aws

用户指南

Amazon VPC Lattice

Amazon VPC Lattice: 用户指南

Table of Contents

什么是 Amazon VPC Lattice？

Amazon VPC Lattice 是一项完全托管的应用程序网络服务，用于连接、保护和监控应用程序的服务和资源。您可以将VPC Lattice与单个虚拟私有云 (VPC) 配合使用，也可以通过一个或多个账户跨多个虚拟私 VPCs 有云 (VPC) 使用。

现代应用程序可以由多个小型模块化组件组成，这些组件通常被称为微服务，例如 HTTP API，数据库等资源，以及由 DNS 和 IP 地址端点组成的自定义资源。虽然现代化有其优势，但当你连接这些微服务和资源时，它也可能带来网络复杂性和挑战。例如，如果开发人员分散在不同的团队中，他们可能会跨多个账户构建和部署微服务和资源，或者 VPCs。

在 VPC Lattice 中，我们将微服务称为服务，仅将资源表示为资源配置。这些是您在VPC Lattice用户指南中看到并将使用的术语。

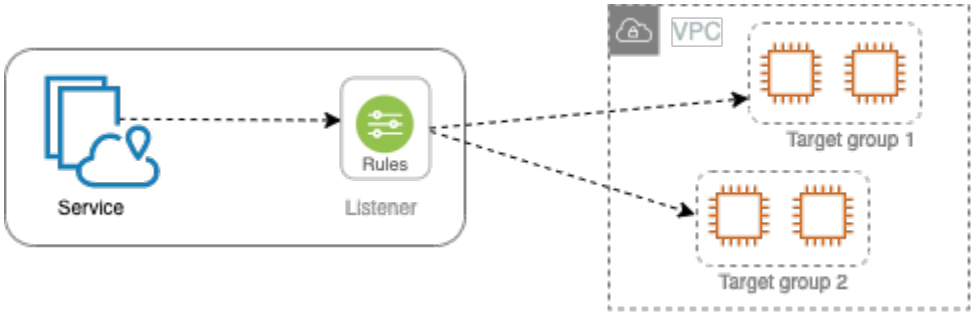
内容

关键组件

要使用 Amazon VPC Lattice，您应该熟悉其关键组件。

服务

> 一种可独立部署的软件单元，用于提供特定的任务或函数。服务可以在 EC2 实例或ECS/EKS/Fargate容器上运行，也可以作为 Lambda 函数在账户或虚拟私有云 (VPC) 中运行。VPC Lattice 服务包含以下组件：目标组、侦听器和规则。



目标群体

运行应用程序或服务的资源集合，也称为目标。这些目标类似于弹性负载均衡提供的目标组，但不可互换。支持的目标类型包括 EC2 实例、IP 地址、Lambda 函数、应用程序负载均衡器、Amazon ECS 任务和 Kubernetes Pod。

侦听器

此为检查连接请求并将其路由到目标组中目标的过程。您可以使用协议和端口号配置监听器。

规则

侦听器的默认组件，用于将请求转发到 VPC Lattice 目标组中的目标。每条规则由优先级、一个或多个操作以及一个或多个条件组成。规则决定了侦听器路由客户端请求的方式。

资源

资源是一个实体，例如亚马逊关系数据库服务 (Amazon RDS) 数据库、Amazon EC2 实例、应用程序终端节点、域名目标或 IP 地址。您可以通过在 AWS Resource Access Manager (AWS RAM) 中创建资源共享、创建资源网关和定义资源配置来共享 VPC 中的资源。

资源网关

资源网关是资源所在的 VPC 的入口点。

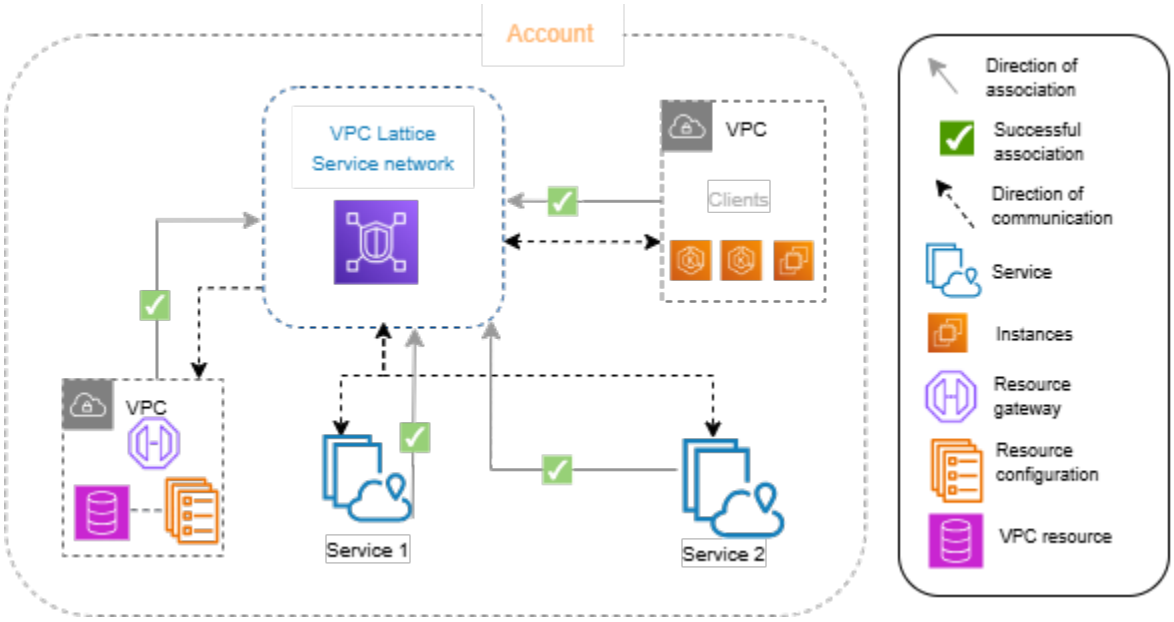
资源配置

资源配置是一个逻辑对象，它表示单个资源或一组资源。资源可以是 IP 地址、域名目标或 Amazon RDS 数据库。

服务网络

服务和资源配置集合的逻辑边界。客户端可以位于与服务网络关联的 VPC 中。如果获得授权，与同一服务网络关联的客户端和服务可以相互通信。

在下图中，客户端可以与两个服务通信，因为 VPC 和服务与同一个服务网络关联。



服务目录

一个中央注册中心，其中包含您拥有或通过 AWS RAM该账户与之共享的所有VPC Lattice服务。

验证策略

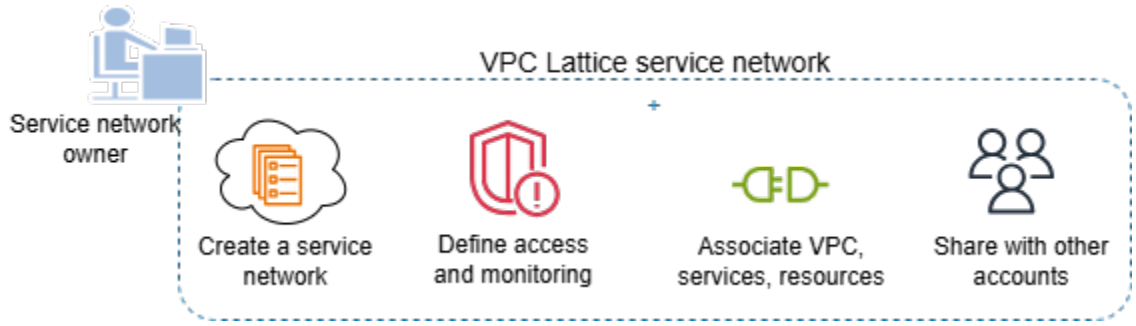
可用于定义服务访问权限的细粒度授权策略。您可以将单独的验证策略附加到单个服务或服务网络。例如，您可以创建一个策略，说明在 EC2 实例的自动扩缩组上运行的支付服务，应如何与在 AWS Lambda中运行的计费服务交互。

资源配置不支持身份验证策略。服务网络的身份验证策略不适用于服务网络中的资源配置。

角色和责任

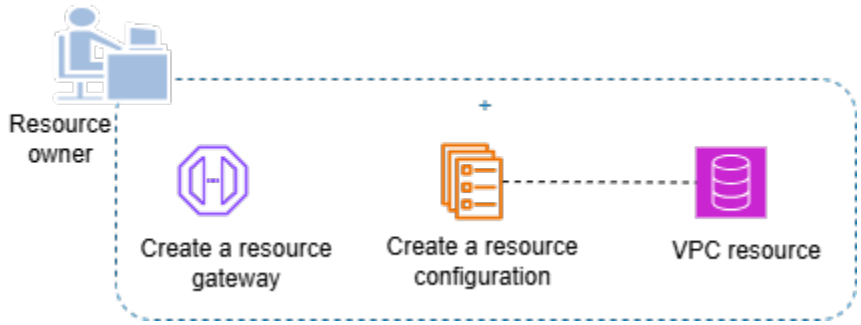
角色决定了谁负责 Amazon VPC Lattice 中信息的设置和流动。通常有两个角色：服务网络所有者和服务所有者，他们的职责可能重叠。

服务网络所有者：服务网络所有者通常是组织中的网络管理员或云管理员。服务网络所有者负责创建、共享和配置服务网络。他们还管理可以访问 VPC Lattice 中服务网络或服务的人员。服务网络所有者可以为与服务网络关联的服务定义粗粒度的访问设置。这些控制用于管理使用验证和授权策略的客户端和服务之间的通信。如果服务或资源配置与服务网络所有者的账户共享，则服务网络所有者还可以将服务或资源配置与单个或多个服务网络相关联。



服务所有者-服务所有者通常是组织中的软件开发人员。服务所有者负责在 VPC Lattice 中创建服务，定义路由规则，将服务与服务网络关联。他们还可以定义细粒度访问设置来限制访问，只有经过验证和授权的服务及客户端才可访问。

资源所有者-资源所有者通常是组织中的软件开发人员，担任数据库等资源的管理员。资源所有者为资源创建资源配置，定义资源配置的访问设置，并将资源配置与服务网络关联起来。



功能

以下是 VPC Lattice 提供的核心功能。

服务发现

与服务网络 VPCs 关联的所有客户机和服务都可以与同一服务网络中的其他服务进行通信。DNS 将service-to-service流量定向 client-to-service并通过 VPC 莱迪思终端节点。当客户端向服务发送请

求时，会使用服务的 DNS 名称。Route 53 Resolver 将流量发送到 VPC Lattice，然后由其识别目标服务。

连接

Client-to-service 并在 AWS 网络基础架构内建立 client-to-resource连接。当您将 VPC 与服务网络关联时，VPC 中的任何客户端都可以（通过资源配置）连接到服务网络中的服务和资源，前提是他们拥有所需的访问权限。VPC 莱迪思支持重叠的 CIDR 技术。

本地访问

您可以使用 VPC 终端节点（由提供支持 AWS PrivateLink）启用从 VPC 到服务网络的连接。服务网络类型的 VPC 终端节点允许您通过 Direct Connect 和 VPN 从本地网络访问服务网络中的服务和资源。通过 VPC 对等连接或 AWS Transit Gateway 也可以通过 VPC 终端节点访问资源和服务的流量。

可观测性

VPC Lattice 为遍历服务网络的每个请求和响应生成指标和日志，以帮助您监控应用程序并对其进行问题排查。默认情况下，指标会发布到服务所有者账户。服务所有者和资源所有者可以选择开启日志记录，并接收所有客户端 access/requests 的服务和资源的日志。服务网络所有者还可以开启服务网络上的登录功能， access/requests 将来自连接到服务网络 VPCs 的客户端的所有服务和资源记录下来。

VPC Lattice 使用以下工具来帮助您监控和排除服务故障： Amazon CloudWatch 日志组、Firehose 传输流和 Amazon S3 存储桶。

安全性

VPC Lattice 提供了一个框架，您可以使用该框架在网络的多个层实施防御策略。第一层是服务、资源配置、VPC 关联和服务网络类型的 VPC 端点的组合。如果没有 VPC 和服务关联或服务网络类型的 VPC 终端节点，客户端就无法访问服务。同样，如果没有 VPC 和资源配置以及服务关联或服务网络类型的 VPC 终端节点，客户端就无法访问资源。

第二层使用户能够将安全组附加到 VPC 和服务网络之间的关联。第三层和第四层是验证策略，可以在服务网络级别和服务级别单独应用。

可用区亲和力

VPC Lattice 支持可用区 (AZ) 关联性来路由流量。当客户端向 VPC Lattice 发送请求时，VPC Lattice 会使用与客户端位于同一可用区的服务或资源的 IP 地址进行响应。如果该可用区不可用，VPC Lattice 将使用其他 AZs可用区的 IP 地址进行响应。从 VPC Lattice 到目标，路由是指向可能分布在各 AZs处的目标。此外，VPC Lattice 不收取可用区间数据传输费用。

访问 VPC Lattice

您可以使用以下任意界面创建、访问和管理 VPC Lattice：

- AWS 管理控制台：提供可用于访问 VPC Lattice 的 Web 界面。
- AWS Command Line Interface (AWS CLI) — 为包括 VPC Lattice 在内的一系列 AWS 服务提供命令。在 AWS CLI Windows、macOS 和 Linux 上都支持。有关 CLI 的更多信息，请参阅 AWS Command Line Interface。有关更多信息 APIs，请参阅 Amazon VPC Lattice API 参考。
- 适用于 Kubernetes 的 VPC Lattice 控制器：管理 Kubernetes 集群的 VPC Lattice 资源。有关将 VPC Lattice 与 Kubernetes 结合使用的更多信息，请参阅 AWS 网关 API 控制器用户指南。
- CloudFormation：帮助您对 AWS 资源进行建模和设置。有关 API 的更多信息，请参阅 Amazon VPC Lattice 资源类型参考。

VPC 莱迪思服务终端节点

端点是用作 AWS Web 服务入口点的 URL。VPC Lattice 支持以下终端节点类型：

- the section called “IPv4 端点”
- 双栈端点（同时支持和 IPv4 ） IPv6

当您发出请求时，您可以指定要使用的端点。如果您未指定终端节点，则默认使用该 IPv4终端节点。要使用不同的端点类型，您必须在请求中指定。有关如何执行此操作的示例，请参阅the section called “指定端点”。有关可用终端节点的表，请参阅 Amazon VPC Lattice 终端节点。

IPv4 端点

IPv4 端点仅支持 IPv4 流量。 IPv4 终端节点适用于所有区域。

如果您指定通用端点 `vpc-lattice.amazonaws.com`，则我们将端点用于 `us-east-1`。要使用其他区域，请指定其关联端点。例如，如果您指定 `vpc-lattice.us-east-2.amazonaws.com` 为端点，我们会将您的请求定向到 `us-east-2` 端点。

IPv4 端点名称使用以下命名约定：

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

例如，该`eu-west-1`区域的 IPv4 终端节点名称为`vpc-lattice.eu-west-1.amazonaws.com`。

双栈（IPv4 和 IPv6）端点

双栈端点同时支持 IPv4 和流量。 IPv6 双栈终端节点适用于所有区域。当您向双栈端点发出请求时，终端节点 URL 会解析为 IPv6 或 IPv4 地址，具体取决于您的网络和客户端使用的协议。

双堆栈端点名称使用以下命名约定：

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

例如，`eu-west-1` 区域的双堆栈端点名称是 `vpc-lattice.eu-west-1.api.aws`。

指定端点

以下示例说明如何使用 for 为`us-east-2`区域指定终端节点`vpc-lattice`。 AWS CLI

- IPv4

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

- 双堆栈

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

定价

VPC Lattice 允许您根据服务配置时间、通过每个服务传输的数据量，以及请求数量付费。作为资源所有者，您需要为传入和传出每种资源的数据付费。作为服务网络所有者，您需要按小时为与服务网络相关的资源配置付费。作为拥有与服务网络关联的 VPC 的使用者，您需要为从您的 VPC 向服务网络中的资源传输和传出数据付费。有关更多信息，请参阅 Amazon VPC Lattice 定价。

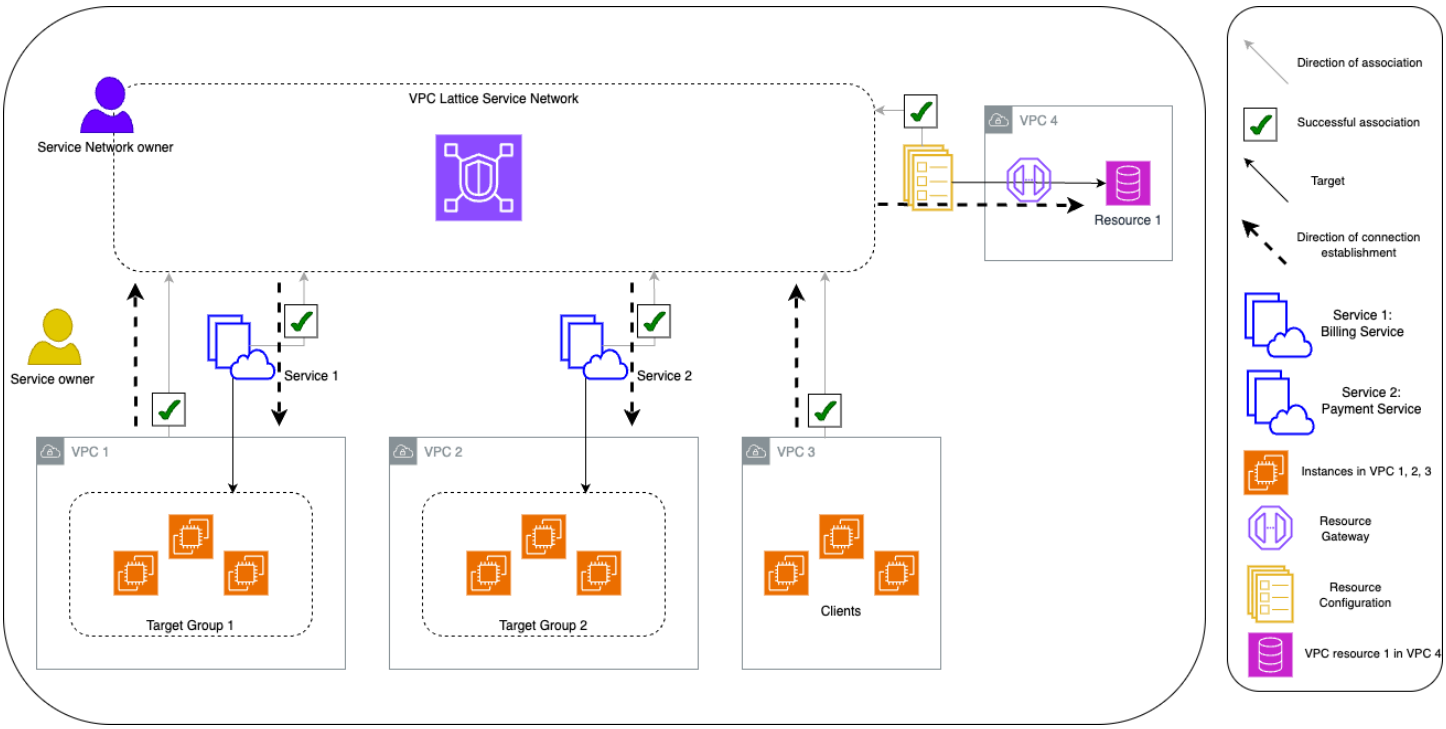
VPC Lattice 的工作原理

VPC Lattice旨在帮助您轻松有效地发现、保护、连接和监控其中的所有服务和资源。VPC Lattice 中的每个组件根据组件与服务网络的关联以及访问设置，在服务网络内进行单向或双向通信。访问设置由此通信所需的验证和授权策略组成。

以下摘要描述了 VPC Lattice 内组件之间的通信：

- VPC 可以通过两种方式连接到服务网络：通过 VPC 关联和通过服务网络类型的 VPC 终端节点。
- 与服务网络关联的服务和资源可以接收来自其 VPC 也连接到服务网络的客户端的请求。
- 只有当客户端位于连接到同一服务网络的 VPC 中时，该客户端才能向与服务网络关联的服务和资源发送请求。仅当 VPC 通过 VPC 终端节点连接到服务网络时，通过 VPC 对等连接、传输网关、Direct Connect 或 VPN 的客户端流量才能到达资源和服务。
- VPC 中与服务网络关联的服务目标也是客户端，可以向与服务网络关联的其他服务和资源发送请求。
- VPC 中与服务网络无关的服务目标不是客户端，也无法向与服务网络关联的其他服务和资源发送请求。
- VPC 中拥有资源但 VPC 未与服务网络关联的客户端不是客户端，无法向与服务网络关联的其他服务和资源发送请求。

以下流程图使用示例场景解释 VPC Lattice 内组件之间的信息流和通信方向。有两个服务与服务网络关联。服务和所有 VPC 都是在与服务网络相同的账户中创建的。这两个服务都配置为允许来自服务网络的流量。



服务 1 是在一组实例上运行的计费应用程序，这些实例已在 VPC 1 的目标组 1 中注册。服务 2 是在一组实例上运行的支付应用程序，这些实例已在 VPC 2 的目标组 2 中注册。VPC 3 处于同一账户中，有客户端但没有服务。资源 1 是一个在 VPC 4 中包含客户数据的数据库。

以下列表按顺序描述了 VPC Lattice 的典型任务工作流。

1. 创建服务网络

   服务网络所有者创建服务网络。

2. 创建服务

   服务所有者创建相应的服务：服务 1 和服务 2。在创建过程中，服务所有者会添加侦听器，并为每项服务定义将请求路由到目标组的规则。

3. 定义路由

   服务所有者为每个服务创建目标组（目标组 1 和目标组 2）。他们通过指定运行服务的目标实例来做到这一点。服务所有者还指定了这些目标所在的 VPC。

   在上图中，实心箭头表示将流量路由到目标组的服务，以及路由到资源的资源配置。

   VPC Lattice 支持可用区 (AZ) 关联性来路由流量。当客户端向 VPC Lattice 发送请求时，VPC Lattice 会使用与客户端位于同一可用区的服务或资源的 IP 地址进行响应。如果该可用区不可

用，VPC Lattice 将使用来自其他可用区的 IP 地址进行响应。从 VPC Lattice 到目标，路由指向目标，这些目标可能分布在多个可用区。此外，VPC Lattice 不收取可用区间数据传输费用。

4. 将服务与服务网络关联

   服务网络所有者或服务所有者将服务与服务网络关联。这些关联显示为带有复选标记的箭头，从服务指向服务网络。当您将服务与服务网络关联时，与该服务网络关联的其他服务以及连接到该服务网络的 VPC 中的客户端可以发现该服务。

   服务网络和目标组之间的虚线箭头表示连接建立的方向。使用服务网络将流量返回到客户端。此图中不包括代表返回流量的箭头。

5. 创建资源网关

   资源所有者在 VPC 4 中创建资源网关，以便能够实现从客户端到资源 1 的连接。

6. 创建资源配置

   资源所有者创建代表资源 1 的资源配置，并为资源 1 指定资源网关。

7. 将资源配置与服务网络相关联

   服务网络所有者或资源所有者将资源配置与服务网络相关联。关联显示为带有复选标记的箭头，指向资源配置中的服务网络。当您将资源配置与服务网络关联时，与该服务网络关联的其他服务以及连接到服务网络的 VPC 中的客户机可以发现该资源配置。

   从服务网络到资源的虚线箭头表示接收来自客户端的请求的资源。使用服务网络将流量返回到客户端。此图中不包括代表返回流量的箭头。

8. 将 VPC 与服务网络连接

   VPC 可以通过两种方式与服务网络连接：将 VPC 关联到服务网络，或者创建 VPC 终端节点。在这里，服务网络所有者将 VPC 1 和 VPC 3 与服务网络关联。关联使用指向服务网络的带有复选标记的箭头显示。通过这些关联，VPC 中的任何资源都可以充当客户端，并可以向服务网络内的服务发出请求。VPC 1 和服务网络之间的虚线箭头表示建立连接的方向。服务网络仅启动指向 service 1 目标组目标资源的连接。VPC 1 中的任何资源都可以充当客户端，发起与服务网络、服务和资源的连接。

   VPC 2 没有代表关联的箭头或复选标记。这意味着服务网络所有者或服务所有者尚未将 VPC 2 与服务网络关联。这是因为在本例中，服务 2 只需要使用相同的请求来接收请求和发送响应。换句话说，服务 2 的目标不是客户端，不需要向服务网络中的其他服务发出请求。

同样，VPC 4 没有代表关联的箭头或复选标记。这意味着服务网络所有者或资源所有者尚未将 VPC 4 与服务网络关联。这是因为资源 1 仅使用相同的请求接收请求和发送响应。它无法向服务网络中的其他服务和资源发出请求。

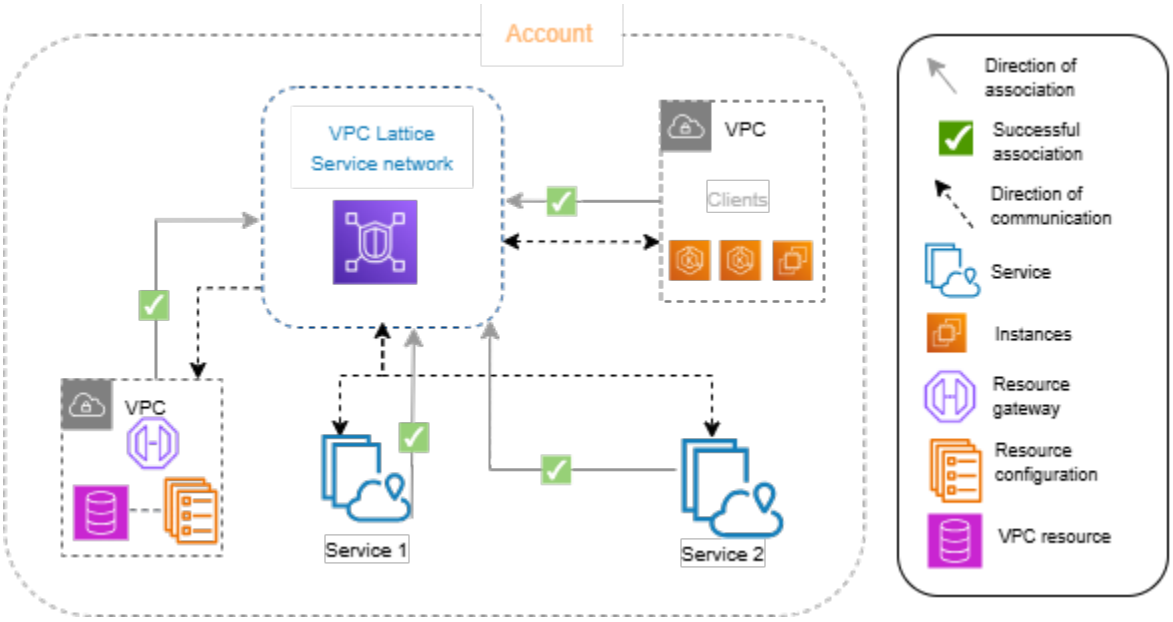
总而言之，程序示意图显示了以下场景：

- 仅通过入口连接的VPC从VPC Lattice连接到其资源。VPC 2 和 VPC 4 代表这些场景。
- 一个 VPC 仅提供从其资源到 VPC Lattice 的出口连接。VPC 3 代表了这种情况。
- 一个 VPC，具有从 VPC Lattice 到其资源的入口连接以及从其资源到 VPC Lattice 的出口连接。VPC 1 代表这种情况。

VPC Lattice 中的服务网络

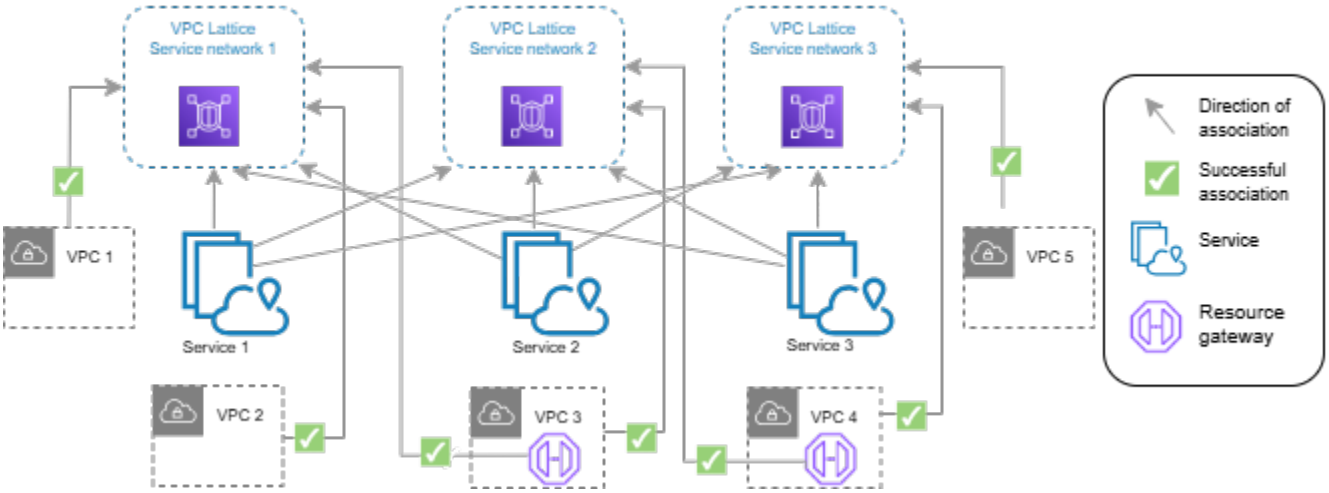
服务网络是一系列服务和资源配置的逻辑边界。可以对与网络相关的服务和资源配置进行发现、连接、可访问性和可观察性授权。要向网络中的服务和资源配置发出请求，您的服务或客户端必须位于通过关联或 VPC 终端节点连接到服务网络的 VPC 中。

下图显示了 Amazon VPC Lattice 中典型服务网络的关键组件。箭头上的复选标记表示服务和 VPC 与服务网络关联。与服务网络关联的 VPC 中的客户端可以通过服务网络与两个服务通信。



您可以将一个或多个服务和资源配置与多个服务网络相关联。您也可以 VPCs 使用一个服务网络连接多个网络。您只能通过关联将一个 VPC 连接到一个服务网络。要将一个 VPC 连接到多个服务网络，您可以使用服务网络类型的 VPC 终端节点。有关服务网络类型的 VPC 终端节点的更多信息，请参阅AWS PrivateLink 用户指南。

在下图中，箭头表示服务和服务网络之间的关联，以及 VPCs 和服务网络之间的关联。您可以看到多个服务与多个服务网络相关联，多个服务网络 VPCs 关联到每个服务网络。每个 VPC 与服务网络只有一个关联。但是，VPC 3 和 VPC 4 连接到两个服务网络。VPC 3 通过 VPC 终端节点连接到服务网络 1。同样，VPC 4 通过 VPC 终端节点连接到服务网络 2。



有关更多信息，请参阅 Amazon VPC Lattice 的配额。

内容

创建 VPC 莱迪思服务网络

使用控制台创建服务网络，并可选择为其配置服务、关联、访问设置和访问日志。

要使用控制台创建服务网络

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务网络。
3. 选择创建服务网络。
4. 对于标识符，输入名称、可选描述和可选标签。名称长度必须介于 3 到 63 个字符之间。您可以使用小写字母、数字和连字符。名称必须以字母或数字开头和结尾。不要使用连续的连字符。描述最多可包含 256 个字符。要添加标签，请选择添加新标签，然后指定标签键和标签值。

5. （可选）要关联服务，请从服务关联和服务中选择服务。该列表包含您账户中的服务，以及从其他账户与您共享的任何服务。如果列表中没有任何服务，您可以选择创建 VPC Lattice 服务来创建服务。

   或者，要在创建服务网络后关联服务，请参阅 the section called “管理服务网络服务关联”。

6. （可选）要关联资源配置，请从资源配置关联、资源配置中选择资源配置服务。该列表包括您的账户中的资源配置以及从其他账户与您共享的所有资源配置。如果列表中没有任何资源配置，则可以通过选择创建 Amazon VPC Lattice 资源配置来创建资源配置。

   或者，要在创建服务网络后关联资源配置，请参阅the section called “管理服务网络资源关联”。

7. （可选）要关联 VPC，请选择添加 VPC 关联。从 VPC 中选择要关联的 VPC，然后从安全组中最多选择 5 个安全组。要创建安全组，请选择创建新安全组。

   或者，您可以跳过此步骤，使用 VPC 终端节点（由提供支持 AWS PrivateLink）将 VPC 连接到服务网络。有关更多信息，请参阅AWS PrivateLink 用户指南中的访问服务网络。

8. 创建服务网络时，必须决定是否打算与其他账户共享服务网络。您的选择是不可变的，并且在创建服务网络后无法更改。如果您选择允许共享，则可以通过与其他帐户共享服务网络 AWS Resource Access Manager。

   要与其他账户共享您的服务网络，请从 AWS RAM 资源共享中选择资源共享。

   要创建资源共享，请转到 AWS RAM 控制台并选择创建资源共享。

9. 对于网络访问，如果您希望关联 VPCs 中的客户端访问此服务网络中的服务，则可以保留默认的身份验证类型 “无”。要应用验证策略来控制对服务的访问，请选择 AWS IAM，然后对验证策略执行以下操作之一：

   - 在输入字段中输入策略。对于可以复制和粘贴的示例策略，请选择策略示例。
   - 选择应用策略模板，然后选择允许已验证和未验证访问模板。此模板允许来自其他账户的客户端通过签署请求（表示已验证），或以匿名方式（表示未验证）访问服务。
   - 选择应用策略模板，然后选择允许仅限已验证访问模板。此模板允许来自其他账户的客户端通过签署请求（表示已验证）访问服务。

10. （可选）要开启访问日志，请选择访问日志切换开关，并按如下方式指定访问日志的目标：

    - 选择CloudWatch 日志组，然后选择一个 CloudWatch 日志组。要创建日志组，请选择在中创建日志组 CloudWatch。
    - 选择 S3 存储桶并输入 S3 存储桶路径，包括任何前缀。要搜索 S3 存储桶，请选择浏览 S3。
    - 选择 Kinesis Data Firehose 传输流，然后选择一个传输流。要创建传输流，请选择在 Kinesis 中创建传输流。

11. （可选）要与其他账户共享您的服务网络，请从 AWS RAM 资源共享中选择资源共享。要创建资源共享，请选择在 RAM 控制台中创建资源共享。
12. 在摘要部分查看您的配置，然后选择创建服务网络。

要使用创建服务网络 AWS CLI

使用 create-service-network 命令。此命令仅创建基本服务网络。要创建功能齐全的服务网络，还必须使用创建服务关联、VPC 关联和访问设置的命令。

管理 VPC Lattice 服务网络的关联

当您将服务或资源配置与服务网络关联时，它允许 VPCs 连接到服务网络的客户机向服务和资源配置发出请求。当您将 VPC 与服务网络连接时，它会使该 VPC 中的所有目标成为客户端，并与服务网络中的其他服务和资源配置进行通信。

服务网络资源关联的私有 DNS 启用属性会覆盖服务网络终端节点和服务网络 VPC 关联的启用私有 DNS 的属性。

如果服务网络所有者创建了服务网络资源关联但未启用私有 DNS，则即使在服务网络终端节点或服务网络 VPC 关联上启用了私有 DNS VPCs ，VPC Lattice 也不会在服务网络所连接的任何资源配置中为该资源配置预置私有托管区域。

内容

管理服务网络服务关联

您可以关联您账户中的服务，或关联不同账户与您共享的服务。这是创建服务网络时的一个可选步骤。但在关联服务之前，服务网络无法完全正常运行。如果服务所有者的账户有必要的访问权限，就可以将自己的服务与服务网络关联。有关更多信息，请参阅 Identity-based VPC 莱迪思的策略示例。

删除服务关联后，该服务将无法再连接到服务网络中的其他服务。

要使用控制台管理服务关联

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务网络。
3. 选择服务网络名称以打开其详细信息页面。
4. 选择服务关联选项卡。
5. 要创建关联，请执行以下操作：
   a. 选择创建关联。
   b. 从服务中选择一个服务。要创建服务，请选择创建 Amazon VPC Lattice 服务。
   c. （可选）要添加标签，请展开服务关联标签，选择添加新标签，然后输入标签键和标签值。
   d. 选择保存更改。
6. 要删除关联，请选中关联的复选框，然后依次选择操作和删除服务关联。提示进行确认时，输入 **`confirm`**，然后选择删除。

要使用创建服务关联 AWS CLI

使用 create-service-network-service-association 命令。

要删除服务关联，请使用 AWS CLI

使用 delete-service-network-service-association 命令。

管理服务网络资源关联

资源配置是一个逻辑对象，它代表单个资源或一组资源。您可以关联位于您的账户中的资源配置或来自不同账户的与您共享的资源配置。这是创建服务网络时的一个可选步骤。如果资源配置所有者的账户具有所需的访问权限，则可以将其资源配置与服务网络相关联。有关更多信息，请参阅 VPC Lattice 基于身份的策略示例。

管理服务网络和资源配置之间的关联

您可以创建或删除服务网络和资源配置之间的关联。

使用控制台管理资源配置关联

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的PrivateLink 和莱迪思下，选择服务网络。

3. 选择服务网络名称以打开其详细信息页面。
4. 选择资源配置关联选项卡。
5. 要创建关联，请执行以下操作：
   a. 选择创建关联。
   b. 对于资源配置，请选择资源配置。
   c. 对于 DNS 名称，选择启用私有 DNS，以允许 VPC Lattice 根据资源配置的域名为您的资源配置关联配置配置配置配置私有托管区域。
   d. （可选）要添加标签，请展开服务关联标签，选择添加新标签，然后输入标签键和标签值。
   e. 选择保存更改。
6. 要删除关联，请选中关联的复选框，然后依次选择操作和删除。提示进行确认时，输入 **`confirm`**，然后选择删除。

要使用创建资源配置关联 AWS CLI

使用 create-service-network-resource-association 命令。

要删除资源配置关联，请使用 AWS CLI

使用 delete-service-network-resource-association 命令。

管理服务网络 VPC 关联

如果客户端与服务网络相关联，则客户端可以向与服务网络关联的资源配置中 VPCs 指定的服务和资源发送请求。通过 VPC 对等连接或传输网关的客户端流量只能通过使用服务网络类型的 VPC 终端节点的服务网络。

创建服务网络时，关联 VPC 是一个可选步骤。如果网络所有者的账户具有所需的访问权限，则可以关联 VPCs 到服务网络。有关更多信息，请参阅 Identity-based VPC 莱迪思的策略示例。

在创建与资源配置的 VPC 关联时，您可以指定私有 DNS 首选项。此首选项允许 VPC Lattice 代表资源使用者配置私有托管区域。有关更多信息，请参阅 the section called “资源提供商的自定义域名”。

当您删除 VPC 关联时，中的客户端将 VPCs 无法再连接到服务网络中的服务。

要使用控制台管理 VPC 关联

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。

2. 在导航窗格中的 VPC Lattice 下，选择服务网络。
3. 选择服务网络名称以打开其详细信息页面。
4. 选择 VPC 关联选项卡。
5. 要创建 VPC 关联，请执行以下操作：

    a. 选择创建 VPC 关联。
    b. 选择添加 VPC 关联。
    c. 从 VPC 中选择一个 VPC，然后从安全组中最多选择 5 个安全组。要创建安全组，请选择创建新安全组。
    d. （可选）要允许 VPC Lattice 根据资源配置的域名配置私有托管区域，请在 DNS 名称中选择“启用 DNS 名称”，然后执行以下操作：

        i. 在“私有 DNS 首选项”中，选择一个首选项。

            如果您选择所有域，则VPC Lattice会为资源配置的任何自定义域名配置一个私有托管区域。

        ii. （可选）如果您选择已验证和指定域或指定域，请输入一个以逗号分隔的域名列表，列出您希望 VPC Lattice 为其配置托管区域。VPC Lattice 只有在托管区域与您的私有域列表匹配时才会配置该区域。您可以使用通配符匹配。

    e. （可选）要添加标签，请展开 VPC 关联标签，选择添加新标签，然后输入标签键和标签值。
    f. 选择保存更改。
6. 要编辑关联的安全组，请选中关联的复选框，然后依次选择操作和编辑安全组。根据需要添加和删除安全组。
7. 要删除关联，请选中关联的复选框，然后依次选择操作和删除 VPC 关联。提示进行确认时，输入 **`confirm`**，然后选择删除。

使用创建 VPC 关联 AWS CLI

使用 create-service-network-vpc-association 命令。

要更新 VPC 关联的安全组，请使用 AWS CLI

使用 update-service-network-vpc-association 命令。

要删除 VPC 关联，请使用 AWS CLI

使用 delete-service-network-vpc-association 命令。

管理服务网络 VPC 终端节点关联

客户端可以通过其 VPC 中的 VPC 终端节点（由 AWS PrivateLink）向资源配置中指定的服务和资源发送请求。服务网络类型的 VPC 终端节点将 VPC 连接到服务网络。通过 VPC 对等连接、Transit Gateway、Direct Connect 或 VPN 来自 VPC 外部的客户端流量可以使用 VPC 终端节点访问服务和资源配置。使用 VPC 终端节点，您可以将一个 VPC 连接到多个服务网络。在 VPC 中创建 VPC 终端节点时，将使用来自该 VPC 的 IP 地址（而不是托管前缀列表中的 IP 地址）来建立与服务网络的连接。

在创建与资源配置的 VPC 关联时，您可以指定私有 DNS 首选项。此首选项允许 VPC Lattice 代表资源使用者配置私有托管区域。有关更多信息，请参阅 the section called “资源提供商的自定义域名”。

使用控制台管理 VPC 终端节点关联

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务网络。
3. 选择服务网络名称以打开其详细信息页面。
4. 选择终端节点关联选项卡，查看连接到您的服务网络的 VPC 终端节点。
5. 选择 VPC 终端节点的终端节点 ID 以打开其详细信息页面。然后修改或删除 VPC 终端节点关联。

使用控制台创建新的 VPC 终端节点关联

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格的 VPC Lattice 下，选择终端节点。
3. 选择创建端点。
4. 对于类型，选择服务网络。
5. 选择要连接到 VPC 的服务网络。
6. 选择 VPC、子网和安全组。
7. （可选）要启用私有 DNS，请选择启用私有 DNS。
8. （可选）要添加标签，请展开 VPC 关联标签，选择添加新标签，然后输入标签键和标签值。
9. 选择创建端点。

要详细了解 VPC 终端节点以及如何连接到服务网络，请参阅AWS PrivateLink 用户指南中的访问服务网络。

编辑 VPC 莱迪思服务网络的访问设置

访问设置使您能够配置和管理客户端对服务网络的访问。访问设置包括验证类型和验证策略。验证策略可帮助您验证和授权流向 VPC Lattice 服务的流量。服务网络的访问设置不适用于与服务网络关联的资源配置。

您可以在服务网络级别、服务级别，或同时在两个级别应用验证策略。通常，验证策略由网络所有者或云管理员应用。他们可以实现粗粒度授权，例如，允许来自组织内部的经过身份验证的呼叫，或者允许符合特定条件的匿名 GET 请求。在服务级别，服务所有者可以应用细粒度控制，这种控制可能更具限制性。有关更多信息，请参阅 使用身份验证策略控制对VPC莱迪思服务的访问。

要使用控制台添加或更新访问策略

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务网络。
3. 选择服务网络名称以打开其详细信息页面。
4. 选择访问选项卡以查看当前的访问设置。
5. 要更新访问设置，请选择编辑访问设置。
6. 如果您希望关联中的客户端 VPCs 访问此服务网络中的服务，请为身份验证类型选择无。
7. 要将资源策略应用于服务网络，则对于验证类型，选择 AWS IAM，并对验证策略执行以下操作之一：
   - 在输入字段中输入策略。对于可以复制和粘贴的示例策略，请选择策略示例。
   - 选择应用策略模板，然后选择允许已验证和未验证访问模板。此模板允许来自其他账户的客户端通过签署请求（表示已验证），或以匿名方式（表示未验证）访问服务。
   - 选择应用策略模板，然后选择允许仅限已验证访问模板。此模板允许来自其他账户的客户端通过签署请求（表示已验证）访问服务。
8. 选择保存更改。

使用添加或更新访问策略 AWS CLI

使用 put-auth-policy 命令。

编辑VPC莱迪思服务网络的监控详情

VPC Lattice 会为每个请求和响应生成指标和日志，从而提高了监控和排查应用程序问题的效率。

您可以启用访问日志，并为日志指定目标资源。VPC Lattice 可以将日志发送到以下资源：CloudWatch 日志组、Firehose 传输流和 S3 存储桶。

要使用控制台启用访问日志或更新日志目标

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务网络。
3. 选择服务网络名称以打开其详细信息页面。
4. 选择监控选项卡。检查访问日志，查看访问日志是否已启用。
5. 要启用或禁用访问日志，请选择编辑访问日志，然后打开或关闭访问日志切换开关。
6. 在启用访问日志时，必须选择传输目标的类型，然后创建或选择访问日志的目标。您还可以随时更改传输目标。例如：
   - 选择CloudWatch 日志组，然后选择一个 CloudWatch 日志组。要创建日志组，请选择在中创建日志组 CloudWatch。
   - 选择 S3 存储桶并输入 S3 存储桶路径，包括任何前缀。要搜索 S3 存储桶，请选择浏览 S3。
   - 选择 Kinesis Data Firehose 传输流，然后选择一个传输流。要创建传输流，请选择在 Kinesis 中创建传输流。
7. 选择保存更改。

要启用访问日志，请使用 AWS CLI

使用 create-access-log-subscription 命令。

要更新日志目标，请使用 AWS CLI

使用 update-access-log-subscription 命令。

要禁用访问日志，请使用 AWS CLI

使用 delete-access-log-subscription 命令。

管理VPC莱迪思服务网络的标签

标签有助于您以不同的方式对服务网络进行分类，例如，按用途、所有者或环境。

您可以为每个服务网络添加多个标签。每个服务网络的标签键必须是唯一的。如果您所添加标签中的键已经与服务网络关联，则会更新该标签的值。可以使用字母、空格、数字（UTF-8）等字符，以及以下特殊字符：+ - =。_ : / @。请不要使用前导空格或尾随空格。标签值区分大小写。

要使用控制台添加或删除标签

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务网络。
3. 选择服务网络名称以打开其详细信息页面。
4. 选择标签选项卡。
5. 要添加标签，请选择添加标签，然后输入标签键和标签值。要添加其他标签，请选择添加新标签。添加完标签后，选择 Save changes（保存更改）。
6. 要删除标签，请选择标签的复选框，然后选择删除。提示进行确认时，输入 **`confirm`**，然后选择删除。

要使用添加或删除标签 AWS CLI

使用 tag-resource 和 untag-resource 命令。

删除 VPC 莱迪思服务网络

在删除服务网络之前，必须先删除该服务网络可能与任何服务、资源配置、VPC 或 VPC 终端节点的所有关联。删除服务网络时，我们还会删除与服务网络相关的所有资源，例如资源策略、验证策略和访问日志订阅。

要使用控制台删除服务网络

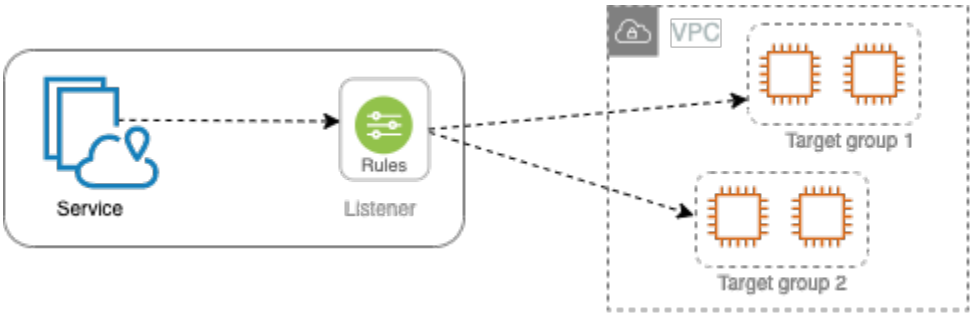
1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务网络。
3. 选中服务网络的复选框，然后依次选择操作和删除服务网络。
4. 提示进行确认时，输入 **`confirm`**，然后选择 Delete（删除）。

要使用删除服务网络 AWS CLI

使用 delete-service-network 命令。

VPC Lattice 中的服务

VPC Lattice 中的服务是一个可独立部署的软件单元，用于交付特定的任务或函数。服务可以在实例、容器上运行，也可以在账户或虚拟私有云（VPC）中作为无服务器函数运行。服务有一个监听器，它使用称为侦听器规则的规则，您可以配置这些规则来帮助将流量路由到目标。支持的目标类型包括 EC2 实例、IP 地址、Lambda 函数、应用程序负载均衡器、Amazon ECS 任务和 Kubernetes Pod。有关更多信息，请参阅 VPC Lattice 中的目标组。您可以将服务与多个服务网络关联。下图显示 VPC Lattice 中典型服务的关键组件。



您可以通过为服务提供名称和描述来创建服务。但是，要控制和监控您的服务流量，务必包含访问设置和监控详细信息。要将流量从服务发送到目标，必须设置侦听器并配置规则。要让流量从服务网络流向服务，必须将您的服务与服务网络关联。

对于目标连接，存在空闲超时和总体连接超时。空闲连接超时为 1 分钟，之后会关闭连接。最长持续时间为 10 分钟，之后我们不允许连接上有新的流，同时开始关闭现有流的过程。

任务

- 步骤 1：创建 VPC Lattice 服务
- 步骤 2：定义路由
- 步骤 3：创建网络关联
- 步骤 4：审核并创建
- 管理 VPC Lattice 服务的关联
- 编辑 VPC Lattice 服务的访问设置
- 编辑 VPC Lattice 服务的监控详细信息
- 管理 VPC Lattice 服务的标签
- 为您的 VPC 莱迪思服务配置自定义域名
- VPC Lattice 自带证书（BYOC）

- 删除 VPC 莱迪思服务

步骤 1：创建 VPC Lattice 服务

创建具有访问设置和监控详细信息的基本 VPC Lattice 服务。但是，在定义路由配置并将其与服务网络关联之前，服务不会完全正常运行。

要使用控制台创建基本服务

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择 Create service。
4. 对于标识符，执行以下操作：

   a. 输入服务的名称。名称必须介于 3-40 个字符之间，并使用小写字母、数字和连字符。必须以字母或数字开头和结尾。不要使用双连字符。

   b. （可选）输入服务网络的描述。您可以在创建过程中或创建后设置或更改描述。描述最多可包含 256 个字符。
5. 要为您的服务指定自定义域名，请选择指定自定义域名配置并输入自定义域名。

   对于 HTTPS 侦听器，您可以选择 VPC 莱迪思用于执行 TLS 终止的证书。如果您现在不选择证书，则可以在为服务创建 HTTPS 侦听器时选择该证书。

   对于 TCP 监听器，您必须为服务指定自定义域名。如果您指定证书，则不使用该证书。相反，您可以在应用程序中执行 TLS 终止。
6. 对于服务访问权限，如果您希望与服务网络 VPCs 关联的客户端访问您的服务，请选择无。要应用验证策略来控制对服务的访问，请选择 AWS IAM。要将资源策略应用于服务，请对验证策略执行以下操作之一：
   - 在输入字段中输入策略。对于可以复制和粘贴的示例策略，请选择策略示例。
   - 选择应用策略模板，然后选择允许已验证和未验证访问模板。此模板允许来自其他账户的客户端通过签署请求（表示已验证），或以匿名方式（表示未验证）访问服务。
   - 选择应用策略模板，然后选择允许仅限已验证访问模板。此模板允许来自其他账户的客户端通过签署请求（表示已验证）访问服务。
7. （可选）要启用访问日志，请开启访问日志切换开关，并按如下方式指定访问日志的目标：
   - 选择CloudWatch 日志组，然后选择一个 CloudWatch 日志组。要创建日志组，请选择在中创建日志组 CloudWatch。

- 选择 S3 存储桶并输入 S3 存储桶路径，包括任何前缀。要搜索 S3 存储桶，请选择浏览 S3。
- 选择 Kinesis Data Firehose 传输流，然后选择一个传输流。要创建传输流，请选择在 Kinesis 中创建传输流。

8. （可选）要与其他账户共享您的服务，请从 AWS RAM 资源共享中选择一个资源共享。要创建资源共享，请选择在 RAM 控制台中创建资源共享。
9. 要查看配置并创建服务，请选择跳转到查看和创建。否则，请选择下一步，定义服务的路由配置。

步骤 2：定义路由

使用侦听器定义路由配置，以便服务可以将流量发送到您指定的目标。

先决条件

在添加侦听器之前，必须先创建 VPC Lattice 目标组。有关更多信息，请参阅 the section called “创建目标组”。

要使用控制台为服务定义路由

1. 选择添加侦听器。
2. 对于侦听器名称，您可以提供自定义侦听器名称，也可以使用侦听器的协议和端口作为侦听器名称。您指定的自定义名称最多可包含 63 个字符，且对账户中的每项服务必须是唯一的。有效字符：a-z、0-9 和连字符（-）。不能将连字符用作第一个或最后一个字符，也不能紧跟在另一个连字符之后。创建侦听器后，不能更改其名称。
3. 选择协议，然后输入端口号。
4. 对于默认操作，选择要接收流量的 VPC Lattice 目标组，然后选择要分配给该目标组的权重。您可以选择为默认操作添加另一个目标组。选择添加操作，然后选择其他目标组并指定其权重。
5. （可选）要添加其他规则，请选择添加规则，然后输入规则的名称、优先级、条件和操作。

   您可以为每条规则指定一个 1 到 100 之间的优先级编号。侦听器不能具有优先级相同的多个规则。规则是按优先级顺序 (从最低值到最高值) 计算的。最后评估默认规则。

   对于条件，输入路径匹配条件的路径模式。每个字符串的最大长度为 200 个字符。比较不区分大小写。
6. （可选）要添加标签，请展开侦听器标签，选择添加新标签，然后输入标签键和标签值。
7. 要查看配置并创建服务，请选择跳转到查看和创建。否则，请选择下一步，将服务与服务网络关联。

步骤 3：创建网络关联

将服务与服务网络关联，以便客户端可以与之通信。

要使用控制台将服务与服务网络关联

1. 对于 VPC Lattice 服务网络，请选择服务网络。要创建服务网络，请选择创建 VPC Lattice 网络。您可以将您的服务与多个服务网络关联。
2. （可选）要添加标签，请展开服务网络关联标签，选择添加新标签，然后输入标签键和标签值。
3. 选择下一步。

步骤 4：审核并创建

要使用控制台查看配置并创建服务

1. 查看服务的配置。
2. 如果需要修改服务配置的任何部分，请选择编辑。
3. 查看或编辑配置后，选择创建 VPC Lattice 服务。
4. 如果为服务指定了自定义域名，则必须在创建服务后配置 DNS 路由。有关更多信息，请参阅 the section called “配置自定义域名”。

管理 VPC Lattice 服务的关联

将服务与服务网络关联后，客户端（与服务网络关联的 VPC 中的资源）就可以向该服务发出请求。您可以关联您账户中的服务，或关联不同账户与您共享的服务。创建服务时，此步骤是可选的。但是，在创建后，只有将服务与服务网络关联，该服务才能与其他服务通信。如果服务所有者的账户有必要的访问权限，就可以将自己的服务与服务网络关联。有关更多信息，请参阅 VPC Lattice 的工作原理。

要使用控制台管理服务网络关联

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择服务名称以打开其详细信息页面。
4. 选择服务网络关联选项卡。
5. 要创建关联，请执行以下操作：

a. 选择创建关联。
b. 从 VPC Lattice 服务网络中选择一个服务网络。要创建服务网络，请选择创建 VPC Lattice 网络。
c. （可选）要添加标签，请展开服务关联标签，选择添加新标签，然后输入标签键和标签值。
d. 选择保存更改。

6. 要删除关联，请选中关联的复选框，然后依次选择操作和删除网络关联。提示进行确认时，输入 **`confirm`**，然后选择删除。

要使用创建服务网络关联 AWS CLI

使用 create-service-network-service-association 命令。

要删除服务网络关联，请使用 AWS CLI

使用 delete-service-network-service-association 命令。

编辑 VPC Lattice 服务的访问设置

访问设置使您能够配置和管理客户端对服务的访问。访问设置包括验证类型和验证策略。验证策略可帮助您验证和授权流向 VPC Lattice 服务的流量。

您可以在服务网络级别、服务级别，或同时在两个级别应用验证策略。在服务级别，服务所有者可以应用细粒度控制，这种控制可能更具限制性。通常，验证策略由网络所有者或云管理员应用。这些策略可以实现粗粒度授权，例如，允许来自组织内部已验证的调用，或者允许匹配特定条件的匿名 GET 请求。有关更多信息，请参阅 使用身份验证策略控制对VPC莱迪思服务的访问。

要使用控制台添加或更新访问策略

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择服务名称以打开其详细信息页面。
4. 选择访问选项卡以查看当前的访问设置。
5. 要更新访问设置，请选择编辑访问设置。
6. 如果您希望关联服务网络 VPCs 中的客户端访问您的服务，请为身份验证类型选择无。
7. 要应用资源策略来控制对服务的访问，则对于验证类型，选择 AWS IAM，并对验证策略执行以下操作之一：

- 在输入字段中输入策略。对于可以复制和粘贴的示例策略，请选择策略示例。
- 选择应用策略模板，然后选择允许已验证和未验证访问模板。此模板允许来自其他账户的客户端通过签署请求（表示已验证），或以匿名方式（表示未验证）访问服务。
- 选择应用策略模板，然后选择允许仅限已验证访问模板。此模板允许来自其他账户的客户端通过签署请求（表示已验证）访问服务。

8. 选择保存更改。

要添加或更新访问策略，请使用 AWS CLI

使用 put-auth-policy 命令。

编辑 VPC Lattice 服务的监控详细信息

VPC Lattice 会为每个请求和响应生成指标和日志，从而提高了监控和排查应用程序问题的效率。

您可以启用访问日志，并为日志指定目标资源。VPC Lattice 可以将日志发送到以下资源：CloudWatch 日志组、Firehose 传输流和 S3 存储桶。

要使用控制台启用访问日志或更新日志目标

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择服务名称以打开其详细信息页面。
4. 选择监控选项卡，然后选择日志。检查访问日志，查看访问日志是否已启用。
5. 要启用或禁用访问日志，请选择编辑访问日志，然后打开或关闭访问日志切换开关。
6. 在启用访问日志时，必须选择传输目标的类型，然后创建或选择访问日志的目标。您还可以随时更改传输目标。例如：
   - 选择CloudWatch 日志组，然后选择一个 CloudWatch 日志组。要创建日志组，请选择在中创建日志组 CloudWatch。
   - 选择 S3 存储桶并输入 S3 存储桶路径，包括任何前缀。要搜索 S3 存储桶，请选择浏览 S3。
   - 选择 Kinesis Data Firehose 传输流，然后选择一个传输流。要创建传输流，请选择在 Kinesis 中创建传输流。
7. 选择保存更改。

要启用访问日志，请使用 AWS CLI

使用 create-access-log-subscription 命令。

要更新日志目标，请使用 AWS CLI

使用 update-access-log-subscription 命令。

要禁用访问日志，请使用 AWS CLI

使用 delete-access-log-subscription 命令。

管理 VPC Lattice 服务的标签

标签有助于您以不同的方式对服务进行分类，例如，按用途、所有者或环境。

您可以为每项服务添加多个标签。每项服务的标签密钥必须是唯一的。如果您添加的标签的密钥已与服务关联，则它会更新该标签的值。可以使用字母、空格、数字（UTF-8）等字符，以及以下特殊字符：+ - =。_ : / @。请不要使用前导空格或尾随空格。标签值区分大小写。

要使用控制台添加或删除标签

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择服务名称以打开其详细信息页面。
4. 选择标签选项卡。
5. 要添加标签，请选择添加标签，然后输入标签键和标签值。要添加其他标签，请选择添加新标签。添加完标签后，选择 Save changes（保存更改）。
6. 要删除标签，请选择标签的复选框，然后选择删除。提示进行确认时，输入 **`confirm`**，然后选择删除。

要使用添加或删除标签 AWS CLI

使用 tag-resource 和 untag-resource 命令。

为您的 VPC 莱迪思服务配置自定义域名

当您创建新服务时，VPC Lattice 会使用以下语法为该服务生成一个唯一的完全限定域名 (FQDN)。

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

但是，VPC Lattice提供的域名并不容易让您的用户记住。您可以向用户提供自定义域名更简单、更直观 URLs 。如果您希望为服务使用自定义域名，例如 www.parking.example.com 而不是 VPC Lattice 生成的 DNS 名称，可以在创建 VPC Lattice 服务时进行配置。当客户端使用您的自定义域名发出请求时，DNS 服务器会将其解析为 VPC Lattice 生成的域名。

先决条件

- 您的服务必须有一个注册域名。如果您还没有注册域名，可以通过 Amazon Route 53，或任何其他商业注册商来注册一个域名。
- 要接收 HTTPS 请求，必须在 AWS Certificate Manager中提供自己的证书。VPC Lattice 不支持默认证书作为后备。因此，如果您不提供与您的自定义域名相对应的 SSL/TLS 证书，则与您的自定义域名的所有 HTTPS 连接都将失败。有关更多信息，请参阅 VPC Lattice 自带证书（BYOC）。

限制和注意事项

- 一项服务不能有多个自定义域名。
- 创建服务后，您无法修改自定义域名。
- 自定义域名对于服务网络必须是唯一的。这意味着无法使用同一服务网络中已存在的自定义域名（针对其他服务）创建服务。

以下过程说明如何为您的服务配置自定义域名。

AWS 管理控制台

为您的服务配置自定义域名

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择创建服务。您将导航到步骤 1：创建服务。
4. 在自定义域配置部分，选择指定自定义域配置。
5. 输入您的自定义域名。
6. 要处理 HTTPS 请求，请在自定义 SSL/TLS 证书中选择与您的自定义域名相匹配的 SSL/TLS 证书。如果您还没有证书，或者现在不想添加证书，可以在创建 HTTPS 侦听器时添加证书。

但是，如果没有证书，您的自定义域名将无法提供 HTTPS 请求。有关更多信息，请参阅 添加 HTTPS 侦听器。

7. 添加用于创建服务的所有其他信息后，选择创建。

AWS CLI

为您的服务配置自定义域名

使用 create-service 命令。

```
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
```

在上面的命令中，对于 `--name`，输入服务的名称。对于 `--custom-domain-name`，输入服务的域名，例如 `parking.example.com`。对于 `--certificate-arn`，在 ACM 中输入证书的 ARN。证书 ARN 可在您的 AWS Certificate Manager账户中找到。

将自定义域名与您的服务关联

首先，如果尚未注册域名，请注册您的自定义域名。Internet 上的域名由 Internet 名称和数字地址分配机构 (ICANN) 管理。您需要通过域名注册商 (ICANN 认可的管理域名注册的组织) 注册域名。您的注册商的网站上会提供关于注册域名的详细说明和定价信息。有关更多信息，请参阅以下资源：

- 要使用 Amazon Route 53 注册域名，请参阅 Amazon Route 53 开发人员指南中的使用 Route 53 注册域名。
- 有关获得认可的注册商的列表，请参阅获得认可的注册商目录。

接下来，使用您的 DNS 服务（例如域名注册商）创建记录，将查询路由到您的服务。有关更多信息，请参阅您的 DNS 服务的文档。或者，您可以使用 Route 53 作为您的 DNS 服务。

如果您使用的是 Route 53，则可以使用别名记录或 CNAME 记录将查询路由到您的服务。我们建议您使用别名记录，因为您可以在 DNS 命名空间的顶级节点（也称为区域顶点）创建别名记录。

如果您使用的是 Route 53，则必须先创建一个托管区，其中包含有关如何在互联网上为域路由流量的信息。创建私有或公有托管区域后，请创建一条记录，以便将您的自定义域名（例如`parking.example.com`）映射到 VPC Lattice 自动生成的域名。my-

service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws如果没有此映射，您的自定义域名将无法在 VPC Lattice 中使用。

以下过程说明如何使用 Route 53 创建私有或公共托管区域

AWS 管理控制台

要创建别名记录以使用 Route 53 将查询路由到您的服务，请参阅将流量路由到 Amazon VPC Lattice 服务域终端节点。

使用莱迪思为服务生成的VPC域名，my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws例如作为值。您可以在VPC Lattice控制台的服务页面上找到这个自动生成的域名。

AWS CLI

在您的托管区域中创建别名记录

1. 获取莱迪思为您的服务生成的 VPC 域名（例如my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws）。
2. 要设置别名，请使用以下命令。

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

对于 change-set.json 文件，使用以下 JSON 示例中的内容创建一个 JSON 文件，然后将其保存在本地计算机上。将*file://~/Desktop/change-set.json*上面的命令替换为保存在本地计算机中的 JSON 文件的路径。注意，以下 JSON 中的“类型”可以是 A 或 AAAA 记录类型。

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

```
                    "DNSName": "lattice-generated-domain-name",
                    "EvaluateTargetHealth": true
                }
            }
        }
    ]
}
```

VPC Lattice 自带证书（BYOC）

要处理 HTTPS 请求，在设置自定义域名之前，必须准备好自己的 SSL/TLS 证书 AWS Certificate Manager (ACM)。这些证书必须具有与服务的自定义域名匹配的使用者备用名称（SAN），或公用名称（CN）。如果 SAN 存在，我们仅检查 SAN 列表中的匹配项。如果 SAN 不存在，则会检查 CN 中的匹配项。

VPC Lattice 使用服务器名称指示（SNI）提供 HTTPS 请求。DNS 会根据自定义域名和与该域名匹配的证书，将 HTTPS 请求路由到您的 VPC Lattice 服务。要在 ACM 中为域名申请 SSL/TLS 证书或将证书导入 ACM，请参阅AWS Certificate Manager 用户指南中的颁发和管理证书以及导入证书。如果无法在 ACM 中请求或导入自己的证书，请使用 VPC Lattice 生成的域名和证书。

VPC Lattice 每项服务仅接受一个自定义证书。但是，您可以将自定义证书用于多个自定义域。这意味着您可以为使用自定义域名创建的所有 VPC Lattice 服务使用相同的证书。

要使用 ACM 控制台查看证书，请打开证书，然后选择证书 ID。您会在关联资源下看到与该证书关联的 VPC Lattice 服务。

限制和注意事项

- VPC Lattice 允许在关联证书的使用者备用名称（SAN），或公用名称（CN）中进行深一级的通配符匹配。例如，如果您使用自定义域名 `parking.example.com` 创建服务，并将自己的证书与 SAN `*.example.com` 关联。当 `parking.example.com` 收到请求时，VPC Lattice 会将 SAN 与具有 apex 域 `example.com` 的任何域名匹配。但是，如果您拥有自定义域 `parking.different.example.com`，并且您的证书具有 SAN `*.example.com`，则请求将失败。
- VPC Lattice 支持一级通配符域匹配。这意味着通配符只能用作一级子域，并且只能保护一个子域级别。例如，如果证书的 SAN 是 `*.example.com`，则不支持 `parking.*.example.com`。
- VPC Lattice 支持每个域名一个通配符。这意味着 `*.*.example.com` 是无效的。有关更多信息，请参阅《AWS Certificate Manager 用户指南》中的请求公有证书。

- VPC Lattice 仅支持使用 2048 位 RSA 密钥的证书。
- ACM 中的 SSL/TLS 证书必须与您关联的 VPC Lattice 服务位于同一区域。

保护证书私有密钥

当您使用 ACM 请求 SSL/TLS 证书时，ACM 会生成一个 public/private key pair。导入证书时，将生成密钥对。公有密钥将成为证书的一部分。为了安全地存储私钥，ACM 使用 AWS KMS别名 a w s/acm 创建了另一个名为 KMS 密钥的密钥。 AWS KMS 使用此密钥来加密证书的私钥。有关更多信息，请参阅《AWS Certificate Manager 用户指南》中的 AWS Certificate Manager中的数据保护。

VPC AWS Lattice使用TLS连接管理器（一项只能访问的服务）来保护和使用您的证书的私钥。 AWS 服务当您使用 ACM 证书创建 VPC 莱迪思服务时，VPC Lattice 会将您的证书与 AWS TLS 连接管理器相关联。为此，我们会 AWS KMS 根据您的 AWS 托管密钥创建授权。此授权允许 TLS 连接管理器 AWS KMS 用于解密证书的私钥。TLS Connection Manager 使用证书和解密（明文）私有密钥，与 VPC Lattice 服务的客户端建立安全连接（SSL/TLS 会话）。当证书与 VPC Lattice 服务取消关联时，该授权就会失效。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的授权。

有关更多信息，请参阅 静态加密。

删除 VPC 莱迪思服务

要删除 VPC Lattice 服务，必须先删除该服务与任何服务网络的所有关联。如果删除服务，则与该服务相关的所有资源也会被删除，例如资源策略、验证策略、侦听器、侦听器规则和访问日志订阅。

要使用控制台删除服务

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 在服务页面上，选择要删除的服务，然后依次选择操作和删除服务。
4. 当系统提示进行确认时，选择 Delete（删除）。

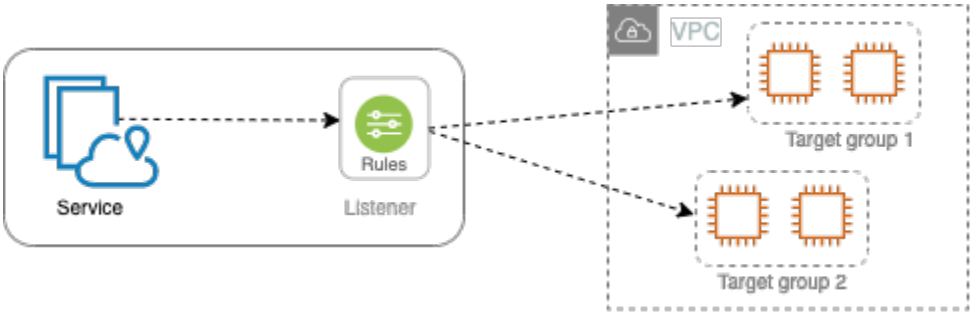
要删除服务，请使用 AWS CLI

使用 delete-service 命令。

VPC Lattice 中的目标组

VPC Lattice 目标组是运行应用程序或服务的目标或计算资源的集合。支持的目标类型包括 EC2 实例、IP 地址、Lambda 函数、应用程序负载均衡器、Amazon ECS 任务和 Kubernetes Pod。您还可以将现有服务附加到目标组。有关将 Kubernetes 与 VPC Lattice 结合使用的更多信息，请参阅 AWS 网关 API 控制器用户指南。

每个目标组均用于将请求路由到一个或多个已注册的目标。在创建侦听器规则时，您可以指定目标组和条件。满足规则条件时，流量会转发到相应的目标组。您可以为不同类型的请求创建不同的目标组。例如，为一般请求创建一个目标组，为包含特定规则条件（如路径或标头值）的请求创建其他目标组。



您可以根据目标组来定义服务的运行状况检查设置。每个目标组均使用默认运行状况检查设置，除非您在创建目标组时将其覆盖或稍后对其进行修改。在侦听器的规则中指定目标组后，服务将持续监控向该目标组注册的所有目标的运行状况。服务将请求路由到运行正常的已注册目标。

要在规则中为服务侦听器指定目标组，目标组必须与服务位于同一账户中。

VPC Lattice 目标组与弹性负载均衡提供的目标组类似，但不可互换。

内容

- VPC Lattice 中的应用程序负载均衡器作为目标
- 协议版本
- VPC Lattice 目标组的标签
- 删除 VPC 莱迪思目标组

创建 VPC Lattice 目标组

将目标注册到目标组。默认情况下，VPC Lattice 服务使用您为目标组指定的端口和协议向已注册的目标发送请求。在将每个目标注册到目标组时，可以覆盖此端口。

要将流量路由到目标组中的目标，请在创建侦听器或侦听器规则时，在操作中指定目标组。有关更多信息，请参阅 VPC Lattice 服务的侦听器规则。您可以在多个侦听器中指定同一个目标组，但这些侦听器必须属于同一服务。要在服务中使用目标组，必须确认目标组未被任何其他服务的侦听器使用。

您可以随时在目标组中添加或删除目标。有关更多信息，请参阅 向 VPC Lattice 目标组注册目标。您也可以修改目标组的运行状况检查设置。有关更多信息，请参阅 VPC Lattice 目标组的运行状况检查。

创建目标组

您可以创建目标组，也可以按如下方式注册目标。

使用控制台创建目标组

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择目标组。
3. 选择创建目标组。
4. 对于选择目标类型，请执行以下操作之一：
   - 选择实例，按实例 ID 注册目标。
   - 选择 IP 地址，按 IP 地址注册目标。
   - 选择 Lambda 函数，将 Lambda 函数注册为目标。
   - 选择应用程序负载均衡器，将应用程序负载均衡器注册为目标。
5. 对于目标组名称，输入目标组的名称。该名称对于您的账户在每个 AWS 地区都必须是唯一的，最多可包含 32 个字符，必须仅包含字母数字字符或连字符，并且不得以连字符开头或结尾。
6. 对于协议和端口，可以根据需要修改默认值。默认协议为 HTTPS，默认端口为 443。

   如果目标类型是 Lambda 函数，则无法指定协议或端口。

7. 对于 IP 地址类型 IPv4，选择使用 IPv4 地址注册目标或选择IPv6使用 IPv6 地址注册目标。创建目标组后，无法更改此设置。

   仅当目标类型为 IP 地址时，此选项才可用。

8. 对于 VPC，选择 Virtual Private Cloud (VPC)。

   如果目标类型是 Lambda 函数，此选项不可用。

9. （可选）对于协议版本，根据需要修改默认值。默认值为 HTTP1。

   如果目标类型是 Lambda 函数，此选项不可用。

10. 对于运行状况检查，根据需要修改默认设置。有关更多信息，请参阅 VPC Lattice 目标组的运行状况检查。

    如果目标类型是 Lambda 函数，运行状况检查不可用。

11. 对于 Lambda 事件结构版本，请选择一个版本。有关更多信息，请参阅 the section called “从 VPC Lattice 服务接收事件”。

    仅当目标类型为 Lambda 函数时，此选项才可用。

12. （可选）要添加标签，请展开标签，选择添加新标签，然后输入标签键和标签值。

13. 选择下一步。

14. 对于注册目标，您可以跳过此步骤，或按如下方式添加目标：

    - 如果目标类型为实例，请选择实例，输入端口，然后选择在下面以待注册的形式添加。
    - 如果目标类型为 IP addresses (IP 地址)，请执行以下操作：

      a. 对于选择网络，保留您为目标组选择的 VPC 或选择其他私有 IP 地址。

      b. 在 “指定 IPs 和定义端口” 中，输入 IP 地址并输入端口。默认端口为目标组端口。

      c. 选择在下面以待注册的形式添加。

    - 如果目标类型是 Lambda 函数，请选择 Lambda 函数。要创建 Lambda 函数，请选择创建新的 Lambda 函数。
    - 如果目标类型是应用程序负载均衡器，请选择应用程序负载均衡器。要创建应用程序负载均衡器，请选择创建应用程序负载均衡器。

15. 选择创建目标组。

    VPC Lattice 可能需要几分钟才能注册目标。有关更多信息，请参阅 “为什么我的 DNS 更改需要这么长时间才在 Route 53 和公共解析器中传播？

要使用创建目标组 AWS CLI

使用create-target-group命令创建目标组，使用 regi ster-targets 命令添加目标。

共享子网

参与者可以在共享 VPC 中创建 VPC Lattice 目标组。以下规则适用于共享子网：

- VPC Lattice 服务的所有部分（如侦听器、目标组和目标）必须由同一账户创建。可以在 VPC Lattice 服务所有者拥有或共享的子网中创建。
- 向目标组注册的目标必须由与目标组相同的账户创建。
- 只有 VPC 的所有者才能将 VPC 与服务网络关联。与服务网络关联的共享 VPC 中的参与者资源，可以向与服务网络关联的服务发送请求。但是，管理员可以通过使用安全组 ACLs、网络或身份验证策略来阻止这种情况。

有关 VPC Lattice 可共享资源的更多信息，请参阅 共享 VPC 莱迪思实体。

向 VPC Lattice 目标组注册目标

您的服务充当客户端的单一联系点，并在其正常运行的注册目标之间分配传入流量。您可以将每个目标注册到一个或多个目标组中。

如果对应用程序的需求增加，则可以向一个或多个目标组注册其他目标来处理需求。一旦注册过程完成，并且目标通过初始运行状况检查，服务就会开始将请求路由到新注册的目标。

如果应用程序需求减少或者您需要为目标提供服务，您可以从目标组取消注册目标。取消注册目标将从目标组中删除目标，但不会影响目标。取消注册后，服务会停止将请求路由到目标。目标将进入 `DRAINING` 状态，直至进行中请求完成。在您准备好目标以继续接收请求时，可以重新将目标注册到目标组。

您的目标组的目标类型将确定如何向该目标组注册目标。有关更多信息，请参阅 Target type。

使用以下控制台过程来注册或取消注册目标。或者，使用 AWS CLI中的 register-targets 和 deregister-targets 命令。

内容

- 通过实例 ID 注册或取消注册目标
- 通过 IP 地址注册或取消注册目标

- 注册或注销 Lambda 函数
- 注册或取消注册应用程序负载均衡器

通过实例 ID 注册或取消注册目标

目标实例必须位于您为目标组指定的虚拟私有云（VPC）中。当您注册实例时，实例还必须处于 `running` 状态。

按实例 ID 注册目标时，可以将服务与自动扩缩组结合使用。将目标组附加到自动扩缩组并且该组横向扩展后，自动扩缩组启动的实例将自动注册到目标组。如果您将目标组与 Auto Scaling 组分离，则实例会自动从目标组中取消注册。有关更多信息，请参阅《Amazon EC2 Auto Scaling 用户指南》中的使用 VPC Lattice 目标组将流量路由到自动扩缩组。

使用控制台按实例 ID 注册或取消注册目标

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 选择目标选项卡。
5. 要注册实例，请选择注册目标。选择实例，输入实例端口，然后选择包括以下待定内容。添加实例后，选择注册目标。
6. 要取消注册实例，请选择实例，然后选择取消注册。

通过 IP 地址注册或取消注册目标

目标 IP 地址必须来自您为目标组指定的 VPC 子网。不能在同一 VPC 中注册其他服务的 IP 地址。不能注册 VPC 端点或公开可路由 IP 地址。

使用控制台按 IP 地址注册或取消注册目标

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 选择 Targets (目标) 选项卡。
5. 要注册 IP 地址，请选择注册目标。对于每个 IP 地址，选择网络，输入 IP 地址和端口，然后选择在下面以待注册的形式添加。指定地址后，选择注册目标。

6. 要注销 IP 地址，请选择 IP 地址，然后选择取消注册。

注册或注销 Lambda 函数

您可以向目标组注册单个 Lambda 函数。如果您不再需要向您的 Lambda 函数发送流量，则可以将其取消注册。在取消注册 Lambda 函数后，进行中的请求会失败，并显示 HTTP 5XX 错误。最好是创建一个新的目标组，而不是替换目标组的 Lambda 函数。

要使用新控制台注册或取消注销 Lambda 函数

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 选择目标选项卡。
5. 如果未注册任何 Lambda 函数，请选择注册目标。选择 Lambda 函数并选择注册目标。
6. 要取消注册 Lambda 函数，请选择 Deregister (取消注册)。当系统提示确认时，输入 **`confirm`**，然后选择取消注册。

注册或取消注册应用程序负载均衡器

您可以向每个目标组注册一个应用程序负载均衡器。如果您不再需要将流量发送到负载均衡器，则可以取消注册。在取消注册负载均衡器后，进行中的请求会失败，并显示 HTTP 5XX 错误。最好是创建一个新的目标组，而不是替换目标组的应用程序负载均衡器。

要使用控制台注册或取消注册应用程序负载均衡器

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 选择目标选项卡。
5. 如果未注册应用程序负载均衡器，请选择注册目标。选择应用程序负载均衡器，然后选择注册目标。
6. 要取消注册应用程序负载均衡器，请选择取消注册。当系统提示确认时，输入 **`confirm`**，然后选择取消注册。

VPC Lattice 目标组的运行状况检查

服务会定期向注册的目标发送请求，以测试其状态。这些测试称为运行状况检查。

每个 VPC Lattice 服务仅将请求路由到运行状况良好的目标。对于注册目标的目标组，每个服务均使用运行状况检查来检查每个目标的运行状况。在注册目标后，目标必须通过一次运行状况检查才会被视为正常。每次运行状况检查完成后，服务将关闭为运行状况检查建立的连接。

限制和注意事项

- 当目标组协议版本为时 HTTP1，默认情况下会启用运行状况检查。
- 当目标组协议版本为时 HTTP2，默认情况下不启用运行状况检查。但是，您可以启用运行状况检查，并将协议版本手动设置为 HTTP1 或 HTTP2。
- 运行状况检查不支持 gRPC 目标组协议版本。但是，如果您启用运行状况检查，则必须将运行状况检查协议版本指定为 HTTP1 或 HTTP2。
- 运行状况检查不支持 Lambda 目标组。
- 运行状况检查不支持应用程序负载均衡器目标组。但您可以使用弹性负载均衡，为应用程序负载均衡器的目标启用运行状况检查。有关更多信息，请参阅应用程序负载均衡器用户指南中的目标组运行状况检查。

运行状况检查设置

如下表所述，您可以为目标组中的目标配置运行状况检查。表中使用的设置名称是 API 中使用的名称。该服务使用指定的端口、协议和 ping 路径每HealthCheckIntervalSeconds秒向每个注册目标发送一次运行状况检查请求。每个运行状况检查请求都是独立的，其结果在整个时间间隔内持续。目标响应所用时间不影响下一运行状况检查请求的时间间隔。如果运行状况检查超过UnhealthyThresholdCount连续失败次数，则该服务会使目标停止服务。当运行状况检查超过HealthyThresholdCount连续成功率时，服务会将目标重新投入使用。

设置	说明
HealthCheckProtocol	服务在对目标执行运行状况检查时使用的协议。可能的协议为 HTTP 和 HTTPS。默认值为 HTTP 协议。

设置	说明
HealthCheckPort	服务在对目标执行运行状况检查时使用的端口。默认情况下，使用每个目标从服务接收流量的端口。
HealthCheckPath	目标运行状况检查的目的地。 如果协议版本为 HTTP1 或 HTTP2，请指定有效的 URI (/path? 查询）。默认值为 /。
HealthCheckTimeoutSeconds	以秒为单位的时间长度，在此期间内，没有来自目标的响应意味着无法通过运行状况检查。范围为 1-120 秒。如果目标类型为 INSTANCE 或 IP，则默认值为 5 秒。指定 0 将此设置重置为默认值。
HealthCheckIntervalSeconds	各个目标的运行状况检查之间的大约时间量 (以秒为单位)。范围为 5–300 秒。如果目标类型为 INSTANCE 或 IP，则默认值为 30 秒。指定 0 将此设置重置为默认值。
HealthyThresholdCount	运行状况不良的目标变为运行状况正常的目标之前，所需的连续运行状况检查成功次数。范围为 2–10。默认值为 5。指定 0 将此设置重置为默认值。
UnhealthyThresholdCount	将目标视为不正常之前所需的连续运行状况检查失败次数。范围为 2–10。默认值为 2。指定 0 将此设置重置为默认值。

设置	说明
Matcher	检查来自目标的成功响应时要使用的代码。这些代码在控制台中称为成功代码。 如果协议版本为 HTTP1 或 HTTP2，则可能的值介于 200 到 499 之间。您可以指定多个值（例如，“200,202”）或一系列值（例如，“200-299”）。默认值为 200。 目前不支持 gRPC 的运行状况检查协议版本。但是，如果您的目标组协议版本是 gRPC，则可以在运行状况检查配置中指定 HTTP1 或 HTTP2 协议版本。

检查目标的运行状况

您可以检查已注册到目标组的目标的运行状况。

使用控制台检查目标的运行状况

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Targets（目标）选项卡上，Health status（运行状况）列指示每个目标的状态。如果状态为 `Healthy` 以外的任何值，则运行状况状态详细信息列将包含更多信息。

要检查目标的生命值，请使用 AWS CLI

使用 list-targets 命令。此命令的输出包含目标运行状况。如果状态是 `Healthy` 以外的任何值，则输出还包括原因代码。

接收有关运行状况不佳的目标的电子邮件通知

使用 CloudWatch 警报启动 Lambda 函数以发送有关不健康目标的详细信息。

修改运行状况检查设置

您可以随时修改目标组的运行状况检查设置。

要使用控制台修改运行状况检查设置

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在运行状况检查选项卡的运行状况检查设置部分，选择编辑。
5. 根据需要修改运行状况检查设置。
6. 选择保存更改。

要修改运行状况检查设置，请使用 AWS CLI

使用 update-target-group 命令。

路由配置

默认情况下，服务会使用您在创建目标组时指定的协议和端口号，将请求路由到目标。此外，您可以覆盖在将目标注册到目标组时用于将流量路由到目标的端口。

目标组支持以下协议和端口：

- 协议：HTTP、HTTPS、TCP
- 端口：1-65535

如果目标组配置了 HTTPS 协议或使用 HTTPS 运行状况检查，则与目标的 TLS 连接将使用来自监听器的安全策略。VPC Lattice 使用您在目标上安装的证书与目标建立 TLS 连接。VPC 莱迪思不验证这些证书。因此，您可以使用自签名证书或已过期的证书。VPC Lattice和目标之间的流量在数据包级别进行身份验证，因此即使目标上的证书无效，也不会面临 man-in-the-middle攻击或欺骗的风险。

只有 T LS 侦听器支持 T CP 目标组。

路由算法

默认情况下，使用循环路由算法将请求路由到运行状况良好的目标。

当 VPC Lattice 服务收到请求时，会使用以下流程：

1. 按优先级顺序评估侦听器规则以确定要应用的规则。
2. 使用默认的循环算法，从规则操作的目标组中选择一个目标。每个目标组的路由都是单独进行的，即使某个目标已在多个目标组中注册。

如果目标组仅包含运行状况不良的注册目标，则会将请求路由到所有目标，而不考虑其运行状况如何。这意味着，如果所有目标同时未通过运行状况检查，则 VPC Lattice 服务将故障打开。故障打开的作用是根据轮循算法，允许流量进入所有目标，无论其运行状况如何。

VPC Lattice 支持可用区 (AZ) 关联性来路由流量。当客户端向 VPC Lattice 发送请求时，VPC Lattice 会使用与客户端位于同一可用区的服务或资源的 IP 地址进行响应。如果该可用区不可用，VPC Lattice 将使用其他 AZs可用区的 IP 地址进行响应。从 VPC Lattice 到目标，路由是指向可能分布在各 AZs处的目标。此外，VPC Lattice 不收取可用区间数据传输费用。

Target type

创建目标组时，指定其目标类型，此类型将确定您在向此目标组注册目标时指定的目标的类型。创建目标组后，您无法更改其目标类型。

以下是可能的目标类型：

`INSTANCE`

这些目标通过实例 ID 指定。

`IP`

目标是 IP 地址。

`LAMBDA`

目标是 Lambda 函数。

`ALB`

目标是应用程序负载均衡器。

注意事项

- 如果目标类型为 `IP`，则必须为目标组指定来自 VPC 子网的 IP 地址。如果需要从此 VPC 外部注册 IP 地址，请创建类型为 `ALB` 的目标组，并向应用程序负载均衡器注册 IP 地址。

- 如果目标类型为 IP，则无法注册 VPC 端点或公开可路由 IP 地址。
- 如果目标类型为 LAMBDA，则可以注册单个 Lambda 函数。当服务收到对 Lambda 函数的请求时，将会调用 Lambda 函数。如果要向服务注册多个 lambda 函数，则需要使用多个目标组。
- 当目标类型为时ALB，您可以将单个内部 Application Load Balancer 注册为最多两个 VPC Lattice 服务的目标。为此，请向两个单独的目标组注册应用程序负载均衡器，这两个目标组用于两个不同的 VPC Lattice 服务。此外，目标应用程序负载均衡器必须至少有一个侦听器，其端口与目标组端口匹配。
- 您可以在启动时自动向VPC Lattice目标组注册您的ECS任务。该目标组必须有一个目标类型 IP。有关更多信息，请参阅亚马逊弹性容器服务开发人员指南中的将 VPC Lattice 与您的 Amazon ECS 服务配合使用。

  或者，使用类型ALB为 VPC Lattice 目标群组的 Amazon ECS 服务注册应用程序负载均衡器。有关更多信息，请参阅《亚马逊弹性容器服务开发人员指南》中的使用负载平衡来分配 Amazon ECS 服务流量。
- 要将 EKS 容器组注册为目标，请使用 AWS 网关 API 控制器，该控制器从 Kubernetes 服务获取 IP 地址。
- 如果目标组协议是 TCP，则仅支持的目标类型是INSTANCEIP、或ALB。

IP 地址类型

当您创建目标类型为 IP 的目标组时，您可以为目标组指定 IP 地址类型。此操作指定负载均衡器使用何种类型的地址向目标发送请求，以及运行状况检查。可能的值为 IPv4 和 IPv6。默认为 IPV4。

注意事项

- 如果您创建的 IP 地址类型为的目标组IPv6，则您为该目标组指定的 VPC 必须具有 IPv6 地址范围。
- 向目标组注册的 IP 地址必须与目标组的 IP 地址类型匹配。例如，如果目标组的 IP IPv6 地址类型为，则无法在目标组中注册该地址IPv4。
- 向目标组注册的 IP 地址必须在您为目标组指定 VPC 的 IP 地址范围内。

VPC Lattice 中的 HTTP 目标

HTTP 请求和 HTTP 响应使用标头字段发送有关 HTTP 消息的信息。HTTP 标头会自动添加。标头字段为冒号分隔的名称值对，各个值对之间由回车符 (CR) 和换行符 (LF) 进行分隔。RFC 2616 信息标

头中定义了标准 HTTP 标头字段集。此外还有应用程序广泛使用和自动添加的非标准 HTTP 标头。例如，有一些带有 x-forwarded 前缀的非标准 HTTP 标头。

x-forwarded 标头

Amazon VPC Lattice 添加了以下 x-forwarded 标头：

x-forwarded-for

源 IP 地址。

x-forwarded-port

目标端口。

x-forwarded-proto

网络连接协议（http | https）。

调用方身份标头

Amazon VPC Lattice 添加了以下调用方身份标头：

x-amzn-lattice-identity

身份信息。如果 AWS 验证成功，则会显示以下字段。

- Principal：经过验证的主体。
- PrincipalOrgID：经过验证主体的组织 ID。
- PrincipalOrgPath— 经过身份验证的委托人的组织路径。
- SessionName：经过验证会话的名称。

如果使用 Roles Anywhere 凭证且验证成功，则会显示以下字段。

- X509Issuer/OU：发布者（OU）。
- X509SAN/DNS：使用者备用名称（DNS）。
- X509SAN/NameCN：发布者备用名称（名称/CN）。
- X509SAN/URI：使用者备用名称（URI）。
- X509Subject/CN：使用者名称（CN）。

x-amzn-lattice-identity-tags

委托人 ID 和任何委托人标签。格式如下所示。

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

VPC Lattice 使用反斜杠 (\) 对值中的任何分号 (;) 进行转义。

`x-amzn-lattice-network`

VPC。格式如下所示。

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

`x-amzn-lattice-target`

目标。格式如下所示。

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

有关 VPC Lattice 资源 ARNs 的信息，请参阅 Amazon VPC Lattice 定义的资源类型。

来电者身份标头不能被伪造。VPC Lattice 会从任何传入的请求中删除这些标头。这些身份标头使用以下格式表示支持空值的地图。解析时，不应依赖这些标头 KEYs中的特定顺序，您应该期望随时都会添加新的 KEYs 标题，并且您应该做好处理空值的准备。

格式如下所示。

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

Lambda 函数作为 VPC Lattice 中的目标

您可以将 Lambda 函数注册为 VPC Lattice 目标组的目标，并配置侦听器规则，将请求转发到 Lambda 函数的目标组。当服务将请求转发到以 Lambda 函数为目标的目标组时，则会调用 Lambda 函数，并以 JSON 格式将请求内容传递给 Lambda 函数。

限制

- Lambda 函数和目标组必须位于同一账户中，且位于同一区域中。
- 您可以发送到 Lambda 函数的请求正文最大大小为 6MB。
- Lambda 函数可以发送的响应 JSON 最大大小为 6MB。
- 协议必须是 HTTP 或 HTTPS。

准备 Lambda 函数

如果您将 Lambda 函数与 VPC Lattice 服务结合使用，则以下建议适用。

调用 Lambda 函数的权限

当您创建目标组并使用 AWS 管理控制台 或注册 Lambda 函数时， AWS CLI VPC Lattice 会代表您向您的 Lambda 函数策略添加所需的权限。

您还可以使用以下 API 调用自行添加权限：

```
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 函数版本控制

您可以为每个目标组注册一个 Lambda 函数。为确保您可以更改 Lambda 函数，并确保 VPC Lattice 服务始终调用当前版本的 Lambda 函数，请创建函数别名，然后在向 VPC Lattice 服务注册 Lambda 函数时，在函数 ARN 中包含该别名。有关更多信息，请参阅开发人员指南中的 Lambda 函数版本和为 Lambda 函数创建别名。AWS Lambda

为 Lambda 函数创建目标组

创建一个要在请求路由中使用的目标组。如果请求内容与侦听器规则匹配，并执行将其转发到此目标组的操作，则 VPC Lattice 服务将调用注册的 Lambda 函数。

要使用控制台创建目标组并注册 Lambda 函数

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择目标组。
3. 选择创建目标组。
4. 对于选择目标类型，选择 Lambda 函数。
5. 对于目标组名称，输入目标组的名称。
6. 对于 Lambda 事件结构版本，请选择一个版本。有关更多信息，请参阅 the section called “从 VPC Lattice 服务接收事件”。

7. （可选）要添加标签，请展开标签，选择添加新标签，然后输入标签键和标签值。
8. 选择下一步。
9. 对于 Lambda function (Lambda 函数)，请执行下列操作之一：
   - 选择现有 Lambda 函数。
   - 创建新的 Lambda 函数并选中。
   - 稍后注册 Lambda 函数。
10. 选择创建目标组。

要创建目标组并使用注册 Lambda 函数 AWS CLI

使用create-target-group和注册目标命令。

从 VPC Lattice 服务接收事件

VPC Lattice 服务支持通过 HTTP 和 HTTPS 调用 Lambda 请求。该服务以 JSON 格式发送事件，并将 `X-Forwarded-For` 标头添加到每个请求中。

Base64 编码

如果 `content-encoding` 标头存在且内容类型不是以下类型之一，则服务对正文进行 Base64 编码：

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

如果 `content-encoding` 标头不存在，Base64 编码取决于内容类型。对于上述内容类型，服务按原样发送正文，不进行 Base64 编码。

事件结构格式

在创建或更新类型为 `LAMBDA` 的目标组时，您可以指定 Lambda 函数接收的事件结构的版本。可能的版本是 `V1` 和 `V2`。

Example示例事件：V2

```
{
```

```
    "version": "2.0",
    "path": "/?query1=value1&query2=value2",
    "method": "GET|POST|HEAD|...",
    "headers": {
        "header-key": ["header-value", ...],
        ...
    },
    "queryStringParameters": {
        "key": ["value", ...]
    },
    "body": "request-body",
    "isBase64Encoded": true|false,
    "requestContext": {
        "serviceNetworkArn": "arn:aws:vpc-
lattice:region:123456789012:servicenetwork/sn-0bf3f2882e9cc805a",
        "serviceArn": "arn:aws:vpc-
lattice:region:123456789012:service/svc-0a40eebed65f8d69c",
        "targetGroupArn": "arn:aws:vpc-
lattice:region:123456789012:targetgroup/tg-6d0ecf831eec9f09",
        "identity": {
            "sourceVpcArn":
 "arn:aws:ec2:region:123456789012:vpc/vpc-0b8276c84697e7339",
            "type": "AWS_IAM",
            "principal": "arn:aws:iam::123456789012:assumed-role/my-role/my-session",
            "principalOrgID": "o-50dc6c495c0c9188",
            "sessionName": "i-0c7de02a688bde9f7",
            "x509IssuerOu": "string",
            "x509SanDns": "string",
            "x509SanNameCn": "string",
            "x509SanUri": "string",
            "x509SubjectCn": "string"
        },
        "region": "region",
        "timeEpoch": "1690497599177430"
    }
}
```

body

请求的正文。仅在协议为 HTTP、HTTPS 或 gRPC 时出现。

headers

请求的 HTTP 标头。仅在协议为 HTTP、HTTPS 或 gRPC 时出现。

`identity`

身份信息。以下是可能的字段。

- `principal`：经过验证的主体。仅在 AWS 身份验证成功时出现。
- `principalOrgID`：经过验证主体的组织 ID。仅在 AWS 身份验证成功时出现。
- `sessionName`：经过验证会话的名称。仅在 AWS 身份验证成功时出现。
- `sourceVpcArn`：发出请求的 VPC 的 ARN。仅在可以识别源 VPC 时才会出现。
- `type`— `AWS_IAM` 如果使用了身份验证策略并且 AWS 身份验证成功，则该值为。

如果使用 Roles Anywhere 凭证且验证成功，则会显示以下字段。

- `x509IssuerOu`：发布者（OU）。
- `x509SanDns`：使用者备用名称（DNS）。
- `x509SanNameCn`：发布者备用名称（名称/CN）。
- `x509SanUri`：使用者备用名称（URI）。
- `x509SubjectCn`：使用者名称（CN）。

`isBase64Encoded`

表示正文是否经过 base64 编码。仅当协议为 HTTP、HTTPS 或 gRPC，且请求正文还不是字符串时才会出现。

`method`

请求中的 HTTP 方法。仅在协议为 HTTP、HTTPS 或 gRPC 时出现。

`path`

来自客户端的请求路径，其中包含查询字符串参数。仅在协议为 HTTP、HTTPS 或 gRPC 时出现。

`queryStringParameters`

HTTP 查询字符串参数。仅在协议为 HTTP、HTTPS 或 gRPC 时出现。

`serviceArn`

接收请求的服务的 ARN。

`serviceNetworkArn`

传送请求的服务网络的 ARN。

`targetGroupArn`

接收请求的目标组的 ARN。

`timeEpoch`

时间（微秒）。

Example示例事件：V1

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

响应 VPC Lattice 服务

来自 Lambda 函数的响应必须包含 Base64 编码状态、状态代码和标头。您可以省略正文。

要在响应的正文中包含二进制内容，您必须对内容进行 Base64 编码并将 `isBase64Encoded` 设置为 `true`。服务对内容进行解码以检索二进制内容，并将其发送到 HTTP 响应主体中的客户端。

VPC Lattice 服务不支持 hop-by-hop标头，例如`Connection`或`Transfer-Encoding`。您可以省略 `Content-Length` 标头，因为服务在向客户端发送响应之前会对其进行计算。

以下是来自 Lambda 函数的示例响应：

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

多值标头

VPC Lattice 支持来自客户端的请求或来自包含多个值的标头或多次包含相同标头的 Lambda 函数的响应。VPC Lattice 将所有值传递给目标。

在以下示例中，有两个标头header1以不同的值命名。

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

在 V2 事件结构中，VPC Lattice 以列表形式发送值。例如：

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

VPC Lattice 采用 V1 事件结构，将这些值组合成一个字符串。例如：

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

多值查询字符串参数

VPC Lattice 支持同一密钥具有多个值的查询参数。

在以下示例中，有两个QS1以不同值命名的参数。

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

在 V2 事件结构中，VPC Lattice 以列表形式发送值。例如：

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

在 V1 事件结构中，VPC Lattice 使用最后一次传递的值。例如：

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

注销 Lambda 函数

如果您不再需要向您的 Lambda 函数发送流量，则可以将其取消注册。在取消注册 Lambda 函数后，进行中的请求会失败，并显示 HTTP 5XX 错误。

要替换 Lambda 函数，建议您创建新的目标组，向新目标组注册新函数，并将侦听器规则更新为使用新目标组而不是现有目标组。

要使用新控制台取消注销 Lambda 函数

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 在 Targets (目标) 选项卡上，选择 Deregister (取消注册)。
5. 当系统提示确认时，输入 **`confirm`**，然后选择取消注册。

要取消注册 Lambda 函数，请使用 AWS CLI

使用 deregister-targets 命令。

VPC Lattice 中的应用程序负载均衡器作为目标

您可以创建一个 VPC Lattice 目标组，将单个内部应用程序负载均衡器注册为目标，并配置 VPC Lattice 服务以将流量转发到该目标组。在这种情况下，应用程序负载均衡器在流量到达后立即接管路由决策。此配置允许您将 Application Load Balancer 的基于请求的第 7 层路由功能与 VPC Lattice 支持的功能结合使用，例如 IAM 身份验证和授权以及跨 VPCs 账户的连接。

限制

- 您可以将单个内部应用程序负载均衡器注册为 VPC Lattice 目标组（类型为 ALB）中的目标。
- 您最多可以将应用程序负载均衡器注册为两个 VPC Lattice 目标组的目标，由两个不同的 VPC Lattice 服务使用。
- VPC Lattice 不会为 ALB 类型的目标组提供运行状况检查。但您可以在负载均衡器级别，为弹性负载均衡中的目标单独配置运行状况检查。有关更多信息，请参阅应用程序负载均衡器用户指南中的目标组运行状况检查

先决条件

创建应用程序负载均衡器，将其注册为 VPCLattice 目标组的目标。负载均衡器必须符合以下标准：

- 负载均衡器方案是内部方案。

- 应用程序负载均衡器必须与 VPC Lattice 目标组位于同一账户中，并且必须处于活动状态。
- 应用程序负载均衡器必须与 VPC Lattice 目标组位于同一 VPC 中。
- 您可以在 Application Load Balancer 上使用 HTTPS 侦听器来终止 TLS，但前提是 VPC Lattice 服务使用与负载均衡器相同的 SSL/TLS 证书。
- 要在 `X-Forwarded-For` 请求标头中保留 VPC Lattice 服务的客户端 IP，必须将应用程序负载均衡器 `routing.http.xff_header_processing.mode` 的属性设置为 `Preserve`。如果值为 `Preserve`，负载均衡器将保留 HTTP 请求中的 `X-Forwarded-For` 标头，并将其发送到目标而不做任何更改。

有关更多信息，请参阅《应用程序负载均衡器用户指南》中的创建应用程序负载均衡器。

步骤 1：创建 ALB 类型的目标组

使用以下过程创建目标组。请注意，VPC Lattice 不支持对ALB目标群体进行健康检查。但您可以为应用程序负载均衡器的目标组配置运行状况检查。有关更多信息，请参阅应用程序负载均衡器用户指南中的目标组运行状况检查。

要创建目标组

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择目标组。
3. 选择创建目标组。
4. 在指定目标组详细信息页面的基本配置下，选择应用程序负载均衡器作为目标类型。
5. 对于目标组名称，输入目标组的名称。
6. 对于 “协议”**HTTP**，选择**HTTPS**、或**TCP**。目标组协议必须与内部应用程序负载均衡器的侦听器协议匹配。
7. 对于端口，指定目标组的端口。此端口必须与内部应用程序负载均衡器的侦听器端口匹配。您也可以在内部应用程序负载均衡器上添加侦听器端口，以匹配您在此指定的目标组端口。
8. 对于 VPC，选择您在创建内部应用程序负载均衡器时选择的相同虚拟私有云（VPC）。此 VPC 应该是包含 VPC Lattice 资源的 VPC。
9. 对于协议版本，选择应用程序负载均衡器支持的协议版本。
10. （可选）添加任何所需的标签。
11. 选择下一步。

步骤 2：将应用程序负载均衡器注册为目标

您可以立即或稍后将负载均衡器注册为目标。

要将应用程序负载均衡器注册为目标

1. 选择立即注册。
2. 对于应用程序负载均衡器，请选择内部应用程序负载均衡器。
3. 对于端口，保留默认端口或根据需要指定不同的端口。此端口必须与内部应用程序负载均衡器上的现有侦听器端口匹配。如果在没有匹配端口的情况下继续，流量将无法到达应用程序负载均衡器。
4. 选择创建目标组。

协议版本

默认情况下，服务使用 HTTP/1.1 向目标发送请求。您可以通过协议版本使用 HTTP/2 或 gRPC 向目标发送请求。

下表汇总了请求协议和目标组协议版本组合的结果。

请求协议	协议版本	结果
HTTP/1.1	HTTP/1.1	成功
HTTP/2	HTTP/1.1	成功
gRPC	HTTP/1.1	错误
HTTP/1.1	HTTP/2	错误
HTTP/2	HTTP/2	成功
gRPC	HTTP/2	如果目标支持 grPC，则成功
HTTP/1.1	gRPC	错误
HTTP/2	gRPC	如果 POST 请求，则成功
gRPC	gRPC	成功

gRPC 协议版本的注意事项

- 唯一支持的侦听器协议是 HTTPS。
- 唯一支持的目标类型是 INSTANCE 和 IP。
- 服务会解析 gRPC 请求，并根据包、服务和方法将 gRPC 调用路由到相应的目标组。
- 不能将 Lambda 函数用作目标。

HTTP/2 协议版本的注意事项

- 唯一支持的侦听器协议是 HTTPS。您可以为目标组协议选择 HTTP 或 HTTPS。
- 唯一支持的侦听器规则是正向和固定响应。
- 唯一支持的目标类型是 INSTANCE 和 IP。
- 服务支持来自客户端的流。服务不支持发往目标的流。

VPC Lattice 目标组的标签

标签有助于按各种标准 (例如，用途、所有者或环境) 对目标组进行分类。

您可以为每个目标组添加多个标签。每个目标组的标签键必须是唯一的。如果您添加的标签中的键已经与目标组关联，它将更新该标签的值。

用完标签后可以将其删除。

限制

- 每个资源的标签数上限 – 50
- 最大密钥长度 - 127 个 Unicode 字符
- 最大值长度 - 255 个 Unicode 字符
- 标签键和值区分大小写。允许使用的字符包括可用 UTF-8 格式表示的字母、空格和数字，以及以下特殊字符： + - = 。_ : / @。请不要使用前导空格或尾随空格。
- 请勿在标签名称或值中使用aws:前缀，因为它已保留供 AWS 使用。您无法编辑或删除带此前缀的标签名称或值。具有此前缀的标签不计入每个资源的标签数限制。

使用控制台更新目标组的标签

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。

2. 在导航窗格中的 VPC Lattice 下，选择目标组。
3. 选择目标组的名称以打开其详细信息页面。
4. 选择标签选项卡。
5. 要添加标签，请选择添加标签，然后输入标签键和标签值。要添加其他标签，请选择添加新标签。添加完标签后，选择 Save changes（保存更改）。
6. 要删除标签，请选择标签的复选框，然后选择删除。提示进行确认时，输入 **`confirm`**，然后选择删除。

要更新目标群组的标签，请使用 AWS CLI

使用 tag-resource 和 untag-resource 命令。

删除 VPC 莱迪思目标组

如果目标组未由任何侦听器规则的转发操作引用，则可以删除该目标组。删除目标组不会影响已注册到目标组的目标。如果您不再需要已注册的 EC2 实例，则可以停止或终止该实例。

使用控制台删除目标组

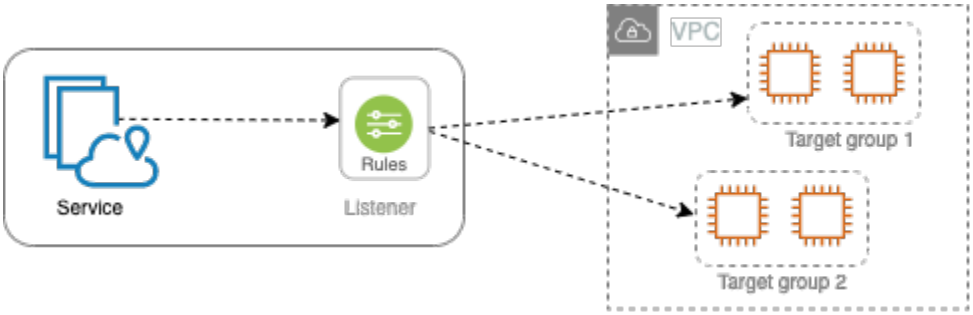
1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中，选择目标组。
3. 选中目标组的复选框，然后依次选择操作和删除。
4. 提示进行确认时，输入 **`confirm`**，然后选择删除。

要使用删除目标组 AWS CLI

使用 delete-target-group 命令。

VPC Lattice 服务的侦听器

在开始使用您的VPC Lattice服务之前，您必须添加一个监听器。侦听器是一个使用您配置的协议和端口检查连接请求的进程。为侦听器定义的规则决定了服务如何将请求路由到其注册的目标。



内容

侦听器配置

侦听器支持以下协议和端口：

- 协议：HTTP、HTTPS、TLS
- 端口：1-65535

如果侦听器协议是 HTTPS，VPC Lattice 将配置和管理与 VPC Lattice 生成的 FQDN 关联的 TLS 证书。VPC Lattice 支持 HTTP/1.1 和 HTTP/2 上的 TLS。当您使用 HTTPS 侦听器配置服务时，VPC Lattice 将使用应用程序层协议协商（ALPN）自动确定 HTTP 协议。如果没有 ALPN，VPC Lattice 默认使用 HTTP/1.1。有关更多信息，请参阅 HTTPS 侦听器。

VPC Lattice 可以侦听 HTTP、HTTPS、HTTP/1.1 和 HTTP/2，并使用这些协议和版本与目标通信。我们不要求侦听器和目标组协议匹配。VPC Lattice 管理协议和版本之间升级和降级的整个过程。有关更多信息，请参阅 协议版本。

您可以创建 TLS 侦听器来确保您的应用程序解密加密流量，而不是 VPC Lattice。有关更多信息，请参阅 TLS 侦听器。

VPC 莱迪思本身不支持。 WebSockets但是，您仍然可以使用 TLS 侦听器或通过 VPC Lattice 资源进行路由，连接到基于 WebSocket 的服务。

VPC Lattice 服务的 HTTP 侦听器

侦听器是用于检查连接请求的进程。您可以在创建 VPC Lattice 服务时定义侦听器。您可以随时向服务添加侦听器。

此页面上的信息可帮助您为服务创建 HTTP 侦听器。有关创建使用其他协议的监听器的信息，请参阅HTTPS 侦听器和。TLS 侦听器

先决条件

- 要将转发操作添加到默认侦听器规则，您必须指定可用的 VPC Lattice 目标组。有关更多信息，请参阅 创建 VPC Lattice 目标组。
- 您可以在多个侦听器中指定同一个目标组，但这些侦听器必须属于同一服务。要在 VPC Lattice 服务中使用目标组，必须验证目标组未被任何其他 VPC Lattice 服务的侦听器使用。

添加 HTTP 侦听器

您可以随时向服务添加侦听器和规则。为侦听器配置一个协议和端口，用于从客户端连接到服务，并为默认侦听器规则配置一个 VPC Lattice 目标组。有关更多信息，请参阅 侦听器配置。

使用控制台添加 HTTP 侦听器

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择服务名称以打开其详细信息页面。
4. 在路由选项卡上，选择添加侦听器。
5. 对于侦听器名称，您可以提供自定义侦听器名称，也可以使用侦听器的协议和端口作为侦听器名称。您指定的自定义名称最多可包含 63 个字符，且对账户中的每项服务必须是唯一的。有效字符：a-z、0-9 和连字符（-）。不能将连字符用作第一个或最后一个字符，也不能紧跟在另一个连字符之后。创建后无法更改名称。
6. 对于协议：端口，选择 HTTP，然后输入端口号。

7. 对于默认操作，选择要接收流量的 VPC Lattice 目标组，然后选择要分配给该目标组的权重。为目标组分配的权重可用于设定目标组接收流量的优先级。例如，如果两个目标组具有相同的权重，则每个目标组将接收一半的流量。如果只指定了一个目标组，则 100% 的流量将发送到一个目标组。

   您可以选择为默认操作添加另一个目标组。选择添加操作，然后选择一个目标组并指定其权重。
8. （可选）要添加其他规则，请选择添加规则，然后输入规则的名称、优先级、条件和操作。

   您可以为每条规则指定一个 1 到 100 之间的优先级编号。侦听器不能具有优先级相同的多个规则。规则是按优先级顺序 (从最低值到最高值) 计算的。最后评估默认规则。有关更多信息，请参阅 侦听器规则。
9. （可选）要添加标签，请展开侦听器标签，选择添加新标签，然后输入标签键和标签值。
10. 检查您的配置，然后选择添加。

要添加 HTTP 侦听器，请使用 AWS CLI

使用 create-listener 命令创建具有默认规则的侦听器，并使用 create-rule 命令创建其他侦听器规则。

VPC Lattice 服务的 HTTPS 侦听器

侦听器是用于检查连接请求的进程。您可以在创建服务时定义侦听器。您可以随时向 VPC Lattice 中的服务添加侦听器。

您可以创建 HTTPS 侦听器，该侦听器使用 TLS 版本 1.2 或 TLS 1.3 直接终止与 VPC Lattice 的 HTTPS 连接。VPC Lattice 将预置和管理与 VPC Lattice 生成的完全限定域名（FQDN）关联的 TLS 证书。VPC Lattice 支持 HTTP/1.1 和 HTTP/2 上的 TLS。当您使用 HTTPS 侦听器配置服务时，VPC Lattice 将通过应用程序层协议协商（ALPN）自动确定 HTTP 协议。如果没有 ALPN，VPC Lattice 默认使用 HTTP/1.1。

VPC Lattice 使用多租户架构，这意味着可以在同一端点上托管多个服务。VPC Lattice 对每个客户端请求使用带有服务器名称指示（SNI）的 TLS。不支持加密的客户端问候 (ECH) 和加密服务器名称指示 (ESNI)。

VPC Lattice 可以侦听 HTTP、HTTPS、HTTP/1.1 和 HTTP/2，并使用这些协议和版本与目标通信。这些侦听器和目标组配置不需要匹配。VPC Lattice 管理协议和版本之间升级和降级的整个过程。有关更多信息，请参阅 协议版本。

为确保您的应用程序解密流量，请改为创建 TLS 侦听器。通过 TLS 直通，VPC 莱迪思不会终止 TLS。有关更多信息，请参阅 TLS 侦听器。

目录

安全策略

VPC Lattice 使用的安全策略是 TLSv1 .2 协议和 SSL/TLS 密码列表的组合。该协议在客户端和服务器之间建立安全连接，有助于确保客户端和 VPC Lattice 中的服务之间传递的所有数据均为私有。密码是使用加密密钥创建编码消息的加密算法。协议使用多种密码对数据进行加密。在连接协商过程中，客户端和 VPC Lattice 按优先顺序提供各自支持的密码和协议列表。默认情况下，会为安全连接选择服务器列表中与任何一个客户端的密码匹配的第一个密码。

VPC Lattice 按以下优先顺序使用以下 TLS 1.2 SSL/TLS 密码：

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

VPC Lattice 还按以下优先顺序使用以下 TLS 1.3 SSL/TLS 密码：

- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256

ALPN 策略

应用程序层协议协商（ALPN）是一种 TLS 扩展，在初始 TLS 握手 hello 消息中发送。通过 ALPN，应用层能够协商应在安全连接（例如 HTTP/1 和 HTTP/2）上使用什么协议。

当客户端发起 ALPN 连接时，VPC Lattice 服务会将客户端 ALPN 首选项列表与其 ALPN 策略进行比较。如果客户端支持来自 ALPN 策略的协议，VPC Lattice 服务会根据 ALPN 策略的首选项列表建立连接。否则，服务不使用 ALPN。

VPC Lattice 支持以下 ALPN 策略：

`HTTP2Preferred`

首选 HTTP/2 而不是 HTTP/1.1。ALPN 首选项列表是 h2、http/1.1。

添加 HTTPS 侦听器

为侦听器配置一个协议和端口，用于从客户端连接到服务，并为默认侦听器规则配置一个目标组。有关更多信息，请参阅 侦听器配置。

先决条件

- 要将转发操作添加到默认侦听器规则，您必须指定可用的 VPC Lattice 目标组。有关更多信息，请参阅 创建 VPC Lattice 目标组。
- 您可以在多个侦听器中指定同一个目标组，但这些侦听器必须属于同一 VPC Lattice 服务。要在 VPC Lattice 服务中使用目标组，必须验证目标组未被任何其他 VPC Lattice 服务的侦听器使用。
- 您可以使用 VPC Lattice 提供的证书，也可以将自己的证书导入到。 AWS Certificate Manager有关更多信息，请参阅 the section called “BYOC”。

使用控制台添加 HTTPS 侦听器

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择服务名称以打开其详细信息页面。
4. 在路由选项卡上，选择添加侦听器。
5. 对于侦听器名称，您可以提供自定义侦听器名称，也可以使用侦听器的协议和端口作为侦听器名称。您指定的自定义名称最多可包含 63 个字符，且对账户中的每项服务必须是唯一的。有效字符：a-z、0-9 和连字符（-）。不能将连字符用作第一个或最后一个字符，也不能紧跟在另一个连字符之后。创建侦听器后，不能更改其名称。
6. 对于协议：端口，选择 HTTPS，然后输入端口号。
7. 对于默认操作，选择要接收流量的 VPC Lattice 目标组，然后选择要分配给该目标组的权重。为目标组分配的权重可用于设定目标组接收流量的优先级。例如，如果两个目标组具有相同的权重，

则每个目标组将接收一半的流量。如果只指定了一个目标组，则 100% 的流量将发送到一个目标组。

您可以选择为默认操作添加另一个目标组。选择添加操作，然后选择一个目标组并指定其权重。

8. （可选）要添加其他规则，请选择添加规则，然后输入规则的名称、优先级、条件和操作。

   您可以为每条规则指定一个 1 到 100 之间的优先级编号。侦听器不能具有优先级相同的多个规则。规则是按优先级顺序 (从最低值到最高值) 计算的。最后评估默认规则。有关更多信息，请参阅 侦听器规则。

9. （可选）要添加标签，请展开侦听器标签，选择“添加新标签”，然后输入标签键和标签值。
10. 对于 HTTPS 侦听器证书设置，如果您在创建服务时未指定自定义域名，VPC Lattice 会自动生成 TLS 证书，以保护流经侦听器的流量。

    如果您使用自定义域名创建了服务，但没有指定匹配的证书，则现在可以通过从自定义 SSL/TLS 证书中选择证书来完成此操作。否则，选择您在创建服务时指定的证书。

11. 检查您的配置，然后选择添加。

要添加 HTTPS 侦听器，请使用 AWS CLI

使用 create-listener 命令创建具有默认规则的侦听器，并使用 create-rule 命令创建其他侦听器规则。

VPC 莱迪思服务的 TLS 监听器

侦听器是用于检查连接请求的进程。您可以在创建 VPC Lattice 服务时定义侦听器。您可以随时向服务添加侦听器。

您可以创建一个 TLS 侦听器，这样 VPC Lattice 就可以将加密流量传递到您的应用程序，而无需对其进行解密。

如果您更喜欢 VPC Lattice 解密加密流量并将未加密的流量发送到您的应用程序，请改为创建 HTTPS 侦听器。有关更多信息，请参阅 HTTPS 侦听器。

注意事项

以下注意事项适用于 TLS 侦听器：

- VPC 莱迪思服务必须具有自定义域名。服务自定义域名用作服务名称指示 (SNI) 匹配。如果您在创建服务时指定了证书，则不会使用该证书。
- TLS 侦听器唯一允许使用的规则是默认规则。

- TLS 侦听器的默认操作必须是向 TCP 目标组执行的转发操作。
- 默认情况下，TCP 目标组的运行状况检查处于禁用状态。如果您为 TCP 目标组启用运行状况检查，则必须指定协议和协议版本。
- TLS 侦听器使用客户端 hello 消息的 SNI 字段路由请求。如果匹配条件与 client-hello 完全匹配，则可以在目标上使用通配符和 SAN 证书。
- 由于从客户端到目标的所有流量都保持加密状态，因此 VPC Lattice 无法读取 HTTP 标头，也无法插入或删除 HTTP 标头。因此，使用 TLS 侦听器时，存在以下限制：
  - 连接时长限制为 10 分钟
  - 身份验证政策仅限于匿名委托人
  - 不支持 Lambda 目标
- Websocket 连接可以使用 TLS 侦听器连接到 VPC 莱迪思服务。存在以下限制：
  - 连接时长限制为 10 分钟
  - 身份验证政策仅限于匿名委托人
  - 不支持 Lambda 目标
- 不支持加密的客户端问候 (ECH)。
- 不支持加密服务器名称指示 (ESNI)。

添加 TLS 监听器

为侦听器配置一个协议和端口，用于从客户端连接到服务，并为默认侦听器规则配置一个目标组。有关更多信息，请参阅 侦听器配置。

使用控制台添加 TLS 侦听器

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择服务名称以打开其详细信息页面。
4. 在路由选项卡上，选择添加侦听器。
5. 对于侦听器名称，您可以提供自定义侦听器名称，也可以使用侦听器的协议和端口作为侦听器名称。您指定的自定义名称最多可包含 63 个字符，且对账户中的每项服务必须是唯一的。有效字符：a-z、0-9 和连字符 (-)。不能将连字符用作第一个或最后一个字符，也不能紧跟在另一个连字符之后。创建侦听器后，不能更改其名称。
6. 对于协议，选择 TLS。对于端口，输入端口号。

7. 对于转发到目标组，请选择使用TCP协议接收流量的VPC Lattice目标组，然后选择要分配给该目标组的权重。您可以选择添加其他目标组。选择 “添加目标组”，然后选择目标组并输入其权重。
8. （可选）要添加标签，请展开侦听器标签，选择“添加新标签”，然后输入标签键和标签值。
9. 检查您的配置，然后选择添加。

要添加 TLS 侦听器，请使用 AWS CLI

使用 create-listener 命令使用默认规则创建监听器。指定 TLS_PASSTHROUGH 协议。

VPC Lattice 服务的侦听器规则

每个侦听器都具有默认规则，您也可以定义其他规则。每条规则由优先级、一个或多个操作以及一个或多个条件组成。您可以随时添加或编辑规则。

内容

默认规则

创建侦听器时，请为默认规则定义操作。默认规则不能有条件。如果未满足侦听器的任一规则条件，则将执行默认规则的操作。

规则优先级

每个规则都有一个优先级。规则是按优先级顺序 (从最低值到最高值) 计算的。最后评估默认规则。您可以随时更改非默认规则的优先级。您不能更改默认规则的优先级。

规则操作

VPC Lattice 服务的侦听器支持转发操作和固定响应操作。

转发操作

您可以使用 forward 操作，将请求路由到一个或多个 VPC Lattice 目标组。如果为某个 forward 操作指定多个目标组，您必须为每个目标组指定权重。每个目标组权重都是一个介于 0 到 999 之间的值。对于将侦听器规则与加权目标组匹配的请求，会根据这些目标组的权重分配给这些目标组。例如，如果指定两个目标组，每个目标组的权重为 10，则每个目标组将接收一半的请求。如果指定两个目标组，一个权重为 10，另一个权重为 20，则权重为 20 的目标组接收的请求将是另一个目标组的两倍。

固定响应操作

您可以使用 fixed-response 操作删除客户端请求并返回自定义 HTTP 响应。您可以使用此操作返回 404 或 500 响应码。

Example的固定响应操作示例 AWS CLI

您可以在创建或更新规则时指定操作。以下操作发送具有指定状态代码的固定响应。

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

规则条件

每个规则条件都有类型和配置信息。当规则的条件满足时，将执行其操作。

以下是规则支持的匹配条件：

标头匹配

路由基于每个请求的 HTTP 标头。您可以使用 HTTP 标头条件来配置基于请求的 HTTP 标头路由请求的规则。您可以指定标准或自定义 HTTP 标头字段的名称。标头名称和匹配评估不区分大小写。您可以通过开启区分大小写来更改此设置。标头名称不支持通配符。标头匹配支持前缀匹配、精确匹配和包含匹配。

方法匹配

路由基于每个请求的 HTTP 请求方法。

您可以使用 HTTP 请求方法条件来配置基于请求的 HTTP 请求方法路由请求的规则。您可以指定标准或自定义 HTTP 方法。方法匹配区分大小写。方法名称必须完全匹配。不支持通配符。

路径匹配

路由基于匹配请求中的路径模式 URLs。

您可以使用路径条件定义规则，根据请求中的 URL 路由请求。不支持通配符。支持路径前缀和精确匹配。

添加规则

您可以随时添加侦听器规则。

要使用控制台添加侦听器

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择服务名称以打开其详细信息页面。
4. 在路由选项卡上，选择编辑侦听器。
5. 展开侦听器规则并选择添加规则。
6. 对于规则名称，输入规则的名称。
7. 对于优先级，输入 1 到 100 之间的优先级。规则是按优先级顺序 (从最低值到最高值) 计算的。最后评估默认规则。
8. 对于条件，输入路径匹配条件的路径模式。每个字符串的最大长度为 200 个字符。比较不区分大小写。不支持通配符。

   要添加标头匹配或方法匹配规则条件，请使用 AWS CLI 或 S AWS DK。
9. 对于操作，选择 VPC Lattice 目标组。
10. 选择保存更改。

要添加规则，请使用 AWS CLI

使用 create-rule 命令。

更新规则

您可以随时更新侦听器规则。您可以修改其优先级、条件、目标组以及每个目标组的权重。无法修改规则的名称。

要使用控制台更新侦听器规则

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择服务名称以打开其详细信息页面。
4. 在路由选项卡上，选择编辑侦听器。
5. 根据需要修改规则优先级、条件和操作。
6. 查看您的更新并选择保存更改。

要更新规则，请使用 AWS CLI

使用 update-rule 命令。

删除规则

您可以随时删除侦听器的非默认规则。您不能删除侦听器的默认规则。删除侦听器时，会删除所有规则。

要使用控制台删除侦听器规则

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择服务名称以打开其详细信息页面。
4. 在路由选项卡上，选择编辑侦听器。
5. 找到规则并选择删除。
6. 选择保存更改。

要删除规则，请使用 AWS CLI

使用 delete-rule 命令。

删除您的 VPC 莱迪思服务的监听器

可以随时删除侦听器。删除侦听器后，其所有规则都会自动删除。

使用控制台删除侦听器

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务。
3. 选择服务名称以打开其详细信息页面。
4. 在路由选项卡上，选择删除侦听器。
5. 提示进行确认时，输入 **`confirm`**，然后选择删除。

要删除监听器，请使用 AWS CLI

使用 delete-listener 命令。

亚马逊 VPC 莱迪思中的 VPC 资源

您可以与组织中的其他团队或外部独立软件供应商 (ISV) 合作伙伴共享 VPC 资源。VPC 资源可以是AWS原生资源，例如 Amazon RDS 数据库、域名或 IP 地址。资源可以位于您的 VPC 或本地网络中，无需进行负载平衡。您可以使用 AWS RAM 指定可以访问资源的委托人。您可以创建一个资源网关，通过该网关可以访问您的资源。您还可以创建代表要共享的资源或一组资源的资源配置。

与您共享资源的委托人可以使用 VPC 终端节点私下访问这些资源。他们可以使用资源VPC终端节点访问一个资源或在VPC Lattice服务网络中汇集多个资源，并使用服务网络VPC终端节点访问服务网络。

以下各节介绍如何在 VPC Lattice 中创建和管理 VPC 资源：

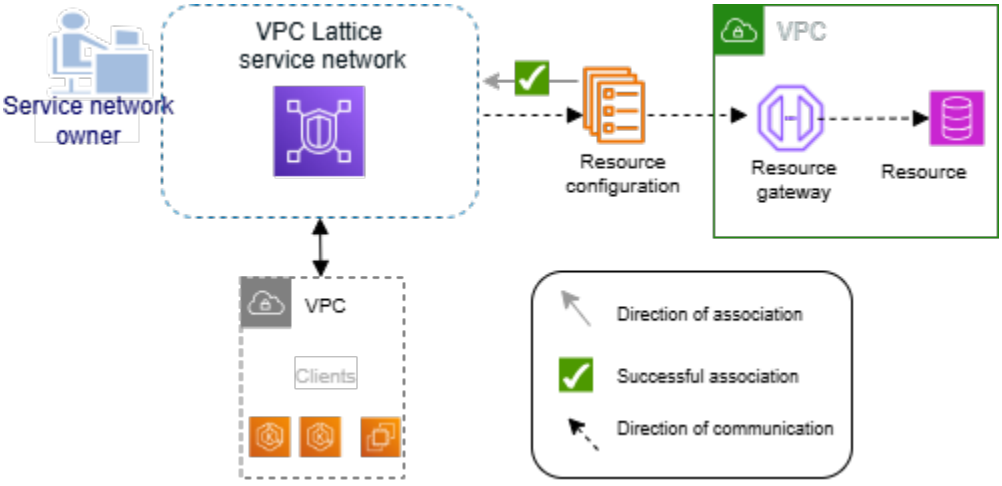
主题

VPC Lattice 中的资源网关

资源网关是接收进入资源所在的 VPC 的流量的点。资源网关跨越多个可用区。

如果您计划让其他 VPCs 或账户访问 VPC 内的资源，VPC 必须有资源网关。您共享的每个资源都与资源网关关联。当其他 VPCs 或账户中的客户访问您的 VPC 中的资源时，该资源会看到来自该 VPC 中资源网关的本地流量。流量的源 IP 地址是可用区中资源网关的 IP 地址。可以将多个资源配置（每个配置都有多个资源）附加到资源网关。

下图显示了客户端如何通过资源网关访问资源：



内容

注意事项

需注意以下资源网关相关事项：

- 为了使您的资源可以从所有可用区访问，您应该创建资源网关，使其跨越尽可能多的可用区。
- VPC 端点和资源网关必须至少有一个可用区重叠。
- 一个 VPC 最多可包含 100 个资源网关。有关更多信息，请参阅 VPC Lattice 的配额。
- VPC Lattice 可能会为您的资源网关 ENIs 添加新内容。
- 具有共享 VPC 子网的资源网关：
  - 资源网关只能由拥有 VPC 的账户部署到共享 VPC 子网中。
  - 资源网关的资源配置只能由拥有该资源网关的账户创建。

安全组

您可以将安全组附加到资源网关。资源网关的安全组规则控制从资源网关到资源的出站流量。

针对从资源网关流向数据库资源的流量建议的出站规则

要使流量从资源网关流向资源，必须为资源接受的侦听器协议和端口范围创建出站规则。

目标位置	协议	端口范围	Comment
CIDR range for resource	TCP	3306	允许从资源网关流向数据库的流量。

IP 地址类型

资源网关可以有 IPv4 IPv6 或双栈地址。资源网关的 IP 地址类型必须与资源网关的子网兼容，资源的 IP 地址类型如下所述：

- IPv4— 为您的资源网关网络接口分配 IPv4 地址。只有当所有选定的子网都有 IPv4 地址范围并且资源也有地址时，才支持此选项。 IPv4 使用此选项时，您可以配置每个资源网关 ENI IPv4 的地址数。
- IPv6— 为您的资源网关网络接口分配 IPv6 地址。仅当所有选定的子网仅为子网并且资源还有地址时， IPv6 才支持此选项。 IPv6 使用此选项时， IPv6 地址会自动分配，无需管理。
- Dualstack — 将 IPv4 和 IPv6 地址分配给您的资源网关网络接口。仅当所有选定的子网同时具有 IPv4 和 IPv6 地址范围，并且资源具有 IPv4 或 IPv6 地址时，才支持此选项。使用此选项时，您可以配置每个资源网关 ENI IPv4 的地址数。

资源网关的 IP 地址类型独立于客户端的 IP 地址类型或通过其访问资源的 VPC 端点。

IPv4 每个 ENI 的地址数

如果您的资源网关具有 IPv4 或双堆栈 IP 地址类型，则可以配置分配给资源网关每个 ENI IPv4 的地址数量。创建资源网关时，可以从 1 到 62 个 IPv4 地址中进行选择。一旦设置了 IPv4地址数量，就无法更改该值。

这些 IPv4 地址用于网络地址转换，并确定资源的最大并发 IPv4 连接数。每个 IPv4 地址最多可以支持每个目标 IP 的 55,000 个并发连接。默认情况下，所有资源网关为每 IPv4 个 ENI 分配 16 个地址。

如果您的资源网关使用 IPv6 地址类型，则资源网关会自动为每个 ENI 接收 /80 CIDR。此值不能更改。每个连接的最大传输单位 (MTU) 为 8500 字节。

资源 Config DNS 解析

您可以指定资源网关如何对作为域名目标的资源配置进行 DNS 解析。此属性是不可变的。您可以选择：

- PUBLI@@ C（默认）-使用公共 DNS 解析器解析域名。
- IN_VPC-使用资源网关所在的 VPC 的 DHCP 选项集中配置的 DNS 服务器来解析域名。如果您使用的是私有 DNS 服务器或者您的域名目标位于 Route53 私有托管区域中，则应使用此选项。

如果 DNS 解析为 IN_VPC，则无法将 ARN 定义的资源配置附加到资源网关。如果资源网关使用 IPv6 仅限子网，则无法将 DNS 解析设置为 IN_VPC。

在 VPC Lattice 中创建资源网关

使用控制台创建资源网关。

使用控制台创建资源网关

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中，在 Lattice PrivateLink 和 Lattice 下，选择资源网关。
3. 选择创建资源网关。
4. 在资源网关名称中，输入一个在您的 AWS 账户中唯一的名称。
5. 对于 IP 地址类型，选择资源网关的 IP 地址类型。
   - 如果您选择了 IPv4Dualstack 作为 IP 地址类型，则可以为资源网关输入每 IPv4 个 ENI 的地址数。

     默认为每 IPv4 个 ENI 有 16 个地址。这是与您的后端资源 IPs 建立连接的合适数量。
6. 对于 VPC，请选择要在其中创建资源网关的 VPC 和子网。
7. 对于安全组，最多可选择五个安全组来控制从 VPC 到服务网络的入站流量。
8. 对于 R esource Config DNS 解析，请选择您希望如何为域名目标解析 DNS。
   - 如果您使用的是私有 DNS 服务器或者您的域名目标位于 Route53 私有托管区域中，请设置为 IN_VPC
9. （可选）若要添加标签，请选择添加新标签，然后输入该标签的键和值。
10. 选择创建资源网关。

要使用创建资源网关 AWS CLI

使用 create-resource-gateway 命令。

在 VPC Lattice 中删除资源网关

使用控制台删除资源网关。

使用控制台删除资源网关

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中，在 Lattice PrivateLink 和 Lattice 下，选择资源网关。
3. 选中要删除的资源网关的复选框，然后依次选择操作、删除。提示进行确认时，输入 **`confirm`**，然后选择删除。

要删除资源网关，请使用 AWS CLI

使用 delete-resource-gateway 命令。

VPC 资源的资源配置

资源配置表示您希望允许其他 VPCs 和账户中的客户访问的资源或一组资源。通过定义资源配置，您可以允许从其他 VPCs 和账户中的客户端到您的 VPC 中的资源的私有、安全、单向的网络连接。资源配置与资源网关关联，通过该资源网关接收流量。要从其他 VPC 访问资源，需要对其进行资源配置。

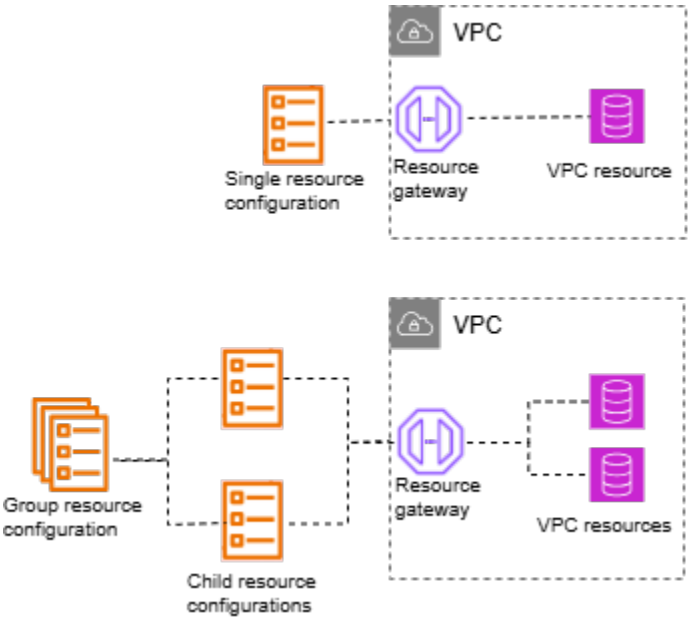
内容

资源配置的类型

资源配置可以有几种类型。不同的类型有助于代表不同种类的资源。这些类型是：

- 单一资源配置：表示 IP 地址或域名。它可以独立共享。
- 组资源配置：它是子资源配置的集合。它可以用来表示一组 DNS 和 IP 地址端点。
- 子资源配置：它是组资源配置的成员。它代表 IP 地址或域名。它不能单独共享；只能作为群组的一部分共享。可以将其添加到群组中或从群组中删除。添加后，有权访问这些配置的用户可以自动访问该组。
- ARN 资源配置：表示由服务配置的支持的资源类型。 AWS 任何团体与子女的关系都会自动得到处理。

下图显示了单个、子资源和组资源配置：



协议

创建资源配置时，可以定义该资源将支持的协议。目前仅支持 TCP 协议。

资源网关

资源配置与资源网关关联。资源网关是一组资源网关 ENIs ，用作资源所在的 VPC 的入口点。多个资源配置可以与同一个资源网关关联。当其他 VPCs 或账户中的客户访问您的 VPC 中的资源时，该资源会看到来自该 VPC 中资源网关 IP 地址的本地流量。

资源提供商的自定义域名

资源提供者可以将自定义域名附加到资源配置，例如`example.com`，资源使用者可以使用该域名来访问资源配置。自定义域名可以由资源提供者拥有和验证，也可以是第三方或 AWS 域名。资源提供者可以使用资源配置来共享缓存集群和 Kafka 集群、基于 TLS 的应用程序或其他 AWS 资源。

以下注意事项适用于资源配置提供者：

- 一个资源配置只能有一个自定义域。
- 资源配置的自定义域名无法更改。
- 所有资源配置使用者都可以看到自定义域名。
- 您可以使用 VPC Lattice 中的域名验证流程来验证您的自定义域名。有关更多信息，请参阅the section called “创建并验证域名”。
- 对于 group 和 child 类型的资源配置，必须先在组资源配置中指定组域。之后，子资源配置可以有自定义域，这些域名是组域的子域。如果群组没有群组域，则可以为子组使用任何自定义域名，但是 VPC Lattice 不会为资源使用者的 VPC 中的子域名配置任何托管区域。

资源使用者的自定义域名

当资源使用者启用与具有自定义域名的资源配置的连接时，他们可以允许 VPC Lattice 管理其 VPC 中的 Route 53 私有托管区域。资源使用者可以精细地选择他们希望允许VPC Lattice为哪些域管理私有托管区域。

当通过资源终端节点、服务网络终端节点或服务网络 VPC 关联启用与资源配置的连接时，资源使用者可以设置`private-dns-enabled`参数。除了`private-dns-enabled`参数外，消费者还可以使用 DNS 选项来指定他们希望 VPC Lattice 为哪些域名管理私有托管区域。消费者可以在以下私有 DNS 首选项之间进行选择：

`ALL_DOMAINS`

VPC Lattice 为所有自定义域名提供私有托管区域。

`VERIFIED_DOMAINS_ONLY`

VPC Lattice 只有在提供商验证了自定义域名后才会配置私有托管区域。

`VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS`

VPC Lattice 为所有经过验证的自定义域名和资源使用者指定的其他域名提供私有托管区域。资源使用者在`private DNS specified domains`参数中指定域名。

SPECIFIED_DOMAINS_ONLY

VPC Lattice 为资源使用者指定的域名提供私有托管区域。资源使用者在`private DNS specified domains` 参数中指定域名。

启用私有 DNS 后，VPC Lattice 会在您的 VPC 中为与资源配置关联的自定义域名创建一个私有托管区域。默认情况下，私有 DNS 首选项设置为`VERIFIED_DOMAINS_ONLY`。这意味着，只有在资源提供商验证了自定义域名后，才会创建私有托管区域。如果您将私有 DNS 首选项设置为`ALL_DOMAINS`或，`SPECIFIED_DOMAINS_ONLY`则无论自定义域名的验证状态如何，VPC Lattice 都会创建私有托管区域。为给定域创建私有托管区域时，从您的 VPC 到该域的所有流量都将通过 VPC Lattice 路由。我们建议您仅在`ALL_DOMAINS`希望这些自定义域名的流量通过 VPC Lattice 时使用`VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS`、或`SPECIFIED_DOMAINS_ONLY`首选项。

我们建议资源使用者将其私有 DNS 首选项设置为`VERIFIED_DOMAINS_ONLY`。这允许用户通过仅允许VPC Lattice为资源使用者账户中经过验证的域名配置私有托管区域来加强其安全边界。

要选择私有 DNS 指定域中的域，资源使用者可以输入完全限定的域名（例如）`my.example.com`或使用通配符（例如`*.example.com`）。

以下注意事项适用于资源配置的使用者：

- 私有 DNS 启用参数无法更改。
- 应在服务网络资源关联上启用私有 DNS，以便在 VPC 中创建私有托管。对于资源配置，服务网络资源关联的私有 DNS 启用状态会覆盖服务网络终端节点或服务网络 VPC 关联的私有 DNS 启用状态。

对于以域名为目标的资源配置，如果满足以下条件，则不会创建私有托管区域条目：

- 资源网关与服务网络 VPC 网络 VPC 关联位于同一 endpoint/service VPC 中。
- 资源网关上的 DNS 解析设置为 IN_VPC。
- 自定义域名或组域名与域名目标相同或更高级别的域名。

服务网络所有者的自定义域名

服务网络资源关联的私有 DNS 启用属性会覆盖服务网络终端节点和服务网络 VPC 关联的启用私有 DNS 的属性。

如果服务网络所有者创建了服务网络资源关联但未启用私有 DNS，则即使在服务网络终端节点或服务网络 VPC 关联上启用了私有 DNS VPCs ，VPC Lattice 也不会在服务网络所连接的任何资源配置中为该资源配置预置私有托管区域。

对于 ARN 类型的资源配置，私有 DNS 标志为真且不可变。

资源定义

在资源配置中，通过以下方式之一标识资源：

- 按亚马逊资源名称 (ARN)：由 AWS 服务配置的支持的资源类型可以通过其 ARN 来识别。仅支持 Amazon RDS 数据库。您无法为可公开访问的集群创建资源配置。
- 按域名目标：您可以使用任何域名。如果您使用私有 DNS 服务器或者您的域位于 Route53 私有托管区域中，则资源网关必须将 DNS 解析设置为 IN_VPC。如果您的域名指向位于您 VPC 之外的 IP，则您的 VPC 中必须有 NAT 网关。
- 按 IP 地址：对于 IPv4，请指定以下范围内的私有 IP：10.0.0.0/8、100.64.0.0/10、172.16.0.0/12、192.16.0.0/12、192.168.0.0/16。对于 IPv6，请指定来自 VPC 的 IP。 IPs 不支持公开。

端口范围

创建资源配置时，您可以定义它将在其上接受请求的端口。不允许客户端通过其他端口进行访问。

访问 资源

使用者可以通过 VPC 端点或服务网络，直接从其 VPC 访问资源配置。作为使用者，您可以启用从自己的 VPC 访问位于您账户中或通过 AWS RAM从其他账户与您共享的资源配置。

- 直接访问资源配置

  您可以在 AWS PrivateLink VPC 中创建资源类型（资源终端节点）的 VPC 终端节点，以便从您的 VPC 私下访问资源配置。有关如何创建资源端点的更多信息，请参阅 AWS PrivateLink用户指南中的访问 VPC 资源。
- 通过服务网络访问资源配置

  您可以将资源配置关联到服务网络，并将您的 VPC 连接到该服务网络。您可以通过关联或使用服务网络 VPC 终端节点将您的 VPC 连接到 AWS PrivateLink 服务网络。

  有关服务网络关联的更多信息，请参阅管理 VPC Lattice 服务网络的关联。

有关服务网络 VPC 端点的更多信息，请参阅 AWS PrivateLink 用户指南中的访问服务网络。

当您的 VPC 启用私有 DNS 后，您无法为相同的资源配置创建资源端点和服务网络端点。

与服务网络类型关联

当您与使用者账户（例如 Account-B）共享资源配置时，Account-B 可以通过资源 VPC 终端节点直接访问资源配置，也可以通过服务网络访问资源配置。 AWS RAM

要通过服务网络访问资源配置，Account-B 必须将资源配置与服务网络相关联。服务网络可在账户之间共享。因此，Account-B 可以与 Account-C 共享其服务网络（资源配置与之关联），从而使您的资源可以从 Account-C 访问。

为了防止此类传递共享，您可以指定不能将自己的资源配置添加到可在账户之间共享的服务网络中。如果您指定此项，则 Account-B 无法将您的资源配置添加到已共享或将来可以与其他账户共享的服务网络中。

服务网络的类型

当您通过 AWS RAM与其他账户（例如 Account-B）共享资源配置时，Account-B 可以通过以下三种方式之一访问资源配置中指定的资源：

- 使用资源类型的 VPC 端点（资源 VPC 端点）。
- 使用服务网络类型的 VPC 端点（服务网络 VPC 端点）。
- 使用服务网络 VPC 关联。

  使用服务网络关联时，每个资源都将从 129.224.0.0/17 区块中为每个子网分配一个 IP，该区块归所有且不可路由。 AWS 这是对托管前缀列表的补充，VPC Lattice 使用该列表，通过 VPC Lattice 网络将流量路由到服务。这两项 IPs 都已更新到您的 VPC 路由表中。

对于服务网络 VPC 端点和服务网络 VPC 关联，资源配置必须与账户 B 中的服务网络相关联。 服务网络可在账户之间共享。因此，Account-B 可以与 Account-C 共享其服务网络（包含资源配置），从而使您的资源可以从 Account-C 访问。 为了防止此类传递共享，您可以禁止将自己的资源配置添加到可在账户之间共享的服务网络中。如果您禁止此操作，那么 Account-B 就无法将您的资源配置添加到已共享或可以与其他账户共享的服务网络中。

通过共享资源配置 AWS RAM

资源配置与集成 AWS Resource Access Manager。也可以通过 AWS RAM与其他账户共享资源配置。当您与账户共享资源配置时，该 AWS 账户中的客户可以私下访问该资源。您可在 AWS RAM中使用资源共享来共享资源配置。

使用 AWS RAM 控制台查看您已添加到的资源共享、您可以访问的共享资源以及与您共享资源的 AWS 账户。有关更多信息，请参阅 AWS RAM 用户指南中的与您共享的资源。

要从与资源配置相同的账户中的其他 VPC 访问资源，您无需通过共享资源配置 AWS RAM。

监控

您可以对资源配置启用监控日志。您可以选择要将日志发送到其中的目的地。

创建并验证域名

域名验证是一个允许您证明自己对给定域名的所有权的实体。作为资源提供商，您可以使用域及其子域名作为资源配置的自定义域名。资源使用者在描述资源配置时可以看到您的自定义域名的验证状态。

开始域名验证

您使用 VPC Lattice 开始域名验证，然后使用您的 DNS 区域来完成该过程。

AWS 管理控制台

开始域名验证

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格的PrivateLink 和莱迪思下，选择域名验证
3. 选择 “开始域名验证”。
4. 在 “域名” 中，输入您拥有的域名。
5. （可选）若要添加标签，请选择添加新标签，然后输入该标签的键和值。
6. 选择 “开始域名验证”。

成功开始域名验证后，VPC Lattice 会返回`Id`和。`txtMethodConfig`您可以使用`txtMethodConfig`来完成对您的域名的验证。

AWS CLI

以下start-domain-verification命令启动域名验证：

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

输出内容如下所示：

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

VPC Lattice 返回Id和。txtMethodConfig您可以使用txtMethodConfig来完成对您的域名的验证。在此示例中，txtMethodConfig如下所示：

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

完成域名验证

要完成域名验证，您需要在 DNS 区域中添加 TXT 记录。如果您使用 Route 53，请使用您的域名的托管区域。当您验证域名时，所有子域名也会被验证。例如，如果您进行验证example.com，则可以将资源配置与任何其他验证关联起来alpha.example.com，beta.example.com而不必执行任何其他验证。

要使用创建 TXT 记录 AWS 管理控制台，请参阅使用 Amazon Route 53 控制台创建记录。

使用 Route 53 创建 TXT 记录 AWS CLI

1. 使用带有以下示例TXT-record.json文件的change-resource-record-sets命令：

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

2. 使用以下 AWS CLI 命令将上一步中的 TXT 记录添加到 Route 53 托管区域：

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

将hosted-zone-id替换为您账户中托管区域的 Route 53 托管区域 ID。change-batch 参数值指向文件夹 () 中的 JSON 文件 (txt-Record.json)。path/to/your

要检查您的域名的验证状态，您可以使用 VPC Lattice 控制台或get-domain-verification命令。

验证域名后，域名将一直处于验证状态，直到您将其删除。如果您从 DNS 区域中删除 TXT 记录，VPC Lattice 会删除，您需要重新验证域名。verification-id如果您删除 DNS 区域中的 TXT 记录，VPC Lattice 会将您的域名验证状态设置为UNVERIFIED。这不会影响任何现有资源终端节点、服务网络终端节点或服务网络 VPC 与您的资源配置的关联。要重新验证您的域名，请重新开始域名验证流程。

在 VPC Lattice 中创建资源配置

创建资源配置。

AWS 管理控制台

使用控制台创建资源配置

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格的PrivateLink 和莱迪思下，选择资源配置。
3. 选择创建资源配置。
4. 输入一个在您的 AWS 账户中唯一的名称。在创建资源配置后无法更改此名称。
5. 对于配置类型，为单个资源或子资源选择资源，为一组子资源选择资源组。
6. 选择您之前创建的资源网关或立即创建一个资源网关。
7. （可选）要输入自定义域名，请执行以下任一操作：
   - 如果您的资源配置类型为 single，则可以输入自定义域名。资源使用者可以使用此域名来访问您的资源配置。
   - 如果您的资源配置类型为 group 和 child，则必须先在组资源配置中指定组域。接下来，子资源配置可以有自定义域，这些域名是组域的子域。
8. （可选）输入验证 ID。

   如果您想验证您的域名，请提供验证 ID。这可以让资源使用者知道您拥有域名。
9. 选择您希望此资源配置代表的资源的标识符。
10. 选择与您要通过其共享资源的端口范围。
11. 对于关联设置，指定此资源配置是否可以与可共享的服务网络相关联。
12. 对于共享资源配置，请选择标识可以访问此资源的主体的资源共享。
13. （可选）对于监控，如果您要监控资源配置的请求和响应，请启用资源访问日志和传输目的地。
14. （可选）若要添加标签，请选择添加新标签，然后输入该标签的键和值。
15. 选择创建资源配置。

AWS CLI

以下create-resource-configuration命令创建单个资源配置并将其与自定义域名关联`example.com`。

```
aws vpc-lattice create-resource-configuration \
    --name my-resource-config \
```

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

以下create-resource-configuration命令创建群组资源配置并将其与自定义域名关联example.com。

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

以下create-resource-configuration命令创建子资源配置并将其与自定义域名关联child.example.com。

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

管理 VPC Lattice 资源配置的关联

与您共享资源配置的消费者账户以及您账户中的客户可以直接使用资源类型的 VPC 终端节点或通过服务网络类型的 VPC 终端节点访问资源配置。因此，您的资源配置将具有端点关联和服务网络关联。

管理服务网络资源关联

创建或删除服务网络关联。

> **Note**
>
> 如果您在创建服务网络和资源配置之间的关联时收到拒绝访问的消息，请检查您的 AWS RAM 策略版本并确保其为版本 2。有关更多信息，请参阅AWS RAM 用户指南。

使用控制台管理服务网络关联

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格的PrivateLink 和莱迪思下，选择资源配置。
3. 选择资源配置名称以打开其详细信息页面。
4. 选择服务网络关联选项卡。
5. 选择创建关联。
6. 从 VPC Lattice 服务网络中选择一个服务网络。要创建服务网络，请选择创建 VPC Lattice 网络。
7. （可选）要添加标签，请展开服务关联标签，选择添加新标签，然后输入标签键和标签值。
8. （可选）要为此服务网络资源关联启用私有 DNS 名称，请选择启用私有 DNS 名称。有关更多信息，请参阅 the section called “服务网络所有者的自定义域名”。
9. 选择保存更改。
10. 要删除关联，请选中关联的复选框，然后依次选择操作和删除。提示进行确认时，输入 **`confirm`**，然后选择删除。

要使用创建服务网络关联 AWS CLI

使用 create-service-network-resource-association 命令。

要删除服务网络关联，请使用 AWS CLI

使用 delete-service-network-resource-association 命令。

管理资源 VPC 终端节点关联

有权访问您的资源配置的消费者账户或您账户中的客户可以使用资源 VPC 终端节点访问资源配置。如果您的资源配置具有自定义域名，则可以使用启用私有 DNS 来允许 VPC Lattice 为您的资源终端节点或服务网络终端节点配置私有托管区域。这样，客户端就可以直接卷曲域名来访问资源配置。有关更多信息，请参阅 the section called “资源使用者的自定义域名”。

AWS 管理控制台

1. 要创建新的端点关联，请转到左侧导航窗格中的PrivateLink 和莱迪思，然后选择终端节点。
2. 选择创建端点。
3. 选择要连接到 VPC 的资源配置。
4. 选择 VPC、子网和安全组。

5. （可选）要打开私有 DNS 并配置 DNS 选项，请选择启用私有 DNS 名称。
6. （可选）要向 VPC 端点添加标签，请选择添加新标签，然后输入标签键和标签值。
7. 选择创建端点。

AWS CLI

以下create-vpc-endpoint命令创建使用私有 DNS 的 VPC 终端节点。私有 DNS 首选项设置为，VERIFIED_AND_SELECTED所选域名设置为example.com和example.org。VPC Lattice 仅为任何经过验证的域名或example.com或example.org提供私有托管区域。

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

使用创建 VPC 终端节点关联 AWS CLI

使用 create-vpc-endpoint 命令。

要删除 VPC 终端节点关联，请使用 AWS CLI

使用 delete-vpc-endpoint 命令。

共享 VPC Lattice 实体

Amazon VPC Lattice 与 AWS Resource Access Manager (AWS RAM) 集成，可实现服务、资源配置和服务网络的共享。 AWS RAM 是一项服务，可让您与其他实体 AWS 账户 或通过 AWS Organizations其他实体共享某些VPC Lattice实体。使用 AWS RAM，您可以通过创建资源共享来共享您拥有的实体。资源共享指定要共享的实体以及要与之共享的消费者。使用者可包括：

- 具体在其组织 AWS 账户 内部或外部 AWS Organizations。
- AWS Organizations中其组织内部的组织单元。
- AWS Organizations中的整个组织。

有关的更多信息 AWS RAM，请参阅《AWS RAM 用户指南》。

内容

共享 VPC 莱迪思实体的先决条件

- 要共享实体，您必须在自己的实体中拥有该实体 AWS 账户。这意味着必须在您的账户中分配或配置该实体。您不能共享已与您共享的实体。
- 要与您的组织或中的组织单位共享实体 AWS Organizations，必须启用与共享 AWS Organizations。有关更多信息，请参阅《AWS RAM 用户指南》中的允许在 AWS Organizations内共享资源。

共享 VPC 莱迪思实体

要共享实体，请先使用创建资源共享 AWS Resource Access Manager。资源共享指定要共享的实体、与之共享的使用者以及委托人可以执行的操作。

当您与其他人共享您拥有的VPC Lattice实体时 AWS 账户，您可以允许这些账户将其实体与您账户中的实体相关联。当您针对共享实体创建关联时，我们会在实体所有者账户和创建该关联的账户中生成一个 Amazon 资源名称 (ARN)。因此，实体所有者和创建关联的账户都可以删除关联。

如果您是组织中的一员， AWS Organizations 并且启用了组织内部共享，则会自动授予组织中的消费者访问该共享实体的权限。否则，消费者会收到加入资源共享的邀请，并在接受邀请后被授予访问共享实体的权限。

注意事项

- 您可以共享三种类型的VPC Lattice实体：服务网络、服务和资源配置。
- 您可以与任何 AWS 账户人共享您的VPC莱迪思实体。
- 您不能与个人 IAM 用户和角色共享您的 VPC 莱迪思实体。
- VPC Lattice支持客户管理服务、资源配置和服务网络的权限。

使用VPC Lattice控制台共享您拥有的实体

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格的 VPC Lattice 下，选择服务、服务网络或资源配置。
3. 选择实体名称以打开其详细信息页面，然后从 “共享” 选项卡中选择 “共享服务”、“共享服务网络” 或 “共享资源配置”。
4. 从 AWS RAM 资源共享中选择资源共享。要创建资源共享，请选择在 RAM 控制台中创建资源共享。
5. 选择 “共享服务”、“共享服务网络” 或 “共享资源配置”。

使用 AWS RAM 控制台共享您拥有的实体

按照《AWS RAM 用户指南》中的创建资源共享中描述的过程操作。

要共享您拥有的实体，请使用 AWS CLI

使用 associate-resource-share 命令。

停止共享 VPC 莱迪思实体

要停止共享您拥有的 VPC Lattice 实体，必须将其从资源共享中移除。停止共享实体后，现有关联仍然存在。不允许与先前共享的实体建立新关联。当实体所有者或协会所有者删除关联时，该关联将从两个

账户中删除。如果账户所有者想要离开资源共享，他们必须要求资源共享的所有者将其账户从与之共享该资源的账户列表中删除。

使用 VPC Lattice 控制台停止共享您拥有的实体

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格的 VPC Lattice 下，选择服务、服务网络或资源配置。
3. 选择实体的名称以打开其详细信息页面。
4. 在共享选项卡上，选中资源共享的复选框，然后选择删除。

使用 AWS RAM 控制台停止共享您拥有的实体

请参阅《AWS RAM 用户指南》中的更新资源共享。

要停止共享您拥有的实体，请使用 AWS CLI

使用 disassociate-resource-share 命令。

责任和权限

使用共享 VPC Lattice 实体时，以下责任和权限适用。

实体所有者

- 服务网络所有者不能修改使用者创建的服务。
- 服务网络所有者不能删除使用者创建的服务。
- 服务网络所有者可以描述服务网络的所有服务关联。
- 服务网络所有者可以解除与服务网络关联的任何服务的关联，无论创建关联的人员为何。
- 服务网络所有者可以描述服务网络的所有 VPC 关联。
- 服务网络所有者可以解除使用者与服务网络关联的任何 VPC 的关联。
- 服务网络所有者可以描述服务网络的所有资源配置关联。
- 无论谁创建了关联，服务网络所有者都可以取消与服务网络关联的任何资源配置的关联。
- 服务网络所有者可以描述服务网络的所有端点关联。
- 无论谁创建了关联，服务网络所有者都可以取消与服务网络关联的任何端点的关联。
- 服务所有者可以描述与服务的所有服务网络关联。

- 服务所有者可将服务从与服务关联的任何服务网络中取消关联
- 资源配置所有者可以描述与资源配置的所有网络关联。
- 资源配置所有者可以取消资源配置与其关联的任何服务网络的关联。
- VPC 终端节点所有者可以描述与其关联的服务网络。
- VPC 终端节点所有者可以解除终端节点与服务网络的关联。
- 只有创建关联的账户才能更新服务网络和 VPC 之间的关联。

实体消费者

- 使用者无法删除他们未创建的服务或资源配置。
- 使用者只能取消与服务网络关联的服务或资源配置的关联。
- 消费者和网络所有者可以描述服务网络与服务或资源配置之间的所有关联。
- 使用者无法检索服务的服务信息或他们不拥有的资源配置的资源配置信息。
- 使用者可以描述与共享服务网络的所有服务关联和资源配置关联。
- 使用者可以将服务或资源配置与共享服务网络相关联。
- 使用者可以看到与共享服务网络的所有 VPC 关联。
- 使用者可以将 VPC 与共享服务网络关联。
- 消费者只能取消与服务网络关联 VPCs 的关联。
- 使用者可以创建服务网络 VPC 终端节点，将其的 VPC 连接到共享服务网络。
- 使用者只能删除他们为将其 VPC 连接到共享服务网络而创建的服务网络 VPC 终端节点。
- 共享服务的使用者无法将服务与非其拥有的服务网络关联。
- 共享服务网络的使用者无法关联非其拥有的 VPC 或服务。
- 共享资源配置的使用者无法将资源配置与他们不拥有的服务网络相关联。
- 共享服务网络的使用者无法关联他们不拥有的 VPC 或服务或资源配置。
- 消费者可以描述与他们共享的服务、服务网络或资源配置。
- 如果两个实体都与消费者共享，则消费者无法将其关联。

跨账户事件

当实体所有者和消费者对共享实体执行操作时，这些操作将在中 AWS CloudTrail记录为跨账户事件。

`CreateServiceNetworkResourceAssociationBySharee`

当实体消费者与共享实体调CreateServiceNetworkResourceAssociation 用时，发送给实体所有者。如果调用者拥有资源配置，则该事件将发送给服务网络的所有者。如果调用方拥有服务网络，则该事件将发送给资源配置的所有者。

`CreateServiceNetworkServiceAssociationBySharee`

当实体消费者与共享实体调CreateServiceNetworkServiceAssociation用时，发送给实体所有者。如果调用方拥有该服务，则事件将发送给服务网络所有者。如果调用方拥有该服务网络，则事件将发送给服务所有者。

`CreateServiceNetworkVpcAssociationBySharee`

当实体消费者通过共享服务网络呼叫CreateServiceNetworkVpcAssociation时，发送给实体所有者。

`DeleteServiceNetworkResourceAssociationByOwner`

当实体所有者DeleteServiceNetworkResourceAssociation 与共享实体通话时发送给关联所有者。如果调用者拥有资源配置，则该事件将发送给服务网络关联的所有者。如果调用者拥有服务网络，则事件将发送给资源关联的所有者。

`DeleteServiceNetworkResourceAssociationBySharee`

当实体消费者与共享实体调DeleteServiceNetworkResourceAssociation 用时，发送给实体所有者。如果调用者拥有资源配置，则该事件将发送给服务网络的所有者。如果调用方拥有服务网络，则该事件将发送给资源配置的所有者。

`DeleteServiceNetworkServiceAssociationByOwner`

当实体所有者DeleteServiceNetworkServiceAssociation与共享实体通话时发送给关联所有者。如果调用方拥有该服务，则事件将发送给服务网络关联所有者。如果调用方拥有该服务网络，则事件将发送给服务关联所有者。

`DeleteServiceNetworkServiceAssociationBySharee`

当实体消费者与共享实体调DeleteServiceNetworkServiceAssociation用时，发送给实体所有者。如果调用方拥有该服务，则事件将发送给服务网络所有者。如果调用方拥有该服务网络，则事件将发送给服务所有者。

`DeleteServiceNetworkVpcAssociationByOwner`

当实体所有者通过共享服务网络呼叫DeleteServiceNetworkVpcAssociation时，发送给协会所有者。

`DeleteServiceNetworkVpcAssociationBySharee`

当实体消费者通过共享服务网络呼叫DeleteServiceNetworkVpcAssociation时，发送给实体所有者。

`GetServiceBySharee`

当实体消费者使用共享服务调GetService用时，发送给实体所有者。

`GetServiceNetworkBySharee`

当实体消费者通过共享服务网络呼叫GetServiceNetwork时，发送给实体所有者。

`GetServiceNetworkResourceAssociationBySharee`

当实体消费者与共享实体调GetServiceNetworkResourceAssociation 用时，发送给实体所有者。如果调用者拥有资源配置，则该事件将发送给服务网络的所有者。如果调用方拥有服务网络，则该事件将发送给资源配置的所有者。

`GetServiceNetworkServiceAssociationBySharee`

当实体消费者与共享实体调GetServiceNetworkServiceAssociation用时，发送给实体所有者。如果调用方拥有该服务，则事件将发送给服务网络所有者。如果调用方拥有该服务网络，则事件将发送给服务所有者。

`GetServiceNetworkVpcAssociationBySharee`

当实体消费者通过共享服务网络呼叫GetServiceNetworkVpcAssociation时，发送给实体所有者。

以下是 `CreateServiceNetworkServiceAssociationBySharee` 事件的示例条目。


```
{
    "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"
}
```

VPC Lattice 适用于 Oracle Database@AWS

VPC Lattice 为 Oracle Database@AWS(ODB) 的 AWS 托管服务集成提供支持，并为您简化了 ODB 网络和内部部署之间的连接。 AWS VPCs 为了支持这种连接，VPC Lattice 代表您配置了以下实体：

默认服务网络

默认服务网络使用命名约定 `default-odb-network-randomHash`

默认服务网络端点

此 AWS 资源没有名称。

资源网关

资源网关使用命名约定 `default-odb-network-randomHash`

VPC Lattice支持 AWS 托管服务集成，即与您的ODB网络的托管集成。默认情况下，Oracle 云基础设施 (OCI) Amazon S3 托管备份处于启用状态。您可以选择启用对 Amazon S3 和零 ETL 的自我管理访问权限。

创建 ODB 网络后，您可以使用或查看已配置的 AWS 管理控制台 资源。 AWS CLI以下示例命令列出了 ODB 网络的默认托管集成以及该服务网络可能拥有的任何其他资源：

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

注意事项

以下注意事项适用于VPC Lattice Oracle Database@AWS：

- 您无法删除 VPC Lattice 配置的默认服务网络、服务网络终端节点、资源网关或任何 ODB 托管集成。要删除这些实体，请删除您的 ODB 网络或禁用托管集成。
- 客户端只能访问 ODB 网络中的托管集成。ODB 网络之外的客户端（例如您的 VPCs客户端）无法使用这些托管集成来访问 S3 或 Zero-ETL。
- 您无法连接到 VPC Lattice 配置的 ODB 网络之外的任何托管集成。
- Amazon S3 的所有流量都通过默认的服务网络终端节点，访问资源将收取标准处理费。所有零 ETL 流量都将通过资源网关，您共享的资源将收取标准数据处理费用。有关更多信息，请参阅 VPC 莱迪思定价。

- Oracle Database@AWS 托管集成不收取每小时费用。
- 您可以像管理任何其他服务网络一样管理VPC Lattice提供的资源。您可以与其他 AWS 账户 或组织共享默认服务网络，并在默认网络中添加新的终端节点、VPC 关联、VPC Lattice 服务和资源。
- VPC Lattice 需要以下权限才能配置 Oracle Database@AWS 资源：

  JSON

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

```
  "Resource": "*",
   "Condition": {
    "StringEquals": {
     "iam:AWSServiceName": [
      "vpc-lattice.amazonaws.com"
     ]
    }
   }
 }
  ]
}
```

要使用 VPC Lattice Oracle Database@AWS，我们建议您熟悉 VPC Lattice 中的服务网络、服务网络关联和资源网关。

主题

- the section called “Oracle 云基础设施 (OCI) Amazon S3 托管备份”
- the section called “Amazon S3 访问权限”
- the section called “亚马逊 Redshift 的零 ETL”
- the section called “访问和共享 VPC 莱迪思实体”

Oracle 云基础设施 (OCI) Amazon S3 托管备份

当您创建 Oracle Database@AWS 数据库时，VPC Lattice 会创建一个名odb-managed-s3-backup-access为的资源配置。此资源配置表示您的数据库到 Amazon S3 的 OCI 托管备份，并且仅支持与 OCI 拥有的 Amazon S3 存储桶的连接。ODB 网络和 S3 之间的流量永远不会离开 Amazon 网络。

Amazon S3 访问权限

除了 OCI 托管备份到 Amazon S3 之外，您还可以创建托管集成，允许从 ODB 网络访问 Amazon S3。当您修改 Oracle Database@AWS 网络以启用 Amazon S3 Access 托管集成时，VPC Lattice 会在默认服务网络odb-s3-access中配置一个名为的资源配置。您可以使用此集成来访问 Amazon S3 以满足自己的需求，包括自行管理的备份或恢复。您可以通过提供身份验证策略来建立边界控制。

注意事项

以下是 Amazon S3 Access 托管集成的注意事项：

- 您只能为 ODB 网络创建一个 Amazon S3 Access 托管集成。
- 这种托管集成仅允许从 ODB 网络访问 Amazon S3，而不能从默认服务网络中的其他 VPC 关联或服务网络终端节点访问 Amazon S3。
- 您无法访问不同 AWS 区域的 S3 存储桶。

启用 Amazon S3 访问托管集成

使用以下命令启用 Amazon S3 Access 托管集成：

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

使用身份验证策略进行安全访问

您可以使用 ODB API 定义身份验证策略，从而保护对 S3 存储桶的访问。以下示例策略授予对特定组织拥有的特定 S3 存储桶的访问权限。

JSON

```
{
  "Version":"2012-10-17",
  "Id": "Policy1515115909152",
  "Statement": [
    {
      "Sid": "GrantAccessToMyOrgS3",
      "Principal": "*",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:s3:::awsexamplebucket1",
        "arn:aws:s3:::awsexamplebucket1/*"
      ],
      "Condition": {
```

```
        "StringNotEquals": {
          "aws:ResourceOrgID": "o-abcd1234"
        }
      }
    }
  ]
}
```

> **Note**
>
> 使用 ODB 托管集成时 `aws:SourceVpcaws:SourceVpce`，S3 存储桶策略不支持、和`aws:VpcSourceIp`条件键。

亚马逊 Redshift 的零 ETL

您可以使用 VPC Lattice 提供的服务网络来启用零 ETL。这种托管集成可将您的 ODB 网络数据库连接到 Amazon Redshift，以帮助分析不同数据库中的数据。您可以使用 AWS Glue 集成启动零 ETL 设置， APIs 并使用 ODB APIs 打开托管集成并设置网络路径。有关更多信息，请参阅与 Amazon Redshift 的零 ETL 集成。

注意事项

以下是托管零 ETL 集成的注意事项：

- 如果您启用托管的零 ETL 集成，则只能使用零 ETL 来访问 ODB 网络中的实例。与您的服务网络关联的其他服务和资源与 Zero-ETL 隔离。

访问和共享 VPC 莱迪思实体

您还可以 VPCs 使用 VPC Lattice 将您的 ODB 网络连接到服务、资源和其他客户端。这些连接选项由 VPC Lattice提供的默认服务网络、资源网关和服务网络终端节点提供支持。

访问莱迪思VPC服务和资源

要访问其他实体，请将您拥有或与您共享的服务或资源关联到默认服务网络。ODB 网络中的客户端可以通过默认服务网络端点访问服务或资源。

注意事项

以下是连接到其他 VPC 莱迪思实体的注意事项：

- 您可以向服务网络添加新的服务网络终端节点、VPC 关联、VPC Lattice 资源和服务，但不能修改由 VPC Lattice 代表 ODB 网络配置的资源。这些必须通过进行管理 Oracle Database@AWS APIs。

通过 VPC 莱迪思共享您的 ODB 网络

您可以与其他客户 VPCs、账户或本地客户共享您的 ODB 网络资源。首先，请为要共享的资源创建资源配置。资源配置必须使用 ODB 网络的默认资源网关。然后，您可以将资源与您的默认服务网络相关联。

其他客户 VPCs 或 AWS 账户 与您共享服务网络的客户可以通过自己的服务网络终端节点或 VPC 关联访问这些资源。有关更多信息，请参阅 the section called “管理关联”。

注意事项

以下是共享 ODB 网络的注意事项：

- 我们建议仅将 ODB 网络实例作为基于 IP 的资源共享。
- VPC Lattice 不支持 OCI 的单客户端访问名称 (SCAN) 侦听器 DNS。

Amazon VPC Lattice 中的安全性

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

您负责维护对托管在此基础结构上的内容的控制。责任共担模式将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在云中运行 AWS 服务的基础架构 AWS Cloud。 AWS 还为您提供可以安全使用的服务。 Third-party 作为AWS 合规计划合规计划合规计划合的一部分，审计师定期测试和验证我们安全的有效性。要了解适用于 Amazon VPC Lattice 的合规计划，请参阅AWS 按合规计划划分的范围内AWS 服务按合规计划。
- 云中的安全性：您负责维护对托管在此基础设施上的内容进行控制。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

本文档可帮助您了解在使用 VPC Lattice 时如何应用分担责任模式。以下主题展示了如何配置 VPC Lattice，以满足您的安全性和合规性目标。您还将学习如何使用其他 AWS 服务，这些服务可帮助您监控和保护您的VPC Lattice服务、服务网络和资源配置。

内容

管理对VPC莱迪思服务的访问权限

默认情况下，VPC Lattice是安全的，因为您必须明确提供访问哪些服务和资源配置以及哪些VPC的访问权限。您可以通过 VPC 关联或服务网络类型的 VPC 终端节点访问服务。对于多账户方案，您可以使用AWS Resource Access Manager跨账户边界共享服务、资源配置和服务网络。

VPC Lattice 提供了一个框架，让您可以在网络的多个层实施 defense-in-depth 策略。

- 第一层 — 服务、资源、VPC 和 VPC 终端节点与服务网络的关联。VPC 可以通过关联或通过 VPC 终端节点连接到服务网络。如果 VPC 未连接到服务网络，则 VPC 中的客户端将无法访问与服务网络关联的服务和资源配置。
- 第二层：为服务网络提供可选的网络级安全保护，例如安全组和网络 ACL。通过使用这些，您可以允许访问 VPC 中的特定客户端组，而不是 VPC 中的所有客户端。
- 第三层：可选的 VPC Lattice 验证策略。您可以将验证策略应用于服务网络和单个服务。通常，服务网络上的验证策略由网络或云管理员操作，以实现粗粒度授权。例如，仅允许来自 AWS Organizations中特定组织的经过验证的请求。对于服务级别的验证策略，服务所有者通常会设置细粒度控制，此类控制可能比在服务网络级别应用的粗粒度授权更具限制性。

> **Note**
>
> 服务网络上的身份验证策略不适用于服务网络中的资源配置。

访问控制方法

- 验证策略
- 安全组
- 网络 ACL

使用身份验证策略控制对VPC莱迪思服务的访问

VPC Lattice 验证策略是 IAM policy 文档，您可以将其附加到服务网络或服务，以控制指定主体是否可以访问一组服务或特定服务。您可以将一个验证策略附加到您要控制访问的每个服务网络或服务。

> **Note**
>
> 服务网络上的身份验证策略不适用于服务网络中的资源配置。

验证策略不同于 IAM 基于身份的策略。IAM 基于身份的策略附加到 IAM 用户、组或角色，并定义这些身份可以对哪些资源执行哪些操作。验证策略附加到服务和服务网络。要使授权成功，验证策略和基于身份的策略都需要具有显式允许语句。有关更多信息，请参阅 授权的工作原理。

您可以使用 AWS CLI 和控制台查看、添加、更新或删除服务和服务网络上的身份验证策略。添加、更新或移除身份验证策略时，可能需要几分钟才能准备就绪。使用时 AWS CLI，请确保您位于正确的区域。您可以更改个人资料的默认区域，也可以将`--region`参数与命令一起使用。

内容

- 验证策略中的常用元素
- 验证策略的资源格式
- 可在验证策略中使用的条件键
- 资源标签
- 校长标签
- 匿名（未经验证）主体
- 示例验证策略
- 授权的工作原理

要开始使用验证策略，请按照以下过程创建适用于服务网络的验证策略。对于您不想应用于其他服务的更具限制性的权限，您可以选择在单个服务上设置验证策略。

使用验证策略管理对服务网络的访问

以下 AWS CLI 任务向您展示如何使用身份验证策略管理对服务网络的访问权限。有关使用控制台的说明，请参阅 VPC Lattice 中的服务网络。

任务

- 向服务网络添加验证策略
- 更改服务网络的验证类型
- 从服务网络中删除验证策略

向服务网络添加验证策略

按照本节中的步骤 AWS CLI 使用：

- 使用 IAM 在服务网络上启用访问控制。
- 向服务网络添加验证策略。如果不添加验证策略，则所有流量都将收到拒绝访问错误。

要启用访问控制并向新服务网络添加验证策略

1. 要在服务网络上启用访问控制，使服务网络能够使用验证策略，请使用具有 `--auth-type` 选项和值为 `AWS_IAM` 的 create-service-network 命令。

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

如果成功，该命令将返回类似于以下内容的输出。

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

2. 使用 put-auth-policy 命令，指定要在其中添加验证策略的服务网络 ID，以及要添加的验证策略。

例如，使用以下命令为 ID 为 *sn-0123456789abcdef0* 的服务网络创建验证策略。

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

使用 JSON 创建策略定义。有关更多信息，请参阅 验证策略中的常用元素。

如果成功，该命令将返回类似于以下内容的输出。

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

要启用访问控制并向现有服务网络添加验证策略

1. 要在服务网络上启用访问控制，使服务网络能够使用验证策略，请使用具有 `--auth-type` 选项和值为 `AWS_IAM` 的 update-service-network 命令。

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

如果成功，该命令将返回类似于以下内容的输出。

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

2. 使用 put-auth-policy 命令，指定要在其中添加验证策略的服务网络 ID，以及要添加的验证策略。

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

使用 JSON 创建策略定义。有关更多信息，请参阅 验证策略中的常用元素。

如果成功，该命令将返回类似于以下内容的输出。

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

更改服务网络的验证类型

要禁用服务网络的验证策略

使用具有 --auth-type 选项和值为 NONE 的 update-service-network 命令。

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

如果之后需要再次启用验证策略，请使用为 --auth-type 选项指定的 AWS_IAM 运行此命令。

从服务网络中删除验证策略

要从服务网络中删除验证策略

使用 delete-auth-policy 命令。

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

如果在将服务网络的验证类型更改为 NONE 之前删除验证策略，则请求会失败。

使用验证策略管理对服务的访问

以下 AWS CLI 任务向您展示如何使用身份验证策略管理对服务的访问权限。有关使用控制台的说明，请参阅 VPC Lattice 中的服务。

任务

- 向服务添加验证策略
- 更改服务的验证类型
- 从服务中删除验证策略

向服务添加验证策略

请按照以下步骤 AWS CLI 使用：

- 使用 IAM 在服务上启用访问控制。
- 向服务添加验证策略。如果不添加验证策略，则所有流量都将收到拒绝访问错误。

要启用访问控制并向新服务添加验证策略

1. 要在服务上启用访问控制，使服务能够使用验证策略，请使用具有 `--auth-type` 选项和值为 `AWS_IAM` 的 create-service 命令。

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

   如果成功，该命令将返回类似于以下内容的输出。

   ```
   {
      "arn": "arn",
      "authType": "AWS_IAM",
      "dnsEntry": {
         ...
   ```

```
    },
    "id": "svc-0123456789abcdef0",
    "name": "Name",
    "status": "CREATE_IN_PROGRESS"
}
```

2. 使用 put-auth-policy 命令，指定要在其中添加验证策略的服务 ID，以及要添加的验证策略。

   例如，使用以下命令为具有 ID *svc-0123456789abcdef0* 的服务创建身份验证策略。

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

   使用 JSON 创建策略定义。有关更多信息，请参阅 验证策略中的常用元素。

   如果成功，该命令将返回类似于以下内容的输出。

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

要启用访问控制并向现有服务添加验证策略

1. 要在服务上启用访问控制，使服务能够使用验证策略，请使用具有 `--auth-type` 选项和值为 `AWS_IAM` 的 update-service 命令。

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

   如果成功，该命令将返回类似于以下内容的输出。

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

2. 使用 put-auth-policy 命令，指定要在其中添加验证策略的服务 ID，以及要添加的验证策略。

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

使用 JSON 创建策略定义。有关更多信息，请参阅 验证策略中的常用元素。

如果成功，该命令将返回类似于以下内容的输出。

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

更改服务的验证类型

要禁用服务的验证策略

使用具有 --auth-type 选项和值为 NONE 的 update-service 命令。

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

如果之后需要再次启用验证策略，请使用为 --auth-type 选项指定的 AWS_IAM 运行此命令。

从服务中删除验证策略

要从服务中删除验证策略

使用 delete-auth-policy 命令。

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

如果在将服务的验证类型更改为 NONE 之前删除验证策略，则请求会失败。

如果启用验证策略，该验证策略需要对服务发出经过验证的请求，则对服务的任何请求都必须包含使用签名版本 4（SigV4）计算的有效请求签名。有关更多信息，请参阅 通过 Sigv4 身份验证的 Amazon VPC Lattice 请求。

验证策略中的常用元素

指定 VPC Lattice 验证策略所用的语法与 IAM 策略相同。有关更多信息，请参阅 IAM 用户指南中的Identity-based策略和基于资源的策略。

验证策略包含以下元素：

- 主体：允许访问语句中的操作和资源的人员或应用程序。在验证策略中，主体是接收此权限的 IAM 实体。主体作为 IAM 实体经过验证，以向特定资源或资源组（如服务网络中的服务）发出请求。

  您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。有关更多信息，请参阅《IAM 用户指南》中的 AWS JSON 策略元素：主体。
- 效果：指定主体请求特定操作时的效果。此值可以是 `Allow` 或 `Deny`。默认情况下，当您使用 IAM 在服务或服务网络上启用访问控制时，主体无权向服务或服务网络发出请求。
- 操作-您授予或拒绝权限的特定 API 操作。VPC Lattice 支持使用`vpc-lattice-svcs`前缀的操作。有关更多信息，请参阅《服务授权参考》中的 Amazon VPC Lattice 服务定义的操作。
- 资源：受操作影响的服务。
- 条件：条件是可选的。您可以使用它们来控制您的策略何时生效。有关更多信息，请参阅《服务授权参考》中的 Amazon VPC Lattice 服务的条件键。

在创建和管理验证策略时，您可能希望使用 IAM Policy Generator。

要求

JSON 中的策略不得包含换行符或空行。

验证策略的资源格式

您可以通过创建一个验证策略来限制对特定资源的访问，该策略使用具有 `<serviceARN>/<path>` 模式的匹配架构，并对 `Resource` 元素进行编码，如以下示例所示。

协议	示例
HTTP	• `"Resource": "arn:aws:vpc-lattice:us-west-2:1234567890:service/svc-0123456789abcdef0/rates"`

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

对 <serviceARN> 使用以下 Amazon 资源名称（ARN）资源格式：

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

例如：

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

可在验证策略中使用的条件键

通过验证策略的 Condition 元素中的条件键，可进一步控制访问。这些条件键是否可用于评估，具体取决于协议以及请求使用的是签名版本 4（SigV4）签名还是匿名签名。条件键区分大小写。

AWS 提供了可用于控制访问权限的全局条件键，例如aws:PrincipalOrgID和aws:SourceIp。要查看全 AWS 局条件键列表，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

下表列出了 VPC Lattice 条件键。有关更多信息，请参阅《服务授权参考》中的 Amazon VPC Lattice 服务的条件键。

条件键	说明	示例	适用于匿名（未经验证）调用方？	适用于 gRPC？
vpc-lattice-svcs:Port	按发出请求的服务端口筛选访问	80	支持	是
vpc-lattice-svcs:RequestMethod	按请求的方式筛选访问权限	GET	是	始终 POST
vpc-lattice-svcs:RequestPath	按请求 URL 的路径部分筛选访问权限	/path	支持	是
vpc-lattice-svcs:RequestHeader/ *header-name* : *value*	按请求标头中的标头名称-值对筛选访问权限	content-type: application/json	支持	是
vpc-lattice-svcs:RequestQueryString/ *key-name*: *value*	按请求 URL 中的查询字符串键值筛选访问权限	quux: [corge, grault]	是	否
vpc-lattice-svcs:ServiceNetworkArn	通过接收请求的服务，其服务网络的 ARN 筛选访问	arn:aws:vpc-lattice:us-west-2:123456789012:servicenetwork/sn-0123456789abcdef0	支持	是
vpc-lattice-svcs:ServiceArn	通过接收请求的服务的 ARN 筛选访问	arn:aws:vpc-lattice:us-west-2:123456	支持	是

条件键	说明	示例	适用于匿名（未经验证）调用方？	适用于 gRPC？
		789012:service/svc-0123456789abcdef0		
vpc-lattice-svcs:SourceVpc	按发出请求的 VPC 筛选访问权限	vpc-1a2b3c4d	支持	是
vpc-lattice-svcs:SourceVpcOwnerAccount	按发出请求的 VPC 的拥有账户筛选访问权限	123456789012	支持	是

资源标签

标签是您分配或分配给 AWS 资源的元数据标签。 AWS 每个 标签具有两个部分：

- 标签键（例如，CostCenter、Environment 或 Project）。标签键区分大小写。
- 一个称为标签值 的可选字段（例如，111122223333 或 Production）。省略标签值与使用空字符串相同。与标签键一样，标签值区分大小写。

有关标记的更多信息，请参阅使用标签控制对 AWS 资源的访问权限

您可以使用aws:ResourceTag/key AWS 全局条件上下文密钥在身份验证策略中使用标签。

以下示例策略授予对带有标签的服务的访问权限Environment=Gamma。此政策允许您引用没有硬编码服务 ARN 或 ID 的服务。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGammaAccess",
      "Effect": "Allow",
      "Principal": "*",
```

```
      "Action": "vpc-lattice-svcs:Invoke",
      "Resource": "arn:aws:vpc-lattice:us-west-2:123456789012:service/
svc-0124446789abcdef0/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Environment": "Gamma",
        }
      }
    }
  ]
}
```

校长标签

您可以根据呼叫者身份所附的标签来控制对服务和资源的访问权限。VPC Lattice 支持使用`aws:PrincipalTag/context`变量根据用户、角色或会话标签上的任何主体标签进行访问控制。有关更多信息，请参阅控制 IAM 主体进行的访问。

以下示例策略仅向带有标签的身份授予访问权限`Team=Payments`。此政策允许您控制访问权限，而无需对账户 ID 或角色 ARN 进行硬编码。

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

匿名（未经验证）主体

匿名委托人是指不使用签名版本 4 (Sigv4) 签署 AWS 请求且位于与服务网络相连的 VPC 内的来电者。如果验证策略允许，匿名主体可以向服务网络中的服务发出未经验证的请求。

示例验证策略

以下是要求由经过验证的主体发出请求的验证策略示例。

所有示例都使用 us-west-2 区域，并包含虚构的账户 ID。

示例 1：限制特定用户访问服务 AWS 组织

以下验证策略示例向任何经过验证的请求授予权限，以访问该策略适用的服务网络中的任何服务。但是，请求必须来自属于条件中指定的 AWS 组织的委托人。

JSON

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

示例 2：限制特定 IAM 角色对服务的访问

以下验证策略示例向任何经过验证的请求授予权限，该请求使用 IAM 角色 `rates-client` 对 `Resource` 元素中指定的服务发出 HTTP GET 请求。`Resource` 元素中的资源与策略附加到的服务相同。

JSON

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

示例 3：限制特定 VPC 中经过验证的主体访问服务

以下验证策略示例仅允许来自 VPC 中主体（VPC ID 为 *vpc-1a2b3c4d*）的经过验证的请求。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
```

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

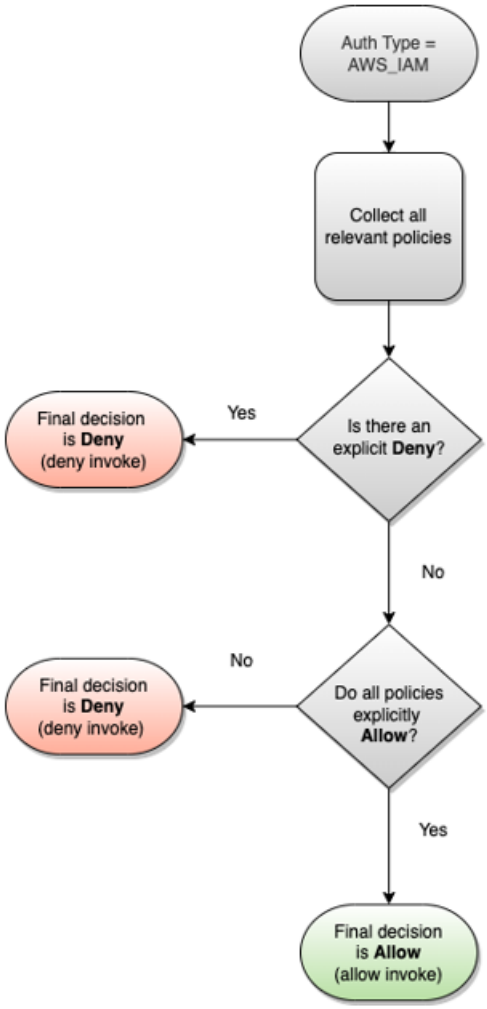
授权的工作原理

当VPC Lattice服务收到请求时， AWS 执行代码会一起评估所有相关的权限策略，以确定是授权还是拒绝该请求。在授权过程中，服务会评估请求上下文中适用的所有 IAM 基于身份的策略和验证策略。默认情况下，当验证类型为 AWS_IAM 时，会隐式拒绝所有请求。所有相关策略的显式允许将覆盖默认值。

授权包括：

- 收集所有相关的 IAM 基于身份的策略和验证策略。
- 评估生成的策略集：
  - 验证请求方（如 IAM 用户或角色）是否有权从请求方所属的账户执行操作。如果没有明确的 allow 语句，则 AWS 不对请求进行授权。
  - 验证服务网络的验证策略是否允许请求。如果启用了身份验证策略，但没有明确的 allow 声明，则 AWS 不对请求进行授权。如果有显式允许语句，或者验证类型为 NONE，则代码将继续。
  - 验证服务的验证策略是否允许请求。如果启用了身份验证策略，但没有明确的 allow 声明，则 AWS 不对请求进行授权。如果有显式允许语句，或者验证类型为 NONE，则执行代码将返回最终决定 Allow。
  - 任何策略中的显式拒绝将覆盖任何允许。

下图显示授权工作流。发出请求时，相关策略允许或拒绝请求访问给定服务。



使用安全组控制 VPC Lattice 中的流量

AWS 安全组充当虚拟防火墙，控制与其关联的实体的进出网络流量。借助 VPC Lattice，您可以创建安全组并将其分配给将 VPC 连接到服务网络的 VPC 关联，从而为您的服务网络实施额外的网络级安全保护。如果您使用 VPC 终端节点将 VPC 连接到服务网络，则也可以为 VPC 终端节点分配安全组。同样，您可以将安全组分配给您创建的资源网关，以允许访问您的 VPC 中的资源。

内容

托管前缀列表

VPC Lattice 提供托管前缀列表，其中包括当您使用服务网络关联通过 VPC 关联将您的 VPC 连接到服务网络时，用于通过 VPC 莱迪思网络路由流量的 IP 地址。这些 IP 要么是私有链路本地 IP，要么是不可路由的公共 IP。

您可以在安全组规则中引用 VPC Lattice 托管前缀列表。此操作允许流量从客户端流出，经过 VPC Lattice 服务网络，流向 VPC Lattice 服务目标。

例如，假设您有一个 EC2 实例注册为美国西部（俄勒冈州）区域（us-west-2）中的目标。您可以向实例安全组添加一条规则，允许来自 VPC Lattice 托管前缀列表的入站 HTTPS 访问，以便该区域中的 VPC Lattice 流量可以到达该实例。如果您从安全组中删除所有其他入站规则，则可以阻止 VPC Lattice 流量以外的任何流量到达实例。

VPC Lattice 的托管前缀列表名称如下：

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

有关更多信息，请参阅《Amazon VPC 用户指南》中的 AWS托管的前缀列表。

Windows 和 macOS 客户端

VPC Lattice 前缀列表中的地址是链路本地地址和不可路由的公有地址。如果您从这些客户端连接到 VPC Lattice，则必须更新其配置，以便将托管前缀列表中的 IP 地址转发到客户端的主要 IP 地址。以下是更新 Windows 客户端配置的命令示例，其中169.254.171.0是托管前缀列表中的地址之一。

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

以下是更新 macOS 客户端配置的命令示例，其中 macOS 客户端169.254.171.0是托管前缀列表中的地址之一。

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

为避免创建静态路由，我们建议您使用 VPC 中的服务网络终端节点来建立连接。有关更多信息，请参阅 the section called “管理服务网络 VPC 终端节点关联”。

安全组规则

无论有没有安全组，使用 VPC Lattice 都不会影响现有的 VPC 安全组配置。但您可以随时添加自己的安全组。

重要注意事项

- 客户端的安全组规则控制VPC Lattice的出站流量。
- 目标的安全组规则控制从 VPC Lattice 到目标的入站流量，包括运行状况检查流量。
- 服务网络和 VPC 之间关联的安全组规则，控制哪些客户端可以访问 VPC Lattice 服务网络。
- 资源网关的安全组规则控制从资源网关到资源的出站流量。

从资源网关流向数据库资源的流量推荐的出站规则

要使流量从资源网关流向资源，必须为开放端口创建出站规则，为资源创建接受的侦听器协议。

目标位置	协议	端口范围	Comment
CIDR range for resource	*TCP*	*3306*	允许从资源网关到数据库的流量

针对服务网络和 VPC 关联的推荐入站规则

要使流量从客户端 VPC 流向与服务网络关联的服务，必须为侦听器端口创建入站规则，为服务创建监听器协议。

来源	协议	端口范围	Comment
VPC CIDR	*listener*	*listener*	允许来自客户端的流量到莱迪思VPC

推荐出站规则，针对从客户端实例流向 VPC Lattice 的流量

默认情况下，安全组允许所有出站流量。但是，如果您有自定义出站规则，则必须允许出站流量到VPC Lattice前缀作为监听器端口和协议，以便客户端实例可以连接到与VPC Lattice服务网络关联的所有服务。您可以通过引用 VPC Lattice 的前缀列表的 ID 来允许此流量。

目标位置	协议	端口范围	Comment
ID of the VPC Lattice prefix list	*listener*	*listener*	允许来自客户端的流量到莱迪思VPC

推荐入站规则，针对从 VPC Lattice 流向目标实例的流量

由于流量来自 VPC Lattice，因此您不能将客户端安全组用作目标安全组的源。您可以引用 VPC 莱迪思的前缀列表的 ID。

来源	协议	端口范围	Comment
ID of the VPC Lattice prefix list	*target*	*target*	允许从 VPC 莱迪思到目标的流量
ID of the VPC Lattice prefix list	*health check*	*health check*	允许从 VPC 莱迪思到目标的运行状况检查流量

管理 VPC 关联的安全组

您可以使用查看、添加或更新 VPC 上的安全组与服务网络关联。 AWS CLI 使用时 AWS CLI，请记住您的命令在 AWS 区域 配置文件中运行。如果您想要在不同的区域中运行命令，可以为配置文件更改默认区域，或者与命令一起使用 `--region` 参数。

开始之前，确认您已在与要添加到服务网络的 VPC 相同的 VPC 中创建安全组。有关更多信息，请参阅 Amazon VPC 用户指南中的使用安全组控制资源流量

要使用控制台在创建 VPC 关联时添加安全组

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务网络。
3. 选择服务网络名称以打开其详细信息页面。
4. 在 VPC 关联选项卡上，选择创建 VPC 关联，然后选择添加 VPC 关联。

5. 选择一个 VPC 和最多 5 个安全组。
6. 选择保存更改。

要使用控制台为现有 VPC 关联添加或更新安全组

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 在导航窗格中的 VPC Lattice 下，选择服务网络。
3. 选择服务网络名称以打开其详细信息页面。
4. 在 VPC 关联选项卡上，选中关联的复选框，然后依次选择操作和编辑安全组。
5. 根据需要添加和删除安全组。
6. 选择保存更改。

在创建 VPC 关联时使用添加安全组 AWS CLI

使用 create-service-network-vpc-association 命令，指定 VPC 关联的 VPC ID 和要添加的安全组 ID。

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

如果成功，该命令将返回类似于以下内容的输出。

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

要为现有 VPC 关联添加或更新安全组，请使用 AWS CLI

使用 update-service-network-vpc-association 命令，指定服务网络 ID 和安全组 ID。这些安全组会覆盖先前关联的任何安全组。更新列表时，至少定义一个安全组。

```
aws vpc-lattice update-service-network-vpc-association
    --service-network-vpc-association-identifier sn-903004f88example \
```

```
    --security-group-ids sg-7c2270198example sg-903004f88example
```

> **Warning**
>
> 您无法删除所有安全组。您必须先删除 VPC 关联，然后在没有任何安全组的情况下重新创建 VPC 关联。删除 VPC 关联时要小心。此操作将阻止流量到达该服务网络中的服务。

使用网络 ACL 控制流向 VPC Lattice 的流量

网络访问控制列表 (ACL) 在子网级别允许或拒绝特定的入站或出站流量。默认网络 ACL 允许所有入站和出站流量。您可以为子网创建自定义网络 ACL，以提供额外的安全层。有关更多信息，请参阅 Amazon VPC 用户指南中的网络 ACL。

内容

您的客户端子网的网络 ACL

客户端子网的网络 ACL 必须允许客户端与 VPC Lattice 之间的流量。您可以从 VPC Lattice 的托管前缀列表中获取允许的 IP 地址范围。

以下是入站规则示例。

来源	协议	端口范围	Comment
vpc_lattice_cidr_block	TCP	1025-65535	允许从 VPC Lattice 到客户端的流量

以下是出站规则的示例。

目标位置	协议	端口范围	Comment
vpc_lattice_cidr_block	*listener*	*listener*	允许来自客户端的流量到莱迪思VPC

目标子网的网络 ACL

目标子网的网络 ACL 必须允许目标端口和运行状况检查端口上的目标与 VPC Lattice 之间的流量。您可以从 VPC Lattice 的托管前缀列表中获取允许的 IP 地址范围。

以下是入站规则示例。

来源	协议	端口范围	Comment
vpc_lattice_cidr_block	*target*	*target*	允许从 VPC 莱迪思到目标的流量
vpc_lattice_cidr_block	*health check*	*health check*	允许从 VPC 莱迪思到目标的运行状况检查流量

以下是出站规则的示例。

目标位置	协议	端口范围	Comment
vpc_lattice_cidr_block	*target*	1024-65535	允许从目标到 VPC Lattice 的流量
vpc_lattice_cidr_block	*health check*	1024-65535	允许从目标到 VPC Lattice 的运行状况检查流量

通过 Sigv4 身份验证的 Amazon VPC Lattice 请求

VPC Lattice 使用签名版本 4 (Sigv4) 或签名版本 4A (Sigv4a) 进行客户端身份验证。有关更多信息，请参阅《IAM 用户指南》中的适用于 API 请求的AWS 签名版本 4。

注意事项

- VPC Lattice 会尝试对任何使用 Sigv4 或 Sigv4A 签名的请求进行身份验证。未经验证，请求失败。
- VPC Lattice 不支持有效负载签名。必须发送 `x-amz-content-sha256` 标头，其值设置为 `"UNSIGNED-PAYLOAD"`。

示例

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

Python

此示例通过安全连接将已签名的请求发送到在网络中注册的服务。如果您希望使用请求，botocore 包可以简化验证过程，但并非严格要求。有关更多信息，请参阅 Boto3 文档中的凭证。

要安装botocore和awscrt软件包，请使用以下命令。有关更多信息，请参阅 AWS CRT Python。

```
pip install botocore awscrt
```

如果您在 Lambda 上运行客户端应用程序，请使用 Lambda 层安装所需的模块，或者将其包含在您的部署包中。

在以下示例中，使用您自己的值替换占位符值。

SIGv4

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

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

```
    prepped = request.prepare()

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

SIGv4A

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

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

    prepped = request.prepare()

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

Java

本示例展示了如何使用自定义拦截器执行请求签名。本示例使用来自 AWS SDK for Java 2.x 的默认凭证提供商类，为您获取正确的凭证。如果您希望使用特定的凭证提供商，可以从 AWS SDK for Java 2.x 中选择一个。仅 适用于 Java 的 AWS SDK 允许通过 HTTPS 进行未签名的有效负载。但您可以扩展签名程序，以支持通过 HTTP 的未签名有效负载。

SIGv4

```
package com.example;
```

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

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

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

public class sigv4 {

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

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

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

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

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

```
            values.forEach(value -> System.out.println("  " + key + ": " + value));
        });

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

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

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

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

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

    }
}
```

SIGv4A

此示例需要对的额外依赖software.amazon.awssdk:http-auth-aws-crt。

```
package com.example;


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

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

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

public class sigv4a {

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

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

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

        SignedRequest signedRequest = signer.sign(r -> r.identity(credentials)
```

```
                .request(httpRequest)
                .putProperty(AwsV4aHttpSigner.SERVICE_SIGNING_NAME, "vpc-lattice-
svcs")
                .putProperty(AwsV4aHttpSigner.PAYLOAD_SIGNING_ENABLED, false)
                .putProperty(AwsV4aHttpSigner.REGION_SET,
 RegionSet.create(String.join(" ",Arrays.copyOfRange(args, 1, args.length)))));

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

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

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

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

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

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

```
        } catch (IOException e) {
            System.err.println("[*] HTTP Request Failed.");
            e.printStackTrace();
        }
    }
}
```

Node.js

本例使用 aws-crt NodeJS 绑定来发送使用 HTTPS 的签名请求。

要安装 aws-crt 包，请使用以下命令。

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

如果存在 AWS_REGION 环境变量，本例将使用 AWS_REGION 指定的区域。默认区域为 us-east-1。

SIGv4

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

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

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

if (process.argv.length === 2) {
```

```
  console.error(process.argv[1] + ' <url>')
  process.exit(1)
}

const algorithm = crt.auth.AwsSigningAlgorithm.SigV4;

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

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

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

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

SIGv4A

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

function sigV4Sign(method, endpoint, service, algorithm) {
    const host = new URL(endpoint).host
```

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

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

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

const algorithm = crt.auth.AwsSigningAlgorithm.SigV4Asymmetric;

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

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

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

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

```
    })
    req.on('error', err => {
      console.log('Error: ' + err)
    })
    req.end()
  }
)
```

Golang

此示例使用适用于 Go 的 Smithy 代码生成器和 G o 编程语言的AWS SDK 来处理请求签名请求。该示例需要 Go 版本为 1.21 或更高版本。

SIGv4

```
package main

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

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

type nopCloser struct {
        io.ReadSeeker
}

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

```
type stringFlag struct {
        set   bool
        value string
}


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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

SIGv4A

```
package main

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

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

type nopCloser struct {
        io.ReadSeeker
}

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

type stringFlag struct {

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

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

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

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

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

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

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

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

        // Create a slice out of the provided regionset
        rs := strings.Split(regionSet.value, ",")

        // Perform the signing on req, using the credentials we retrieved from the
SDK
        err = signer.SignRequest(&sigv4a.SignRequestInput{
                Request:     req,
                Credentials: creds,
                Service:     "vpc-lattice-svcs",
```

```
                RegionSet: rs,
        })

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

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

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

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

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

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

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

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

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

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

Golang-GRPC

此示例使用适用于 Go 编程语言的AWS SDK 来处理 GRPC 请求的请求签名。这可以与 GRPC 示例代码存储库中的 echo 服务器一起使用。

```
package main

import (
```

```
    "context"
    "crypto/tls"
    "crypto/x509"

    "flag"
    "fmt"
    "log"
    "net/http"
    "net/url"
    "strings"
    "time"

    "google.golang.org/grpc"
    "google.golang.org/grpc/credentials"

    "github.com/aws/aws-sdk-go-v2/aws"
    v4 "github.com/aws/aws-sdk-go-v2/aws/signer/v4"
    "github.com/aws/aws-sdk-go-v2/config"

    ecpb "google.golang.org/grpc/examples/features/proto/echo"
)

const (
    headerContentSha    = "x-amz-content-sha256"
    headerSecurityToken = "x-amz-security-token"
    headerDate          = "x-amz-date"
    headerAuthorization = "authorization"
    unsignedPayload     = "UNSIGNED-PAYLOAD"
)

type SigV4GrpcSigner struct {
    service      string
    region       string
    credProvider aws.CredentialsProvider
    signer       *v4.Signer
}

func NewSigV4GrpcSigner(service string, region string, credProvider
 aws.CredentialsProvider) *SigV4GrpcSigner {
    signer := v4.NewSigner()
    return &SigV4GrpcSigner{
        service:      service,
        region:       region,
        credProvider: credProvider,
```

```
        signer:       signer,
    }
}

func (s *SigV4GrpcSigner) GetRequestMetadata(ctx context.Context, uri ...string)
 (map[string]string, error) {
    ri, _ := credentials.RequestInfoFromContext(ctx)
    creds, err := s.credProvider.Retrieve(ctx)
    if err != nil {
        return nil, fmt.Errorf("failed to load credentials: %w", err)
    }

    // The URI we get here is scheme://authority/service/ - for siging we want to
 include the RPC name
    // But RequestInfoFromContext only has the combined /service/rpc-name - so read the
 URI, and
    // replace the Path with what we get from RequestInfo.
    parsed, err := url.Parse(uri[0])
    if err != nil {
        return nil, err
    }
    parsed.Path = ri.Method

    // Build a request for the signer.
    bodyReader := strings.NewReader("")
    req, err := http.NewRequest("POST", uri[0], bodyReader)
    if err != nil {
        return nil, err
    }
    date := time.Now()
    req.Header.Set(headerContentSha, unsignedPayload)
    req.Header.Set(headerDate, date.String())
    if creds.SessionToken != "" {
        req.Header.Set(headerSecurityToken, creds.SessionToken)
    }
    // The signer wants this as //authority/path
    // So get this by triming off the scheme and the colon before the first slash.
    req.URL.Opaque = strings.TrimPrefix(parsed.String(), parsed.Scheme+":")

    err = s.signer.SignHTTP(context.Background(), creds, req, unsignedPayload,
 s.service, s.region, date)
    if err != nil {
        return nil, fmt.Errorf("failed to sign request: %w", err)
    }
```

```
    // Pull the relevant headers out of the signer, and return them to get
    // included in the request we make.
    reqHeaders := map[string]string{
        headerContentSha:    req.Header.Get(headerContentSha),
        headerDate:          req.Header.Get(headerDate),
        headerAuthorization: req.Header.Get(headerAuthorization),
    }
    if req.Header.Get(headerSecurityToken) != "" {
        reqHeaders[headerSecurityToken] = req.Header.Get(headerSecurityToken)
    }

    return reqHeaders, nil
}

func (c *SigV4GrpcSigner) RequireTransportSecurity() bool {
    return true
}

var addr = flag.String("addr", "some-lattice-service:443", "the address to connect to")
var region = flag.String("region", "us-west-2", "region")

func callUnaryEcho(client ecpb.EchoClient, message string) {
    ctx, cancel := context.WithTimeout(context.Background(), 10*time.Second)
    defer cancel()
    resp, err := client.UnaryEcho(ctx, &ecpb.EchoRequest{Message: message})
    if err != nil {
        log.Fatalf("client.UnaryEcho(_) = _, %v: ", err)
    }
    fmt.Println("UnaryEcho: ", resp.Message)
}

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

    pool, _ := x509.SystemCertPool()
    tlsConfig := &tls.Config{
        RootCAs: pool,
    }
```

```
    authority, _, _ := strings.Cut(*addr, ":") // Remove the port from the addr
    opts := []grpc.DialOption{
        grpc.WithTransportCredentials(credentials.NewTLS(tlsConfig)),

        // Lattice needs both the Authority to be set (without a port), and the SigV4
 signer
        grpc.WithAuthority(authority),
        grpc.WithPerRPCCredentials(NewSigV4GrpcSigner("vpc-lattice-svcs", *region,
 cfg.Credentials)),
    }

    conn, err := grpc.Dial(*addr, opts...)

    if err != nil {
        log.Fatalf("did not connect: %v", err)
    }
    defer conn.Close()
    rgc := ecpb.NewEchoClient(conn)

    callUnaryEcho(rgc, "hello world")
}
```

Amazon VPC Lattice 中的数据保护

责任共担模式 AWS 分担责任模型适用于 Amazon VPC Lattice 中的数据保护。如本模型所述 AWS ，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础架构上的内容的控制。此内容包括您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 AWS Shared Responsibility Model and GDPR 博客文章。

传输中加密

VPC Lattice 是一项完全托管的服务，由控制面板和数据面板组成。每个面板在服务中都有不同的用途。控制平面提供了用于创建、更新 read/describe、删除和列出 (CRUDL) 资源（例如`CreateService`和`UpdateService`）的管理 API。与 VPC 莱迪思控制平面的通信在传输途中受到 TLS 的保护。数据平面是 VPC Lattice Invoke API，它提供服务之间的互连。当您使用 HTTPS 或 TLS 时，TLS 会对与 VPC 莱迪思数据平面的通信进行加密。密码套件和协议版本使用 VPC Lattice 提供的默认值，并且不可配置。有关更多信息，请参阅 VPC Lattice 服务的 HTTPS 侦听器。

静态加密

默认情况下，静态数据加密有助于降低保护敏感数据的操作开销和复杂性。同时，还支持构建符合严格加密合规性和监管要求的安全应用程序。

内容

Server-side 使用 Amazon S3 托管密钥进行加密 (SSE-S3)

当您对 Amazon S3 托管密钥 (SSE-S3) 使用服务器端加密时，每个对象都使用唯一的密钥进行加密。作为额外的保护措施，我们使用定期轮换的根密钥对密钥本身进行加密。Amazon S3 服务器端加密使用市面上最强的分组密码之一，即 256 位高级加密标准 (AES-256) GCM 来加密您的数据。对于之前加密的对象 AES-GCM， AES-CBC 仍支持解密这些对象。有关更多信息，请参阅使用带有 Amazon 加密密钥的服务器端 S3-managed 加密 (SSE-S3)。

如果您为 VPC Lattice 访问日志的 S3 存储桶启用使用亚马逊 S3-managed 加密密钥 (SSE-S3) 的服务器端加密，我们会自动加密每个访问日志文件，然后再将其存储在您的 S3 存储桶中。有关更多信息，请参阅亚马逊 CloudWatch 用户指南中的发送到 Amazon S3 的日志。

Server-side 使用加密 AWS KMS 密钥存储在 AWS KMS (SSE-KMS)

Server-side 使用 AWS KMS 密钥加密 (SSE-KMS) 与此类似 SSE-S3，但使用此服务会带来额外的好处和费用。该 AWS KMS 密钥有单独的权限，可提供额外的保护，防止未经授权访问您的 Amazon S3 中的对象。 SSE-KMS 还为您提供审计跟踪，显示您的 AWS KMS 密钥何时被使用以及由谁使用。有关更多信息，请参阅通过 AWS Key Management Service (SSE-KMS) 使用服务器端加密。

内容

证书私有密钥的加密和解密

您的 ACM 证书和私钥使用别名的 AWS aws/acm托管 KMS 密钥进行加密。您可以在 AWS KMS 控制台的AWS 托管密钥下查看带有此别名的密钥 ID。

VPC Lattice 不会直接访问您的 ACM 资源。它使用 AWS TLS 连接管理器来保护和访问您的证书的私钥。当您使用 ACM 证书创建 VPC Lattice 服务时，VPC Lattice 会将您的证书与 AWS TLS Connection Manager 关联。这是通过在 AWS KMS 您的 AWS 托管密钥上创建带有前缀的授权来完成的aws/acm。授权是一种策略工具，允许 TLS Connection Manager 在加密操作中使用 KMS 密钥。授权允许被授权主体（TLS Connection Manager）对 KMS 密钥调用指定授权操作，以解密证书的私有密钥。然后，TLS连接管理器使用证书和解密（纯文本）私钥与VPC莱迪思服务的客户端建立安全连接（SSL/TLS 会话）。当证书与 VPC Lattice 服务取消关联时，该授权就会失效。

如果您想删除对 KMS 密钥的访问权限，我们建议您使用中的或`update-service`命令从服务中替换 AWS 管理控制台 或删除证书 AWS CLI。

VPC Lattice 的加密上下文

加密上下文是一组可选的键值对，其中包含有关您的私钥可能用于什么的上下文信息。 AWS KMS 将加密上下文绑定到加密数据，并将其用作其他经过身份验证的数据以支持经过身份验证的加密。

当您的 TLS 密钥与 VPC Lattice 和 TLS Connection manager 一起使用时，VPC Latrice 服务的名称将包含在用于静态加密密钥的加密上下文中。您可以通过查看 CloudTrail 日志中的加密上下文（如下一节所示）或查看 ACM 控制台中的 “关联资源” 选项卡，来验证您的证书和私钥用于哪个 VPC Lattice 服务。

要解密数据，在请求中包含相同的加密上下文。VPC Lattice 在所有 AWS KMS 加密操作中使用相同的加密环境，其中密钥为`aws:vpc-lattice:arn`，值为 VPC 莱迪思服务的亚马逊资源名称 (ARN)。

下面的示例显示操作输出中的加密上下文，例如 `CreateGrant`：

```
"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"
}
```

监控 VPC Lattice 的加密密钥

当您在VPC莱迪思服务中使用 AWS 托管密钥时，您可以使用AWS CloudTrail来跟踪VPC Lattice发送到的请求。 AWS KMS

CreateGrant

当您将 ACM 证书添加到 VPC Lattice 服务时，系统会代表您发送 `CreateGrant` 请求，使 TLS Connection Manager 能够解密与 ACM 证书关联的私有密钥

您可以在 “事件历史记录” 中CloudTrail将该**CreateGrant**操作作为事件查看CreateGrant。

以下是该CreateGrant操作的事件历史记录中的示例 CloudTrail 事件记录。


```
{
    "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"
}
```

在上面的CreateGrant示例中，被授权者主体是 TLS 连接管理器，加密上下文具有 VPC Lattice 服务 ARN。

ListGrants

您可以使用 KMS 密钥 ID 和账户 ID 来调用 ListGrants API。这将为您提供指定 KMS 密钥的所有授权列表。有关更多信息，请参阅 ListGrants。

在中使用以下ListGrants命令 AWS CLI 查看所有授权的详细信息。

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

下面是示例输出。

```
{
```

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

在上面的ListGrants示例中，被授权者主体是 TLS 连接管理器，加密环境具有 VPC 莱迪思服务 ARN。

Decrypt

VPC Lattice 使用 TLS Connection Manager 调用 Decrypt 操作来解密您的私有密钥，以便在您的 VPC Lattice 服务中提供 TLS 连接。您可以在事件历史记录 Decryp t 中将该**Decrypt**操作作为CloudTrail事件进行查看。

以下是该Decrypt操作的事件历史记录中的示例 CloudTrail 事件记录。

```
{
    "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"
}
```

适用于 Amazon VPC Lattice 的身份和访问管理

以下各节介绍如何使用 AWS Identity and Access Management (IAM) 通过控制谁可以执行 VPC Lattice API 操作来帮助保护您的 VPC 莱迪思资源。

主题

- Amazon VPC Lattice 如何与 IAM 配合使用
- 亚马逊 VPC 莱迪思 API 权限

- Identity-based 亚马逊 VPC 莱迪思的政策
- 使用适用于 Amazon VPC 莱迪思的服务相关角色
- AWS 适用于 Amazon VPC 莱迪思的托管策略

Amazon VPC Lattice 如何与 IAM 配合使用

在使用 IAM 管理对 VPC Lattice 的访问之前，了解哪些 IAM 功能可用于 VPC Lattice。

IAM 功能	VPC Lattice 支持
Identity-based 政策	是
Resource-based 政策	是
策略操作	是
策略资源	是
策略条件键	是
ACL	否
ABAC（策略中的标签）	是
临时凭证	是
服务角色	否
Service-linked 角色	是

要全面了解 VPC Lattice 和其他 AWS 服务如何与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中的与 IAM 配合使用的AWS 服务。

Identity-based VPC 莱迪思的策略

支持基于身份的策略：是

Identity-based 策略是您可以附加到身份（例如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 IAM JSON 策略元素引用。

Resource-based VPC 莱迪思内部的策略

支持基于资源的策略：是

Resource-based 策略是您附加到中的资源的 JSON 策略文档 AWS。在支持基于资源的策略的 AWS 服务中，服务管理员可以使用它们来控制对该 AWS 服务的特定资源的访问权限。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。

VPC Lattice 支持验证策略，这是一种基于资源的策略，允许您控制对服务网络中服务的访问。有关更多信息，请参阅 使用身份验证策略控制对VPC莱迪思服务的访问。

VPC Lattice 还支持基于资源的权限策略，以便与 AWS Resource Access Manager集成。您可以使用这些基于资源的策略来授予管理与其他 AWS 账户或组织的服务、资源配置和服务网络连接的权限。有关更多信息，请参阅 共享 VPC Lattice 实体。

VPC Lattice 的策略操作

支持策略操作：是

在 IAM 策略语句中，您可以从支持 IAM 的任何服务中指定任何 API 操作。对于 VPC Lattice，使用以下前缀为 API 操作命名：`vpc-lattice:`。例如：`vpc-lattice:CreateService`、`vpc-lattice:CreateTargetGroup` 和 `vpc-lattice:PutAuthPolicy`。

要在单个语句中指定多个操作，请使用逗号分隔，如下所示：

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

您也可以使用通配符指定多项操作。例如，您可以指定名称以单词 `Get` 开头的所有操作，如下所示：

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

有关 VPC Lattice API 操作的完整列表，请参阅《服务授权参考》中的 Amazon VPC Lattice 定义的操作。

VPC Lattice 的策略资源

支持策略资源：是

在 IAM policy 声明中，Resource 元素指定了该声明涵盖的一个或多个对象。对于 VPC Lattice，每个 IAM policy 语句都适用于您使用 ARN 指定的资源。

具体的 Amazon 资源名称（ARN）格式取决于资源。当您提供 ARN 时，请用您的资源特定信息替换*italicized*文本。

- 访问日志订阅：

```
"Resource": "arn:aws:vpc-lattice:region:account-id:accesslogsubscription/access-log-subscription-id"
```

- 侦听器：

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

- 资源网关

```
"Resource": "arn:aws:vpc-lattice:region:account-id:resourcegateway/resource-gateway-id"
```

- 资源配置

```
"Resource": "arn:aws:vpc-lattice:region:account-id:resourceconfiguration/resource-configuration-id"
```

- 规则：

```
"Resource": "arn:aws:vpc-lattice:region:account-id:service/service-id/
listener/listener-id/rule/rule-id"
```

- 服务：

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

- 服务网络：

```
"Resource": "arn:aws:vpc-lattice:region:account-id:servicenetwork/service-network-id"
```

- 服务网络服务关联：

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkserviceassociation/service-network-service-association-id"
```

- 服务网络资源配置关联

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkresourceassociation/service-network-resource-association-id"
```

- 服务网络 VPC 关联：

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkvpcassociation/service-network-vpc-association-id"
```

- 目标组：

```
"Resource": "arn:aws:vpc-lattice:region:account-id:targetgroup/target-group-id"
```

VPC Lattice 的策略条件键

支持特定于服务的策略条件键：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

`Condition` 元素根据定义的条件指定语句何时执行。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

要查看 VPC Lattice 条件密钥列表，请参阅《服务授权参考》中的 Amazon VPC Lattice 条件密钥。

AWS 支持全局条件密钥和特定于服务的条件密钥。有关 AWS 全局条件键的信息，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

VPC Lattice 中的访问控制列表（ACL）

支持 ACL：否

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，但它们不使用 JSON 策略文档格式。

Attribute-based 使用 VPC 莱迪思实现访问控制 (ABAC)

支持 ABAC（策略中的标签）：是

Attribute-based 访问控制 (ABAC) 是一种授权策略，它根据称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name` `aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的使用 ABAC 授权定义权限。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的使用基于属性的访问权限控制（ABAC）。

在 VPC Lattice 中使用临时凭证

支持临时凭证：是

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的临时安全凭证和使用 IAM 的。AWS 服务

VPC Lattice 的服务角色

支持服务角色：否

服务角色是由一项服务担任、代表您执行操作的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。

> **Warning**
>
> 更改服务角色的权限可能会破坏 VPC Lattice 的功能。只有在 VPC Lattice 提供指导的情况下，才能编辑服务角色。

Service-linked VPC 莱迪思的角色

支持服务关联角色：是

服务相关角色是一种与服务相关联的 AWS 服务服务角色。该服务可以代替您执行操作。 Service-linked 角色出现在您的， AWS 账户 并且归服务所有。IAM 管理员可以查看但不能编辑服务关联角色的权限。

有关创建或管理 VPC Lattice 服务相关角色的信息，请参阅 使用适用于 Amazon VPC 莱迪思的服务相关角色。

亚马逊 VPC 莱迪思 API 权限

您必须授予 IAM 身份（如用户或角色）调用所需 VPC Lattice API 操作的权限，如 VPC Lattice 的策略操作 中所述。此外，对于某些 VPC Lattice 操作，您必须授予 IAM 身份从其他 AWS API 调用特定操作的权限。

API 所需的权限

从 API 调用以下操作时，必须授予 IAM 用户调用指定操作的权限。

`CreateResourceConfiguration`

- `vpc-lattice:CreateResourceConfiguration`
- `ec2:DescribeSubnets`
- `rds:DescribeDBInstances`
- `rds:DescribeDBClusters`

`CreateResourceGateway`

- `vpc-lattice:CreateResourceGateway`
- `ec2:AssignPrivateIpAddresses`
- `ec2:AssignIpv6Addresses`
- `ec2:CreateNetworkInterface`
- `ec2:CreateNetworkInterfacePermission`
- `ec2:DeleteNetworkInterface`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`

`DeleteResourceGateway`

- `vpc-lattice:DeleteResourceGateway`
- `ec2:DeleteNetworkInterface`

UpdateResourceGateway

- vpc-lattice:UpdateResourceGateway
- ec2:AssignPrivateIpAddresses
- ec2:AssignIpv6Addresses
- ec2:UnassignPrivateIpAddresses
- ec2:CreateNetworkInterface
- ec2:CreateNetworkInterfacePermission
- ec2:DeleteNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:ModifyNetworkInterfaceAttribute

CreateServiceNetworkResourceAssociation

- vpc-lattice:CreateServiceNetworkResourceAssociation
- ec2:AssignIpv6Addresses
- ec2:CreateNetworkInterface
- ec2:CreateNetworkInterfacePermission
- ec2:DescribeNetworkInterfaces

CreateServiceNetworkVpcAssociation

- vpc-lattice:CreateServiceNetworkVpcAssociation
- ec2:DescribeVpcs
- ec2:DescribeSecurityGroups（仅在提供安全组时才需要）

UpdateServiceNetworkVpcAssociation

- vpc-lattice:UpdateServiceNetworkVpcAssociation
- ec2:DescribeSecurityGroups（仅在提供安全组时才需要）

CreateTargetGroup

- vpc-lattice:CreateTargetGroup
- ec2:DescribeVpcs

RegisterTargets

- vpc-lattice:RegisterTargets

- ec2:DescribeInstances（仅当目标组类型为 INSTANCE 时才需要）
- ec2:DescribeVpcs（仅当目标组类型为 INSTANCE 或 IP 时才需要）
- ec2:DescribeSubnets（仅当目标组类型为 INSTANCE 或 IP 时才需要）
- lambda:GetFunction（仅当目标组类型为 LAMBDA 时才需要）
- lambda:AddPermission（仅当目标组还没有调用指定 Lambda 函数的权限时才需要）

DeregisterTargets

- vpc-lattice:DeregisterTargets

CreateAccessLogSubscription

- vpc-lattice:CreateAccessLogSubscription
- logs:GetLogDelivery
- logs:CreateLogDelivery

DeleteAccessLogSubscription

- vpc-lattice:DeleteAccessLogSubscription
- logs:DeleteLogDelivery

UpdateAccessLogSubscription

- vpc-lattice:UpdateAccessLogSubscription
- logs:UpdateLogDelivery

Identity-based 亚马逊 VPC 莱迪思的政策

默认情况下，用户和角色没有创建或修改 VPC Lattice 资源的权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略（控制台）。

有关 VPC Lattice 定义的操作和资源类型的详细信息，包括每种资源类型的 ARN 格式，请参阅《服务授权参考》中的 Amazon VPC Lattice 的操作、资源和条件键。

内容

策略最佳实践

Identity-based 策略决定了是否有人可以在您的账户中创建、访问或删除 VPC Lattice 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的AWS 托管策略或工作职能的AWS 托管策略。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性：IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的使用 IAM Access Analyzer 验证策略。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的使用 MFA 保护 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。

完全访问所需的额外权限

要使用与VPC Lattice集成的其他 AWS 服务以及整套VPC Lattice功能，您必须拥有特定的额外权限。这些权限不包括在 `VPCLatticeFullAccess` 托管策略中，因为存在混淆代理权限升级风险。

您必须将以下策略附加到您的角色，并与 `VPCLatticeFullAccess` 托管策略一起使用。

JSON

```
{
    "Version":"2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "firehose:TagDeliveryStream",
                "lambda:AddPermission",
                "s3:PutBucketPolicy"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutResourcePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:CalledVia": [
                        "vpc-lattice.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/vpc-
lattice.amazonaws.com/AWSServiceRoleForVpcLattice"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
delivery.logs.amazonaws.com/AWSServiceRoleForLogDelivery*"
        }
    ]
```

```
}
```

本策略提供以下额外权限：

- `iam:AttachRolePolicy`：允许您将指定的托管策略附加到指定的 IAM 角色。
- `iam:PutRolePolicy`：允许您添加或更新嵌入在指定的 IAM 角色中的内联策略文档。
- `s3:PutBucketPolicy`：允许您将存储桶策略应用于 Amazon S3 存储桶。
- `firehose:TagDeliveryStream`：允许您为 Firehose 传输流添加或更新标签。

Identity-based VPC 莱迪思的策略示例

主题

策略示例：管理与服务网络的 VPC 关联

以下示例演示了一个策略，该策略授予使用此策略的用户创建、更新和删除服务网络的 VPC 关联的权限，但仅限于条件中指定的 VPC 和服务网络。有关指定条件密钥的更多信息，请参阅 VPC Lattice 的策略条件键。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "vpc-lattice:CreateServiceNetworkVpcAssociation",
            "vpc-lattice:UpdateServiceNetworkVpcAssociation",
            "vpc-lattice:DeleteServiceNetworkVpcAssociation"
         ],
         "Resource": [
```

```
            "*"
         ],
         "Condition": {
            "StringEquals": {
               "vpc-lattice:ServiceNetworkArn": "arn:aws:vpc-lattice:us-
west-2:123456789012:servicenetwork/sn-903004f88example",
               "vpc-lattice:VpcId": "vpc-1a2b3c4d"
            }
         }
      }
   ]
}
```

策略示例：创建与服务网络的服务关联

如果您不使用条件键来控制对 VPC Lattice 资源的访问，则可以在 Resource 元素中指定资源 ARN 来控制访问。

以下示例演示了一个策略，该策略将服务关联限制为，使用此策略的用户可通过指定用于 CreateServiceNetworkServiceAssociation API 操作的服务和服务网络的 ARN，来创建服务网络。有关指定 ARN 值的详细信息，请参阅 VPC Lattice 的策略资源。

JSON

```
{
   "Version":"2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "vpc-lattice:CreateServiceNetworkServiceAssociation"
         ],
         "Resource": [
            "arn:aws:vpc-lattice:us-
west-2:123456789012:servicenetworkserviceassociation/*",
            "arn:aws:vpc-lattice:us-west-2:123456789012:service/
svc-04d5cc9b88example",
            "arn:aws:vpc-lattice:us-west-2:123456789012:servicenetwork/
sn-903004f88example"
         ]
      }
```

```
   ]
}
```

策略示例：为资源添加标签

以下示例演示了一个策略，该策略授予使用此策略的用户在 VPC Lattice 资源上创建标签的权限。

JSON

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

策略示例：创建服务相关角色

当您的任何用户首次创建 VPC Lattice 资源时， AWS 账户 VPC Lattice 都需要权限才能创建服务相关角色。如果服务相关角色尚不存在，VPC Lattice 会在您的账户中创建此角色。服务相关角色向 VPC Lattice 授予权限，使其可以 AWS 服务 代表您呼叫其他人。有关更多信息，请参阅 the section called “使用服务关联角色”。

为使自动角色创建操作成功，用户必须具有 `iam:CreateServiceLinkedRole` 操作的权限。

```
"Action": "iam:CreateServiceLinkedRole"
```

以下示例演示了一个策略，该策略授予使用此策略的用户为 VPC Lattice 创建服务相关角色的权限。

JSON

```
{
   "Version":"2012-10-17",
```

```
   "Statement": [
      {
         "Effect": "Allow",
         "Action": "iam:CreateServiceLinkedRole",
         "Resource": "arn:aws:iam::*:role/aws-service-role/vpc-
lattice.amazonaws.com/AWSServiceRoleForVpcLattice",
         "Condition": {
            "StringLike": {
               "iam:AWSServiceName":"vpc-lattice.amazonaws.com"
            }
         }
      }
   ]
}
```

有关更多信息，请参阅 IAM 用户指南中的Service-linked 角色权限。

使用适用于 Amazon VPC 莱迪思的服务相关角色

Amazon VPC Lattice 使用服务相关角色来获得代表您呼叫他人所需的权限。 AWS 服务 有关更多信息，请参阅 IAM 用户指南中的Service-linked角色。

VPC Lattice 使用名为的服务相关角色。AWSServiceRoleForVpcLattice

Service-linked VPC 莱迪思的角色权限

AWSServiceRoleForVpcLattice 服务相关角色仅信任以下服务来担任该角色：

- `vpc-lattice.amazonaws.com`

名为的角色权限策略AWSVpcLatticeServiceRolePolicy允许 VPC Lattice 在`AWS/VpcLattice`命名空间中发布 CloudWatch 指标。有关更多信息，请参阅AWSVpcLatticeServiceRolePolicy《AWS 托管策略参考》。

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务关联角色。有关更多信息，请参阅 the section called “策略示例：创建服务相关角色”。

为VPC Lattice创建服务相关角色

您无需手动创建服务关联角色。当您在 AWS 管理控制台、或 AWS API 中创建 VPC 莱迪思资源时 AWS CLI，VPC Lattice 会为您创建服务相关角色。

如果您删除该服务关联角色，然后需要再次创建，您可以使用相同流程在账户中重新创建此角色。当您创建 VPC Lattice 资源时，VPC Lattice 会再次为您创建服务相关角色。

编辑 VPC 莱迪思的服务相关角色

您可以使用 IAM 编辑 AWSServiceRoleForVpcLattice 的描述。有关更多信息，请参阅《IAM 用户指南》中的编辑服务相关角色描述。

删除 VPC 莱迪思的服务相关角色

如果您不再需要使用 Amazon VPC Lattice，我们建议您将其删除AWSServiceRoleForVpcLattice。

只有在删除 AWS 账户中的所有 VPC Lattice 资源后，才能删除此服务相关角色。

使用 IAM 控制台 AWS CLI、或 AWS API 删除AWSServiceRoleForVpcLattice服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的删除服务相关角色。

删除服务相关角色后，当您在 AWS 账户中创建 VPC Lattice 资源时，VPC Lattice 会再次创建该角色。

VPC Lattice 服务相关角色支持的区域

VPC Lattice 支持在服务可用的所有区域中使用服务相关角色。

AWS 适用于 Amazon VPC 莱迪思的托管策略

AWS 托管策略是由创建和管理的独立策略 AWS。 AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住， AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于使用案例的客户管理型策略来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。 AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务 的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管式策略。

AWS 托管策略： VPCLatticeFullAccess

本策略提供对 Amazon VPC Lattice 的完全访问权限，以及对其他相关服务的有限访问权限。其中包含执行以下操作的权限：

- ACM — 检索自定义 SSL/TLS 域名的证书 ARN。
- CloudWatch — 查看访问日志和监控数据。
- CloudWatch 日志-设置访问日志并将其发送到 CloudWatch 日志。
- Amazon EC2 — 配置网络接口并检索有关 EC2 实例和 VPC 的信息。这用于创建资源配置、资源网关和目标组、配置 VPC Lattice 实体关联以及注册目标。
- 弹性负载均衡：检索有关应用程序负载均衡器的信息，将其注册为目标。
- Firehose — 检索有关用于存储访问日志的传输流的信息。
- Lambda：检索有关 Lambda 函数的信息，将其注册为目标。
- Amazon RDS — 检索有关 RDS 集群和实例的信息。
- Amazon S3：检索有关用于存储访问日志的 S3 存储桶的信息。

要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 VPCLatticeFullAccess。

要使用与VPC Lattice集成的其他 AWS 服务以及整套VPC Lattice功能，您必须拥有特定的额外权限。这些权限不包括在 `VPCLatticeFullAccess` 托管策略中，因为存在混淆代理权限升级风险。有关更多信息，请参阅 完全访问所需的额外权限。

AWS 托管策略： VPCLatticeReadOnlyAccess

本策略提供对 Amazon VPC Lattice 的只读访问权限，以及对其他相关服务的有限访问权限。其中包含执行以下操作的权限：

- ACM — 检索自定义 SSL/TLS 域名的证书 ARN。
- CloudWatch — 查看访问日志和监控数据。
- CloudWatch 日志-查看访问日志订阅的日志传送信息。
- Amazon EC2：检索有关 EC2 实例和 VPC 的信息，以创建目标组和注册目标。
- 弹性负载均衡：检索有关应用程序负载均衡器的信息。
- Firehose — 检索有关传输流的信息，以进行访问日志传输。
- Lambda：查看有关 Lambda 函数的信息。
- Amazon RDS — 检索有关 RDS 集群和实例的信息。
- Amazon S3：检索有关 S3 存储桶的信息，以进行访问日志传输。

要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 VPCLatticeReadOnlyAccess。

AWS 托管策略： VPCLatticeServicesInvokeAccess

本策略提供调用 Amazon VPC Lattice 服务的访问权限。

要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 VPCLatticeServicesInvokeAccess。

AWS 托管策略： AWSVpcLatticeServiceRolePolicy

此策略附加到一个名为的服务相关角色，该角色AWSServiceRoleForVpcLattice允许VPC Lattice代表您执行操作。您不能将此策略附加到您的 IAM 实体。有关更多信息，请参阅 使用适用于 Amazon VPC 莱迪思的服务相关角色。

要查看此策略的权限，请参阅《AWS 托管式策略参考》中的 AWSVpcLatticeServiceRolePolicy。

VPC Lattice 更新至 AWS 托管策略

查看有关VPC Lattice AWS 托管策略自该服务开始跟踪这些变更以来更新的详细信息。要获得有关本页面更改的自动提醒，请订阅《VPC Lattice 用户指南》的 RSS 源。

更改	描述	日期
VPCLatticeFullAccess	VPC Lattice 增加了只读权限来描述 Amazon RDS 集群和实例。	2024 年 12 月 1 日
VPCLatticeReadOnlyAccess	VPC Lattice 增加了只读权限来描述 Amazon RDS 集群和实例。	2024 年 12 月 1 日
AWSVpcLatticeServiceRolePolicy	VPC Lattice 增加了权限，允许 VPC Lattice 创建请求者管理的网络接口。	2024 年 12 月 1 日
VPCLatticeFullAccess	VPC Lattice 添加了一个新策略，授予对 Amazon VPC Lattice 的完全访问权限，以及对其他依赖服务的有限访问权限。	2023 年 3 月 31 日
VPCLatticeReadOnlyAccess	VPC Lattice 添加了一个新策略，授予对 Amazon VPC Lattice 的只读访问权限，以及对其他依赖服务的有限访问权限。	2023 年 3 月 31 日

更改	描述	日期
VPCLatticeServicesInvokeAccess	VPC Lattice 添加了一个新策略，授予调用 Amazon VPC Lattice 服务的访问权限。	2023 年 3 月 31 日
AWSVpcLatticeServiceRolePolicy	VPC Lattice 为其服务相关角色添加权限，允许 VPC Lattice 在命名空间中发布 CloudWatch 指标。`AWS/VpcLattice` 该`AWSVpcLatticeServiceRolePolicy` 策略包括调用 CloudWatch PutMetricDataAPI 操作的权限。有关更多信息，请参阅 使用适用于 Amazon VPC 莱迪思的服务相关角色。	2022 年 12 月 5 日
VPC Lattice 开始跟踪更改	VPC Lattice 开始跟踪其 AWS 托管策略的变更。	2022 年 12 月 5 日

Amazon VPC Lattice 的合规性验证

Third-party 作为多项合规计划的一部分，审计师对 Amazon VPC Lattice 的安全与 AWS 合规性进行评估。

要了解是否属于特定合规计划的范围，请参阅AWS 服务 “按合规计划划分的范围” ”，然后选择您感兴趣的合规计划。 AWS 服务 有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。有关您在使用时的合规责任的更多信息 AWS 服务，请参阅AWS 安全文档。

使用接口终端节点访问 Amazon VPC Lattice (AWS PrivateLink)

您可以创建接口 VPC 端点，在您的 VPC 和 Amazon VPC Lattice 之间建立私有连接。接口终端节点由一项技术提供支持 AWS PrivateLink，该技术使您无需互联网网关、NAT设备、VPN连接或 Direct

Connect 连接即可私密访问VPC Lattice API。VPC 中的实例不需要公有 IP 地址即可与 VPC Lattice API 进行通信。

每个接口端点由子网中的一个或多个网络接口表示。

接口 VPC 端点的注意事项

在为 VPC Lattice 设置接口 VPC 终端节点之前，请务必查看AWS PrivateLink 指南 AWS PrivateLink中的访问权限 AWS 服务。

VPC Lattice 支持从 VPC 调用其所有 API 操作。

为 VPC Lattice 创建接口 VPC 端点

您可以使用 Amazon VPC 控制台或 AWS Command Line Interface (AWS CLI) 为 VPC 莱迪思服务创建 VPC 终端节点。有关更多信息，请参阅AWS PrivateLink 指南中的创建接口 VPC 终端节点。

使用以下服务名称为 VPC Lattice 创建 VPC 端点：

`com.amazonaws.region.vpc-lattice`

如果为端点启用私有 DNS，您可以使用该区域的默认 DNS 名称（例如 `vpc-lattice.us-east-1.amazonaws.com`）向 VPC Lattice 发出 API 请求。

Amazon VPC Lattice 的弹性

AWS 全球基础设施是围绕 AWS 区域 可用区构建的。

AWS 区域 提供多个物理隔离和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络连接。

利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错能力和可扩展性。

有关 AWS 区域 和可用区的更多信息，请参阅AWS 全球基础设施。

Amazon VPC 莱迪思的基础设施安全

作为一项托管服务，Amazon VPC Lattice 受到 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅AWS 云安全。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的基础设施保护。

您可以使用 AWS 已发布的 API 调用通过网络访问 VPC Lattice。客户端必须支持以下内容：

- 传输层安全性协议（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 (PFS) 的密码套件，例如 DHE（短暂的）或 ECDHE（椭圆曲线短暂的 Diffie-Hellman）。 Diffie-Hellman大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

监控 Amazon VPC Lattice

使用本节中的功能监控您的 Amazon VPC Lattice 服务网络、服务、目标组和 VPC 连接。

内容

CloudWatch 亚马逊 VPC Lattice 的指标

Amazon VPC Lattice 将与您的目标群体和服务相关的数据发送到亚马逊 CloudWatch，并将其处理为可读的近乎实时的指标。这些指标会保留 15 个月，使您能够访问历史信息，并更好地了解您的 Web 应用程序或服务的执行情况。此外，可以设置用于监测特定阈值的警报，并在达到相应阈值时发送通知或执行操作。有关更多信息，请参阅 Amazon CloudWatch 用户指南。

Amazon VPC Lattice 使用您 AWS 账户中的服务相关角色向亚马逊发送指标。 CloudWatch有关更多信息，请参阅 使用适用于 Amazon VPC 莱迪思的服务相关角色。

内容

查看 Amazon CloudWatch 指标

您可以使用 CloudWatch 控制台或查看目标群体和服务的Amazon CloudWatch 指标 AWS CLI。

使用 CloudWatch 控制台查看指标

1. 打开 Amazon CloudWatch 控制台，网址为https://console.aws.amazon.com/cloudwatch/。
2. 在导航窗格中，选择指标。
3. 选择 `AWS/VpcLattice` 命名空间。
4. （可选）要跨所有维度查看某个指标，请在搜索字段中输入其名称。
5. (可选) 要按维度筛选，请选择下列选项之一：

- 要仅显示为目标组报告的指标，请选择目标组。要查看单个目标组的指标，请在搜索字段中输入其名称。
- 要仅显示为服务报告的指标，请选择服务。要查看单个服务的指标，请在搜索字段中输入其名称。

要查看指标，请使用 AWS CLI

使用以下 CloudWatch list-Metrics AWS CLI 命令列出可用指标：

```
aws cloudwatch list-metrics --namespace AWS/VpcLattice
```

有关每个指标及其维度的信息，请参阅 目标组指标 和 服务指标。

目标组指标

VPC Lattice 会自动将与目标群体相关的指标存储在 AWS/VpcLattice Amazon CloudWatch 命名空间中。有关目标组的更多信息，请参阅 VPC Lattice 中的目标组。

Dimensions

要筛选目标群体的指标，请使用以下维度：

- AvailabilityZone
- TargetGroup

指标	说明	TargetGroup 协议
TotalConnectionCount	连接总数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。	HTTP, HTTPS, TCP

指标	说明	TargetGroup 协议
	统计数据 • 最有用的统计数据是 Sum。	
ActiveConnectionCount	活动连接数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS, TCP
ConnectionErrorCount	连接失败总数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS, TCP

指标	说明	TargetGroup 协议
HTTP1_ConnectionCount	HTTP/1.1 连接总数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS
HTTP2_ConnectionCount	HTTP/2 连接总数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS

指标	说明	TargetGroup 协议
ConnectionTimeoutCount	连接的连接超时总数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS, TCP
TotalReceivedConnectionBytes	接收的连接字节总数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS, TCP

指标	说明	TargetGroup 协议
TotalSentConnectionBytes	发送的连接字节总数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS, TCP
TotalRequestCount	请求总数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS

指标	说明	TargetGroup 协议
ActiveRequestCount	活动请求总数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS
RequestTime	最后一个字节的请求时间（以毫秒为单位）。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Average 和 pNN.NN（百分比）。	HTTP, HTTPS

指标	说明	TargetGroup 协议
HTTPCode_2XX_Count, HTTPCode_3XX_Count, HTTPCode_4XX_Count, HTTPCode_5XX_Count	聚合 HTTP 响应代码。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS
TLSConnectionErrorCount	TLS 连接错误总数，不包括失败的证书验证。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS, TCP

指标	说明	TargetGroup 协议
TotalTLSConnectionHandshakeCount	成功的 TLS 连接握手总数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。	HTTP, HTTPS, TCP

服务指标

VPC Lattice 会自动将与服务相关的指标存储在 AWS/VpcLattice Amazon CloudWatch 命名空间中。有关服务的更多信息，请参阅 VPC Lattice 中的服务。

Dimensions

要筛选目标群体的指标，请使用以下维度：

- AvailabilityZone
- Service

指标	说明
RequestTimeoutCount	等待响应超时的请求总数。 报告标准 • 从资源收到流量之时起，始终报告（无论是零值还是非零值）。

指标	说明
	报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。
TotalRequestCount	请求总数。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。
RequestTime	请求时间（毫秒）。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Average 和 pNN.NN（百分比）。

指标	说明
HTTPCode_2XX_Count，HTTPCode_3XX_Count，HTTPCode_4XX_Count，HTTPCode_5XX_Count	聚合 HTTP 响应代码。 报告标准 • 总是从资源接收流量时开始报告（无论是零值还是非零值）。 报告频率 • 每分钟一次。 统计数据 • 最有用的统计数据是 Sum。

Amazon VPC Lattice 的访问日志

访问日志会捕获有关您的VPC Lattice服务和资源配置的详细信息。您可以使用这些访问日志来分析流量模式，并审计网络中的所有服务。对于 VPC Lattice 服务，我们发布VpcLatticeAccessLogs；对于资源配置，我们发布VpcLatticeResourceAccessLogs，需要单独配置。

访问日志是可选的，默认情况下处于禁用状态。启用访问日志后，您可以随时禁用。

定价

发布访问日志时需要付费。以您的名义在 AWS 本地发布的日志称为公开日志。有关销售日志定价的更多信息，请参阅 Amazon CloudWatch 定价，选择日志，然后在 Vended Logs 下查看定价。

内容

启用访问日志所需的 IAM 权限

要启用访问日志并将日志发送到其目标，必须在策略中将以下操作附加到您正在使用的 IAM 用户、组或角色。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Sid": "ManageVPCLatticeAccessLogSetup",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "vpc-lattice:CreateAccessLogSubscription",
                "vpc-lattice:GetAccessLogSubscription",
                "vpc-lattice:UpdateAccessLogSubscription",
                "vpc-lattice:DeleteAccessLogSubscription",
                "vpc-lattice:ListAccessLogSubscriptions"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

有关更多信息，请参阅《AWS Identity and Access Management 用户指南》中的添加和删除 IAM 标识权限。

更新附加到您正在使用的 IAM 用户、组或角色的策略后，请转到 启用访问日志。

访问日志目标

您可以将访问日志发送到以下目标。

Amazon CloudWatch 日志

- VPC Lattice 通常会在 2 分钟内将 CloudWatch 日志传送到日志。但请记住，实际日志传输时间是尽最大努力计算的，可能会有额外的延迟。
- 如果 CloudWatch 日志组没有特定权限，则会自动创建资源策略并将其添加到日志组中。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的发送到 CloudWatch 日志的日志。
- 您可以在 CloudWatch控制台的 “日志组 CloudWatch ” 下找到发送到的访问日志。有关更多信息，请参阅 Amazon CloudWatch 用户指南中的查看发送到 CloudWatch 日志的日志数据。

Amazon S3

- VPC Lattice 通常会在 6 分钟内将日志传输到 Amazon S3。但请记住，实际日志传输时间是尽最大努力计算的，可能会有额外的延迟。
- 如果存储桶没有特定权限，系统将自动创建存储桶策略，并将其添加到您的 Amazon S3 存储桶。有关更多信息，请参阅亚马逊 CloudWatch用户指南中的发送到 Amazon S3 的日志。
- 发送到 Amazon S3 的访问日志使用以下命名约定：

```
[bucket]/[prefix]/AWSLogs/[accountId]/VpcLattice/AccessLogs/[region]/[YYYY/
MM/DD]/[resource-id]/[accountId]_VpcLatticeAccessLogs_[region]_[resource-
id]_YYYYMMDDTHHmmZ_[hash].json.gz
```

- VpcLatticeResourceAccessLogs 发送到 Amazon S3 的内容使用以下命名约定：

```
[bucket]/[prefix]/AWSLogs/[accountId]/VpcLattice/ResourceAccessLogs/[region]/[YYYY/
MM/DD]/[resource-id]/[accountId]_VpcLatticeResourceAccessLogs_[region]_[resource-
id]_YYYYMMDDTHHmmZ_[hash].json.gz
```

Amazon Data Firehose

- VPC Lattice 通常会在 2 分钟内将日志传送到 Firehose。但请记住，实际日志传输时间是尽最大努力计算的，可能会有额外的延迟。
- 系统会自动创建一个服务相关角色，该角色授予 VPC Lattice 向 Amazon Data Firehose发送访问日志的权限。为使自动角色创建成功，用户必须具有 `iam:CreateServiceLinkedRole` 操作的权限。有关更多信息，请参阅 Amazon CloudWatch 用户指南 Amazon Data Firehose中的发送到的日志。

- 有关查看发送到 Amazon Data Firehose日志的更多信息，请参阅《Amazon Data Firehose 开发人员指南》中的监控 Amazon Kinesis Data Streams。

启用访问日志

完成以下过程，配置访问日志，以捕获访问日志并将其传输到您选择的目标。

内容

使用控制台启用访问日志

您可以在创建期间启用服务网络、服务或资源配置的访问日志。您还可以在创建服务网络、服务或资源配置后启用访问日志，如以下过程所述。

要使用控制台创建基本服务

1. 打开位于 https://console.aws.amazon.com/vpc/ 的 Amazon VPC 控制台。
2. 选择服务网络、服务或资源配置。
3. 选择操作和编辑日志设置。
4. 打开访问日志切换开关。
5. 为访问日志添加传输目标，如下所示：
   - 选择CloudWatch 日志组，然后选择一个日志组。要创建日志组，请选择在中创建日志组 CloudWatch。
   - 选择 S3 存储桶并输入 S3 存储桶路径，包括任何前缀。要搜索 S3 存储桶，请选择浏览 S3。
   - 选择 Kinesis Data Firehose 传输流，然后选择一个传输流。要创建传输流，请选择在 Kinesis 中创建传输流。
6. 选择保存更改。

使用启用访问日志 AWS CLI

使用 CLI 命令create-access-log-subscription启用服务网络或服务的访问日志。

请求追踪

VPC Lattice 支持跨客户端、目标和日志的请求跟踪和关联，以实现可观察性和使用标头进行调试。 x-amzn-requestid此标头可以由客户端设置和发送，也可以由VPC Lattice生成，发送到目标，也可以在访问日志中找到。

默认 行为

- VPC Lattice 会为每个请求自动生成此标头。
- 该值是随机生成的标识符（默认为 UUID 样式）。
- 生成的标识符是：
  - 已传播到下游目标。
  - 在客户端的响应标头中返回。
  - 登录访问日志

示例（默认响应）

以下是发送给客户端的响应示例，默认行为为 VPC Lattice 为 value eo x-amzn-requestid f 标头生成随机值。

```
{
    "HTTP/1.1 200 OK
    x-amzn-requestid: a9f2c7a1-6b4f-4c79-9e87-ff5a1234a001"
}
```

客户机设定值

- 客户端可以选择在传入的请求上设置此标头，以覆盖自动生成的值。
- 注意事项
  - 标头值不需要遵循 UUID 格式。
  - 如果标头值超过 512 字节，VPC Lattice 会将其截断为 512 字节。

- 成功重写后，提供的标头值将：
  - 出现在响应标题中
  - 传播到目标
  - 出现在访问日志和指标中

示例（覆盖客户端请求）

以下是客户端发送的带有标头值的请求的示例。

```
{
    "GET /my-service/endpoint HTTP/1.1
    Host: my-api.example.com
    x-amzn-requestid: trace-request-foobar"
}
```

示例（默认覆盖响应）

以下是向客户端发送的带有被覆盖值的响应的示例。

```
{
    "HTTP/1.1 200 OK
    x-amzn-requestid: trace-request-foobar"
}
```

访问日志内容

下表描述了访问日志条目的字段。

字段	说明	Format
`callerPrincipalTags`	请求 PrincipalTags 中的。	JSON
`hostHeader`	请求的授权标头。	字符串
`sslCipher`	用于建立客户端 TLS 连接的一组密码的 OpenSSL 名称。	字符串
`serviceNetworkArn`	服务网络 ARN。	arn: aws: vpc-lattice::: servicenetwork/ *region account id*
`resolvedUser`	启用并完成验证后用户的 ARN。	null | ARN | "Anonymous" | "Unknown"
`authDeniedReason`	启用验证后访问被拒绝的原因。	null | "Service" | "Network" | "Identity"

字段	说明	Format
requestMethod	请求的方法标头。	字符串
targetGroupArn	目标主机所属的目标主机组。	字符串
tlsVersion	TLS 版本。	TLSv*x*
userAgent	用户代理标头。	字符串
serverNameIndication	[仅限 HTTPS]在服务器名称指示（SNI）的 ssl 连接套接字上设置的值。	字符串
destinationVpcId	目标 VPC ID。	vpc-*xxxxxxxx*
sourceIpPort	源的 IP 地址和端口。	*ip*:*port*
targetIpPort	目标的 IP 地址和端口。	*ip*:*port*
serviceArn	服务 ARN。	arn: aws: vpc-lattice::: service/*region account id*
sourceVpcId	源 VPC ID。	vpc-*xxxxxxxx*
requestPath	请求的路径。	LatticePath?:*path*
startTime	请求开始时间。	*YYYY*-*MM*-*DD* T*HH*:*MM*: *SS* Z
protocol	协议。目前是 HTTP/1.1 或 HTTP/2。	字符串
responseCode	HTTP 响应代码。仅记录最终标头的响应代码。有关更多信息，请参阅 访问日志问题排查。	整数
bytesReceived	收到的正文和标头字节。	整数
bytesSent	发送的正文和标头字节。	整数

字段	说明	Format
duration	请求从开始时间到最后一个字节输出的总持续时间（毫秒）。	整数
requestToTargetDuration	请求从开始时间到发送到目标的最后一个字节的总持续时间（毫秒）。	整数
responseFromTargetDuration	请求从目标主机读取第一个字节到发送到客户端的最后一个字节的总持续时间（毫秒）。	整数
grpcResponseCode	gRPC 响应代码。有关更多信息，请参阅状态代码及其在 gRPC 中的使用。仅当服务支持 gRPC 时，才会记录此字段。	整数
requestId	这是一个唯一标识符，作为标x-amzn-requestid题的值自动包含在响应中。它支持跨客户端、目标和日志的请求关联，以实现可观察性和调试。	字符串
callerPrincipal	经过验证的主体。	字符串
callerX509SubjectCN	使用者名称（CN）。	字符串
callerX509IssuerOU	发布者（OU）。	字符串
callerX509SANNameCN	发布者备用名称（名称/CN）。	字符串
callerX509SANDNS	使用者备用名称（DNS）。	字符串
callerX509SANURI	使用者备用名称（URI）。	字符串

字段	说明	Format
sourceVpcArn	发出请求的 VPC 的 ARN。	arn: aws: ec2::: vpc/ *region account id*

字段	说明	Format
failureReason	指明请求失败的原因。可能的值包括： • TargetConnectionError -请求未能连接到目标组中的目标。 • TargetProtocolError -目标未使用有效数据进行响应。这可能表明目标具有无效的 TLS 记录或使用了无效的目标组协议。 • TargetDataTimeout -已达到空闲超时。 • TargetConnectionClosed -目标在完成响应之前关闭了连接。 • ClientConnectionClosed -客户端在收到完整响应之前关闭了连接。 • ClientRateLimited -客户端超出了连接限制，VPC Lattice 限制了连接速率。 • ClientAccessDenied -VPC Lattice 拒绝访问该资源。使用，authDeniedReason 详细了解 VPC Lattice 拒绝访问的原因。 • ClientProtocolError -客户发送的数据无法理解。这可能表明客户端使用了无效的 TLS 记录或无效的协议。	字符串

字段	说明	Format
	• ConnectionDuration Exceeded -连接已达到最大连接持续时间限制。 • InternalError -处理请求时出现内部错误。	

示例

以下是示例日志条目。

```
{
    "callerPrincipalTags" : "{ "TagA": "ValA", "TagB": "ValB", ... }",
    "hostHeader": "example.com",
    "sslCipher": "-",
    "serviceNetworkArn": "arn:aws:vpc-lattice:us-west-2:123456789012:servicenetwork/
svn-1a2b3c4d",
    "resolvedUser": "Unknown",
    "authDeniedReason": "null",
    "requestMethod": "GET",
    "targetGroupArn": "arn:aws:vpc-lattice:us-west-2:123456789012:targetgroup/
tg-1a2b3c4d",
    "tlsVersion": "-",
    "userAgent": "-",
    "serverNameIndication": "-",
    "destinationVpcId": "vpc-0abcdef1234567890",
    "sourceIpPort": "178.0.181.150:80",
    "targetIpPort": "131.31.44.176:80",
    "serviceArn": "arn:aws:vpc-lattice:us-west-2:123456789012:service/svc-1a2b3c4d",
    "sourceVpcId": "vpc-0abcdef1234567890",
    "requestPath": "/billing",
    "startTime": "2023-07-28T20:48:45Z",
    "protocol": "HTTP/1.1",
    "responseCode": 200,
    "bytesReceived": 42,
    "bytesSent": 42,
    "duration": 375,
    "requestToTargetDuration": 1,
    "responseFromTargetDuration": 1,
    "grpcResponseCode": 1,
```

```
    "requestId": "a9f2c7a1-6b4f-4c79-9e87-ff5a1234a001"
}
```

资源访问日志内容

下表描述了资源访问日志条目的字段。

字段	说明	Format
serviceNetworkArn	服务网络 ARN。	arn: *partition* vpc-latti ce::: servicenetwork/ *region account id*
serviceNetworkReso urceAssociationId	服务网络资源 ID。	*snra-xxx*
vpcEndpointId	用于访问资源的终端节点 ID。	字符串
sourceVpcArn	源 VPC ARN 或从中发起连接的 VPC。	字符串
resourceConfigurat ionArn	被访问的资源配置的 ARN。	字符串
protocol	用于与资源配置通信的协议。目前仅支持 tcp。	字符串
sourceIpPort	发起连接的源的 IP 地址和端口。	*ip*:*port*
destinationIpPort	发起连接时使用的 IP 地址和端口。这将是 SN-E/SN-A 的 IP。	*ip*:*port*
gatewayIpPort	资源网关用于访问资源的 IP 地址和端口。	*ip*:*port*
resourceIpPort	资源的 IP 地址和端口。	*ip*:*port*

示例

以下是示例日志条目。

```
{
    "eventTimestamp": "2024-12-02T10:10:10.123Z",
    "serviceNetworkArn": "arn:aws:vpc-lattice:us-west-2:1234567890:servicenetwork/
sn-1a2b3c4d",
    "serviceNetworkResourceAssociationId": "snra-1a2b3c4d",
    "vpcEndpointId": "vpce-01a2b3c4d",
    "sourceVpcArn": "arn:aws:ec2:us-west-2:1234567890:vpc/vpc-01a2b3c4d",
    "resourceConfigurationArn": "arn:aws:vpc-lattice:us-
west-2:0987654321:resourceconfiguration/rcfg-01a2b3c4d",
    "protocol": "tcp",
    "sourceIpPort": "172.31.23.56:44076",
    "destinationIpPort": "172.31.31.226:80",
    "gatewayIpPort": "10.0.28.57:49288",
    "resourceIpPort": "10.0.18.190:80"
}
```

访问日志问题排查

本章节包含您可能在访问日志中看到的 HTTP 错误代码的解释。

错误代码	可能的原因
HTTP 400：错误请求	• 客户端发送的请求格式错误，不符合 HTTP 规范。 • 整个请求标头超过 60K，或超过 100 个标头。 • 客户端在发送完整的请求正文之前关闭了连接。
HTTP 403：禁止访问	已为服务配置验证，但传入请求未经过验证或授权。
HTTP 404：服务不存在	您正在尝试连接到不存在或未注册到正确服务网络的服务。
HTTP 500：内部服务器错误	VPC Lattice 遇到错误，例如无法连接到目标。
HTTP 502：无效网关	VPC Lattice 遇到错误。

CloudTrail 亚马逊 VPC Lattice 的日志

Amazon VPC Lattice 与AWS CloudTrail一项服务集成，该服务提供用户、角色或角色所执行操作的 AWS 服务记录。 CloudTrail 将 VPC Lattice 的所有 API 调用捕获为事件。捕获的调用包括来自VPC莱迪思控制台的调用和对VPC Lattice API操作的代码调用。通过收集的信息 CloudTrail，您可以确定向 VPC Lattice 发出的请求、发出请求的 IP 地址、发出请求的时间以及其他详细信息。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根用户凭证还是用户凭证发出的。
- 请求是否代表 IAM Identity Center 用户发出。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

CloudTrail 在您创建账户 AWS 账户 时在您的账户中处于活动状态，并且您可以自动访问 CloudTrail 活动历史记录。 CloudTrail 事件历史记录提供了过去 90 天中记录的管理事件的可查看、可搜索、可下载且不可变的记录。 AWS 区域有关更多信息，请参阅《AWS CloudTrail 用户指南》中的 “使用 CloudTrail 事件历史记录”。查看活动历史记录不 CloudTrail收取任何费用。

要持续记录 AWS 账户 过去 90 天内的事件，请创建跟踪或 CloudTrailLake 事件数据存储。

CloudTrail 步道

跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。使用创建的所有跟踪 AWS 管理控制台 都是多区域的。您可以通过使用 AWS CLI创建单区域或多区域跟踪。建议创建多区域跟踪，因为您可以捕获账户 AWS 区域 中的所有活动。如果您创建单区域跟踪，则只能查看跟踪的 AWS 区域中记录的事件。有关跟踪的更多信息，请参阅《AWS CloudTrail 用户指南》中的为您的 AWS 账户创建跟踪和为组织创建跟踪。

通过创建跟踪，您可以免费将正在进行的管理事件的一份副本传送到您的 Amazon S3 存储桶，但会收取 Amazon S3 存储费用。 CloudTrail 有关 CloudTrail 定价的更多信息，请参阅AWS CloudTrail 定价。有关 Amazon S3 定价的信息，请参阅 Amazon S3 定价。

CloudTrail 湖泊事件数据存储

CloudTrail L@@ ak e 允许你对自己的事件运行基于 SQL 的查询。 CloudTrail Lake 将基于行的 JSON 格式的现有事件转换为 Apache ORC 格式。ORC 是一种针对快速检索数据进行优化的列式存储格式。事件将被聚合到事件数据存储中，它是基于您通过应用高级事件选择器选择的条件的不

可变的事件集合。应用于事件数据存储的选择器用于控制哪些事件持续存在并可供您查询。有关 CloudTrail Lake 的更多信息，请参阅AWS CloudTrail 用户指南中的使用 AWS CloudTrail Lake。

CloudTrail 湖泊事件数据存储和查询会产生费用。创建事件数据存储时，您可以选择要用于事件数据存储的定价选项。定价选项决定了摄取和存储事件的成本，以及事件数据存储的默认和最长保留期。有关 CloudTrail 定价的更多信息，请参阅AWS CloudTrail 定价。

要监控其他操作，请使用访问日志。有关更多信息，请参阅 访问日志。

VPC 莱迪思管理事件位于 CloudTrail

管理事件提供有关对中的资源执行的管理操作的信息 AWS 账户。这些也称为控制面板操作。默认情况下， CloudTrail 记录管理事件。

Amazon VPC Lattice 将 VPC 莱迪思控制平面操作记录为管理事件。有关 VPC Lattice 记录的 Amazon VPC Lattice 控制平面操作列表 CloudTrail，请参阅亚马逊 VPC Lattice API 参考。

VPC 莱迪思事件示例

事件代表来自任何来源的单个请求，包括有关所请求的 API 操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此事件不会按任何特定顺序出现。

以下示例显示了该CreateService操作 CloudTrail 的事件。


```
{
  "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"
}
```

以下示例显示了该DeleteService操作 CloudTrail 的事件。

```
{
  "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"
}
```


有关 CloudTrail 录音内容的信息，请参阅《AWS CloudTrail 用户指南》中的CloudTrail录制内容。

Amazon VPC Lattice 的配额

您的每个配额 AWS 账户 都有默认配额，以前称为限制 AWS 服务。除非另有说明，否则每个配额都是 Region-specific。您可以请求增加某些配额，但其他一些配额无法增加。

要查看 VPC Lattice 的配额，请打开服务限额控制台。在导航窗格中，选择 AWS 服务，然后选择 VPC Lattice。

要请求提高配额，请参阅《服务配额用户指南》中的请求提高配额。

您 AWS 账户 有以下与 VPC 莱迪思相关的配额。

Name	默认值	可调整	说明
验证策略大小	每个受支持的区域：10 KB	否	验证策略中 JSON 文件的最大大小。
每个组资源配置的子资源配置数	每个受支持的区域：60 个	是	组资源配置中子资源配置的最大数量。如需增加容量和增加限制，请联系 Supp AWS ort。
每个 AWS 区域的域名验证	每个受支持的区域：5 个	是	每个账户可以创建的域验证的最大数量。如需增加容量和增加限制，请联系 Supp AWS ort。
每个服务的侦听器数	每个受支持的区域：2 个	是	可以为服务创建的最大侦听器数。如需增加容量和增加限制，请联系 Supp AWS ort。
每个服务网络的资源配置数	每个受支持的区域：500 个	是	与服务网络关联的资源配置的最大数量。如需增加容量和增加限制，请联系 Supp AWS ort。

Name	默认值	可调整	说明
每个 AWS 区域的资源配置	每个支持的区域：2000 个	是	一个 AWS 账户在每个 AWS 区域可以拥有的最大资源配置数量。如需增加容量和增加限制，请联系 Supp AWS ort。
每个 VPC 的资源网关数	每个受支持的区域：500 个	是	VPC 中资源网关的最大数量。如需增加容量和增加限制，请联系 Supp AWS ort。
每个侦听器的规则数	每个受支持的区域：10 个	是	可以为服务侦听器定义的最大规则数。如需增加容量和增加限制，请联系 Supp AWS ort。
每个关联的安全组数	每个受支持的区域：5 个	否	可以添加到 VPC 与服务网络之间的关联中的最大安全组数。
每个服务网络的服务关联数	每个受支持的区域：500 个	是	可以与单个服务网络关联的最大服务数。如需增加容量和增加限制，请联系 Supp AWS ort。
每个区域的服务网络数	每个受支持的区域：50 个	是	每个区域的最大服务网络数。如需增加容量和增加限制，请联系 Supp AWS ort。
每个区域的服务数	每个支持的区域：2000 个	是	每个区域的最大服务数。如需增加容量和增加限制，请联系 Supp AWS ort。

Name	默认值	可调整	说明
每个区域的目标组	每个受支持的区域：500 个	是	每个区域的最大目标组数。如需增加容量和增加限制，请联系 Supp AWS ort。
每个服务的目标组数	每个受支持的区域：10 个	是	可以与服务关联的最大目标组数。如需增加容量和增加限制，请联系 Supp AWS ort。
每个目标组的目标	每个受支持的区域：1,000 个	是	可以与单个目标组关联的最大目标数。如需增加容量和增加限制，请联系 Supp AWS ort。
每个服务网络的 VPC 关联数	每个受支持的区域：500 个	是	可以与单个服务网络关联的最大 VPC 数。如需增加容量和增加限制，请联系 Supp AWS ort。
每个服务网络的服务网络类型 VPC 端点	每个受支持的区域：200 个	是	与服务网络关联的服务网络端点的最大数量。如需增加容量和增加限制，请联系 Supp AWS ort。

VPC Lattice 不支持以下可用区：`use1-az3`、`usw1-az2`、`apne1-az3`、、`apne2-az2`、`euw1-az4cac1-az3`、`ilc1-az2`。

以下限制也适用。

限制	值	说明
每个可用区每个服务的带宽	10 Gbps	每个可用区为每个服务分配的默认带宽。这可以增加，请联系您的解决方

限制	值	说明
		案架构师 (SA) 或技术客户经理 (TAM) 寻求进一步帮助。
每个连接的最大传输单元（MTU）	8500 字节	服务可以接受的最大数据包的大小。
每个可用区每项服务的每秒请求数	10000	对于 HTTP 服务，这是每个可用区每项服务每秒的默认请求数。这可以增加，请联系您的解决方案架构师 (SA) 或技术客户经理 (TAM) 寻求进一步帮助。
VPC 莱迪思服务的每个连接的连接空闲时间	1 minute	在没有活动请求（对于 HTTP 和 GRPC）或 VPC 莱迪思服务没有活动数据传输的情况下，连接可以处于空闲状态的默认时间。 TLS-PASSTHROUGH您可以使用 HTTP 和应用程序级 keepalive 将此空闲超时延长至最大连接生命周期持续时间。这可以增加，请联系您的解决方案架构师 (SA) 或技术客户经理 (TAM) 寻求进一步帮助。
VPC 莱迪思服务的每个连接的最长连接寿命	10 分钟	VPC 莱迪思服务在客户端和服务器之间可以打开连接的最长时间。
VPC 莱迪思资源的每个连接的最长连接寿命	NA	VPC Lattice 不对资源施加任何生命周期连接限制。客户端和服务器在确定生命周期连接持续时间的同时，还要知道 VPC Lattice 资源的空闲超时（即 350 秒）。
VPC 莱迪思资源的每个连接的连接空闲时间	350 秒	您可以使用 TCP keepalive 来延长此空闲超时时间。

限制	值	说明
每个 VPC 的服务网络	1 个服务网络	您只能通过关联将一个 VPC 连接到一个服务网络。要将一个 VPC 连接到多个服务网络，您可以使用服务网络类型的 VPC 终端节点。

亚马逊 VPC 莱迪思用户指南的文档历史记录

下表介绍了 VPC Lattice 的文档发布。

变更	说明	日期
为资源网关添加了可配置的 IP 地址	VPC Lattice 现在支持资源网关的可配置 IP 地址。	2025 年 10 月 7 日
为添加了 VPC 晶格 Oracle Database@AWS	VPC Lattice Oracle Database@AWS 已发布。	2025 年 6 月 26 日
增加了对管理端点的双栈支持	VPC Lattice 现在支持所有 VPC 莱迪思管理的双栈（IPv4 和 IPv6）端点。 APIs	2025 年 4 月 30 日
共享和访问资源	VPC Lattice 现在支持跨虚拟私有云和账户边界共享和访问资源。这包括对VPCLatticeReadOnlyAccess和VPCLatticeFullAccess政策的更新。	2024 年 12 月 1 日
TLS 直通	VPC Lattice 现在支持 TLS 直通，允许您在应用程序中执行 TLS 终止以进行身份验证。 end-to-end	2024 年 5 月 14 日
Lambda 事件结构版本	VPC Lattice 现在支持新版本的 Lambda 事件结构。	2023 年 9 月 7 日
Support 支持共享 VPCs	参与者可以在共享 VPC 中创建 VPC Lattice 目标组。	2023 年 7 月 5 日
通用版本	VPC Lattice 用户指南通用版本（GA）发布	2023 年 3 月 31 日

VPC Lattice 现已报告其 AWS 托管策略的变更	托管策略的更改在 “安全” 一章的 “VPC Lattice AWS 托管策略” 中报告。	2023 年 3 月 29 日
支持应用程序负载均衡器目标类型	VPC Lattice 现在支持创建应用程序负载均衡器类型的目标组。	2023 年 3 月 29 日
Support 支持所有实例类型	VPC Lattice 现在支持所有实例类型。	2023 年 3 月 27 日
IPv6 支持	VPC Lattice 现在同时支持 IP 目标组 IPv4 和 IPv6 IP 目标组。	2023 年 3 月 27 日
HTTP2 运行状况检查的协议版本	当目标组协议版本为时，现在支持运行状况检查 HTTP2。	2023 年 3 月 27 日
修复了侦听器规则的响应操作	除了转发操作外，VPC Lattice 服务的侦听器现在还支持固定响应操作。	2023 年 3 月 27 日
支持自定义域名	现在可以为 VPC Lattice 服务配置自定义域名	2023 年 2 月 14 日
支持 BYOC（自带证书）	VPC Lattice 支持在 ACM 中使用您自己的 SSL/TLS 证书作为自定义域名。	2023 年 2 月 14 日
VPC Lattice 提供不支持的实例类型的更新列表报告	不支持的实例列表中又增加了三个实例。	2023 年 1 月 26 日
VPC Lattice 现已报告其 AWS 托管策略的变更	从 2022 年 12 月 5 日开始，“安全性”章节的“VPC Lattice 的AWS 托管策略”中报告了对托管策略的更改。列出的第一个更改是增加了 CloudWatch 监控所需的权限。	2022 年 12 月 5 日

初始版本	VPC Lattice 用户指南首次发布	2022 年 12 月 5 日

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。