跡は、指定した S3 バケットにイベントをログファイルとして配信できるようにする CloudTrail 設定です。

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。

VPC Lattice のログファイルエントリを理解する

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルには、単一か複数のログエントリがあります。イベントはあらゆるソースからの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストのパラメータなどの情報が含まれます。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、特定の順序では表示されません。

ログ内のキーと値のペアに関する情報については、「AWS CloudTrail ユーザーガイド」の「CloudTrail レコードの内容」を参照してください。

CreateService API アクションのコールのログエントリの例を以下に示します。

```
{
  "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"
}
```

DeleteService API アクションのコールのログエントリの例を以下に示します。

```
{
  "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"
}
```

Amazon VPC Lattice のクォータ

AWS アカウント には、各 について、以前は制限と呼ばれていたデフォルトのクォータがあります AWS のサービス。特に明記されていない限り、クォータは地域固有です。一部のクォータについては引き上げをリクエストできますが、その他のクォータについては引き上げることはできません。

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、 AWS サポートに問い合わせるか、Service Quotas ユーザーガイドの「クォータの引き上げのリクエスト」を参照してください。 Service Quotas

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	いいえ	認可ポリシー内の JSON ファイルの最大サイズ。
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	はい	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。
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 : 5	はい	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。

名前	デフォルト	引き上げ可能	説明
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	いいえ	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	はい	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: 10	はい	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。
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：500	はい	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。

名前	デフォルト	引き上げ可能	説明
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	はい	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：5	はい	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	はい	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。
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	はい	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。

以下の制限も適用されます。

制限	値
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
接続あたりの最大送信単位 (MTU)	8500 バイト
アベイラビリティーゾーンごとのサービスごとの 1 秒あたりのリクエスト数	10,000

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
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 による AWS マネージドポリシーへの変更の報告を開始	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」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 による AWS マネージドポリシーへの変更の報告を開始	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。