aws

使用者指南

Amazon VPC Lattice

Amazon VPC Lattice: 使用者指南

Table of Contents

什麼是 Amazon VPC Lattice？

Amazon VPC Lattice 是一項全受管應用程式聯網服務，可用來連接、保護和監控應用程式的 服務和資源。您可以將 VPC Lattice 與單一虛擬私有雲端 (VPC) 搭配使用，或從一或多個帳戶跨多個 VPCs 使用。

現代應用程式可以包含多個小型和模組化元件，通常稱為微服務，例如 HTTP API、資料庫等資源，以及由 DNS 和 IP 地址端點組成的自訂資源。雖然現代化具有優勢，但當您連接這些微服務和資源時，它也會帶來聯網複雜性和挑戰。例如，如果開發人員分散在不同團隊中，他們可能會在多個帳戶或 VPCs 之間建置和部署微服務和資源。

在 VPC Lattice 中，我們將微服務稱為服務，並僅代表資源做為資源組態。這些是您在 VPC Lattice 使用者指南中看到和將使用的術語。

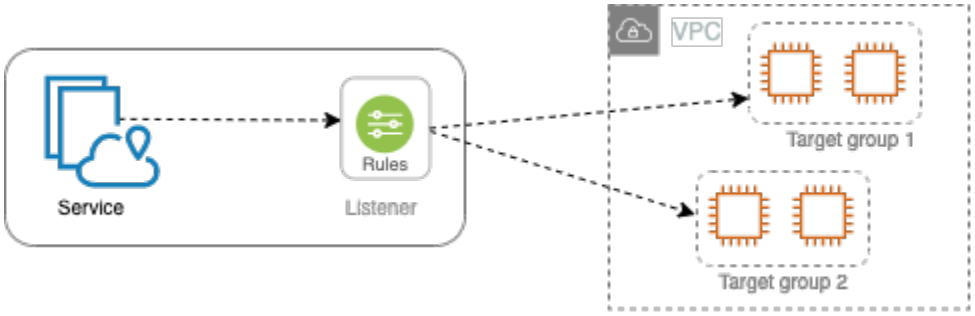
目錄

關鍵元件

若要使用 Amazon VPC Lattice，您應該熟悉其主要元件。

服務

> 可獨立部署的軟體單位，可交付特定任務或函數。服務可以在帳戶或虛擬私有雲端 (VPC) 內的 EC2 執行個體或 ECS/EKS/Fargate 容器，或以 Lambda 函數的形式執行。VPC Lattice 服務具有下列元件：目標群組、接聽程式和規則。



目標群組

執行應用程式或服務的資源集合，也稱為目標。這些類似於 Elastic Load Balancing 提供的目標群組，但不可互換。支援的目標類型包括 EC2 執行個體、IP 地址、Lambda 函數、Application Load Balancer、Amazon ECS 任務和 Kubernetes Pod。

接聽程式

檢查連線請求並將其路由到目標群組中目標的程序。您可以使用通訊協定和連接埠號碼來設定接聽程式。

規則

接聽程式的預設元件，可將請求轉送至 VPC Lattice 目標群組中的目標。每個規則由優先順序、一或多個動作及一或多個條件組成。規則決定接聽程式如何路由用戶端請求。

資源

資源是實體，例如 Amazon Relational Database Service (Amazon RDS) 資料庫、Amazon EC2 執行個體、應用程式端點、網域名稱目標或 IP 地址。您可以在 AWS Resource Access Manager () 中建立資源共享AWS RAM、建立資源閘道，以及定義資源組態，以在 VPC 中共用資源。

資源閘道

資源閘道是資源所在的 VPC 傳入點。

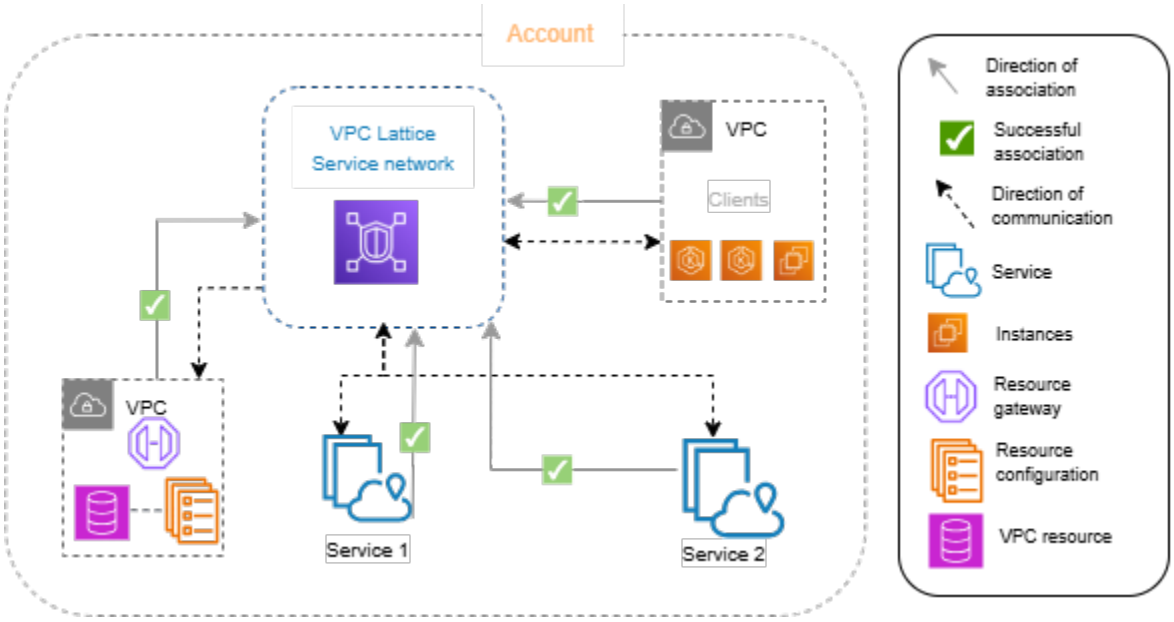
資源組態

資源組態是代表單一資源或一組資源的邏輯物件。資源可以是 IP 地址、網域名稱目標或 Amazon RDS 資料庫。

服務網路

服務和資源組態集合的邏輯界限。用戶端可以位於與服務網路相關聯的 VPC 中。如果與相同服務網路相關聯的用戶端和服務獲得授權，則可以彼此通訊。

在下圖中，用戶端可以與兩個 服務通訊，因為 VPC 和服務與相同的服務網路相關聯。



服務目錄

您擁有或透過 與您的帳戶共用的所有 VPC Lattice 服務的中央登錄檔 AWS RAM。

驗證政策

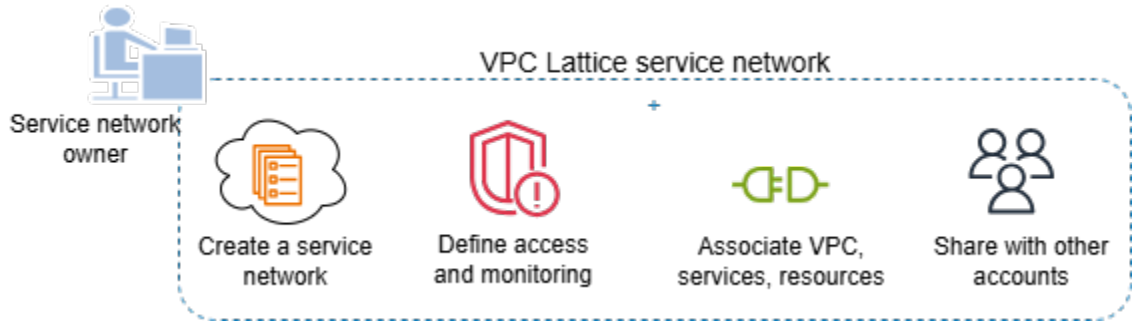
可用來定義 服務存取權的精細授權政策。您可以將個別的身分驗證政策連接至個別 服務或 服務網路。例如，您可以建立政策，說明在 EC2 執行個體的自動擴展群組上執行的付款服務，應如何與在其中執行的帳單服務互動 AWS Lambda。

資源組態不支援 Auth-policies。服務網路的身分驗證政策不適用於服務網路中的資源組態。

角色和責任

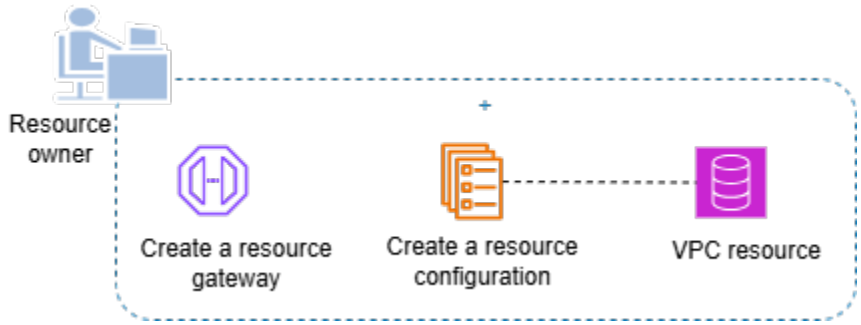
角色決定誰負責 Amazon VPC Lattice 內的資訊設定和流程。通常有兩個角色：服務網路擁有者和服務擁有者，其責任可能會重疊。

服務網路擁有者 – 服務網路擁有者通常是組織中的網路管理員或雲端管理員。服務網路擁有者建立、共用和佈建服務網路。他們也會管理誰可以存取 VPC Lattice 內的服務網路或服務。服務網路擁有者可以為與服務網路相關聯的服務定義粗略的存取設定。這些控制項用於使用身分驗證和授權政策來管理用戶端和服務之間的通訊。如果服務或資源組態與服務網路擁有者的帳戶共用，服務網路擁有者也可以將服務或資源組態與單一或多個服務網路建立關聯。



服務擁有者 – 服務擁有者通常是組織中的軟體開發人員。服務擁有者會在 VPC Lattice 內建立服務、定義路由規則，以及將服務與服務網路建立關聯。他們也可以定義精細的存取設定，限制只能存取已驗證和授權的服務和用戶端。

資源擁有者 – 資源擁有者通常是組織中的軟體開發人員，並擔任資料庫等資源的管理員。資源擁有者會建立資源的資源組態、定義資源組態的存取設定，以及將資源組態與服務網路建立關聯。



功能

以下是 VPC Lattice 提供的核心功能。

服務探索

與服務網路相關聯的 VPCs 中的所有用戶端和服務都可以與相同服務網路中的其他服務通訊。DNS 透過 VPC Lattice 端點引導client-to-serviceservice-to-service流量。當用戶端想要將請求傳送至服

務時，會使用服務的 DNS 名稱。Route 53 Resolver 會將流量傳送至 VPC Lattice，然後識別目的地服務。

連線能力

在網路基礎設施中 AWS 建立Client-to-serviceclient-to-resource連線。當您將 VPC 與服務網路建立關聯時，如果 VPC 中的任何用戶端具有必要的存取權，則可以與服務網路中的服務和資源 （透過資源組態） 連線。VPC Lattice 支援重疊 CIDR 技術。

內部部署存取

您可以使用 VPC 端點 （由 提供支援 AWS PrivateLink) 從 VPC 啟用與服務網路的連線。服務網路類型的 VPC 端點可讓您透過 Direct Connect 和 VPN 從內部部署網路存取服務網路中的 服務和資源。周遊 VPC 對等互連或 AWS Transit Gateway 也可以透過 VPC 端點存取資源和服務的流量。

可觀測性

VPC Lattice 會針對周遊服務網路的每個請求和回應產生指標和日誌，以協助您監控應用程式並進行疑難排解。根據預設，指標會發佈至服務擁有者帳戶。服務擁有者和資源擁有者可以選擇開啟記錄，並接收對其服務和資源的所有用戶端存取/請求的日誌。服務網路擁有者也可以在服務網路上開啟記錄，以記錄 VPCs 中連線至服務網路之用戶端對服務和資源的所有存取/請求。

VPC Lattice 使用下列工具，協助您監控和疑難排解服務： Amazon CloudWatch 日誌群組、Firehose 交付串流和 Amazon S3 儲存貯體。

安全

VPC Lattice 提供架構，可讓您在網路的多層實作防禦策略。第一層是服務、資源組態、VPC 關聯和 VPC 端點類型服務網路的組合。如果沒有 VPC 和服務關聯或 VPC 端點類型的服務網路，用戶端無法存取 服務。同樣地，如果沒有 VPC 和資源組態以及服務關聯或類型為服務網路的 VPC 端點，用戶端就無法存取資源。

第二層可讓使用者將安全群組連接到 VPC 和服務網路之間的關聯。第三層和第四層是可在服務網路層級和服務層級個別套用的身分驗證政策。

可用區域親和性

VPC Lattice 支援路由流量的可用區域 (AZ) 親和性。當用戶端將請求傳送至 VPC Lattice 時，VPC Lattice 會使用與用戶端相同 AZ 的服務或資源 IP 地址來回應。如果該 AZ 無法使用，VPC Lattice 會以來自其他 AZs IP 地址回應。從 VPC Lattice 到目標，路由到目標，可能分散到 AZs。此外，VPC Lattice 中沒有跨可用區域資料傳輸費用。

存取 VPC Lattice

您可以使用下列任一界面來建立、存取和管理 VPC Lattice：

- AWS 管理主控台 – 提供可用來存取 VPC Lattice 的 Web 界面。
- AWS Command Line Interface (AWS CLI) – 為廣泛的 AWS 服務提供命令，包括 VPC Lattice。Windows、MacOS 和 Linux AWS CLI 支援 。如需 CLI 的詳細資訊，請參閱 AWS Command Line Interface。如需 APIs的詳細資訊，請參閱 Amazon VPC Lattice API 參考。
- Kubernetes 專用 VPC Lattice 控制器 – 管理 Kubernetes 叢集的 VPC Lattice 資源。如需搭配 Kubernetes 使用 VPC Lattice 的詳細資訊，請參閱AWS 閘道 API 控制器使用者指南。
- CloudFormation – 協助您建立和設定 AWS 資源的模型。如需詳細資訊，請參閱 Amazon VPC Lattice 資源類型參考。

VPC Lattice 服務端點

端點是做為 AWS Web 服務進入點的 URL。VPC Lattice 支援下列端點類型：

- the section called “IPv4 端點”
- 雙堆疊端點 (同時支援 IPv4 和 IPv6)

當您提出請求時，您可以指定要使用的端點。如果您沒有指定端點，則預設使用 IPv4 端點。若要使用不同的端點類型，您必須在請求中將其指定。如需如何執行此作業的範例，請參閱 the section called “指定端點”。如需可用端點的資料表，請參閱 Amazon VPC Lattice 端點。

IPv4 端點

IPv4 端點僅支援 IPv4 流量。IPv4 端點適用於所有區域。

如果您指定一般端點、`vpc-lattice.amazonaws.com`，我們會使用 `us-east-1` 的端點。若要使用不同的區域，請指定其相關聯的端點。例如，如果您指定 `vpc-lattice.us-east-2.amazonaws.com` 做為端點，我們會將您的請求導向 `us-east-2` 端點。

IPv4 端點名稱使用以下命名慣例：

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

例如，`eu-west-1` 區域的 IPv4 端點名稱是 `vpc-lattice.eu-west-1.amazonaws.com`。

雙堆疊 (IPv4 和 IPv6) 端點

雙堆疊端點同時支援 IPv4 和 IPv6 流量。雙堆疊端點適用於所有區域。當您請求雙堆疊端點時，端點 URL 會解析為 IPv6 或 IPv4 位址，這點取決於您的網路和用戶端使用的通訊協定。

雙堆疊端點名稱使用以下命名慣例：

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

例如，`eu-west-1` 區域的雙堆疊端點名稱是 `vpc-lattice.eu-west-1.api.aws`。

指定端點

下列範例示範如何使用適用於 的 AWS CLI 指定 `us-east-2`區域的端點`vpc-lattice`。

- IPv4

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

- 雙堆疊

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

定價

使用 VPC Lattice，您需要為佈建服務的時間、透過每個服務傳輸的資料量，以及請求數量付費。身為資源擁有者，您需要為往返每個資源的資料付費。身為服務網路擁有者，您需要為與服務網路相關聯的資源組態按小時付費。身為與服務網路建立關聯之 VPC 的消費者，您需要為從 VPC 傳入和傳出服務網路中資源的資料付費。如需詳細資訊，請參閱 Amazon VPC Lattice 定價。

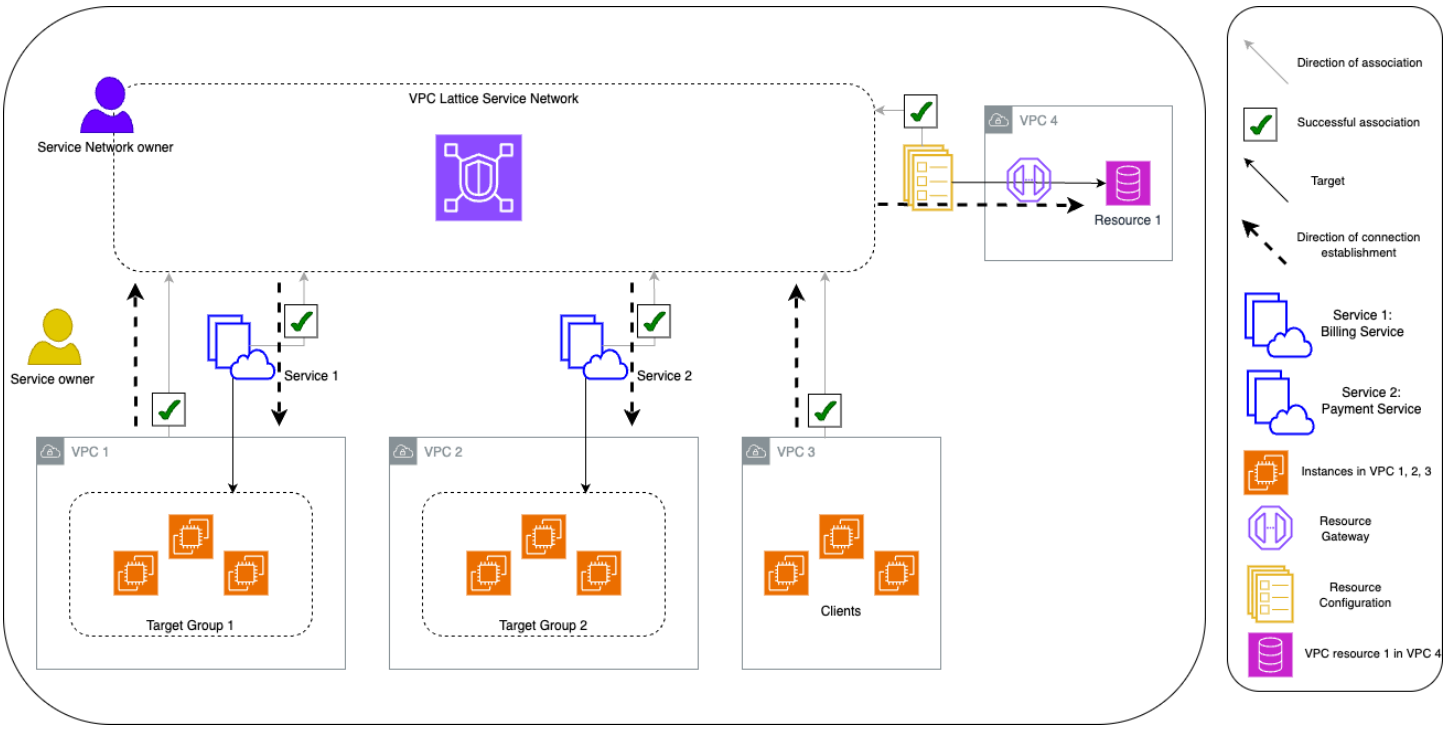
VPC Lattice 的運作方式

VPC Lattice 旨在協助您輕鬆有效地探索、保護、連線和監控其中的所有服務和資源。VPC Lattice 中的每個元件會根據與服務網路的關聯及其存取設定，在服務網路內進行單向或雙向通訊。存取設定包含此通訊所需的身分驗證和授權政策。

下列摘要說明 VPC Lattice 內元件之間的通訊：

- VPC 有兩種方式可以連接到服務網路 - 透過 VPC 關聯和類型為服務網路的 VPC 端點。
- 與服務網路相關聯的服務和資源可以接收來自其 VPCs 也連接到服務網路之用戶端的請求。
- 只有當用戶端位於連線至相同服務網路的 VPC 中時，才能將請求傳送至與服務網路相關聯的 服務和資源。只有當 VPC 透過 VPC 端點連接到服務網路時，周遊 VPC 互連連線、傳輸閘道、Direct Connect 或 VPN 的用戶端流量才能連接資源和服務。
- VPCs 中與服務網路相關聯的服務目標也是用戶端，並且可以將請求傳送到與服務網路相關聯的其他服務和資源。
- VPCs 中與服務網路無關的服務目標不是用戶端，無法將請求傳送至與服務網路相關聯的其他 服務和資源。
- 具有資源但 VPC 與服務網路沒有關聯的 VPCs 用戶端不是用戶端，無法將請求傳送至與服務網路相關聯的其他 服務和資源。

下列流程圖使用範例案例來說明 VPC Lattice 內元件之間的資訊和通訊方向。服務網路有兩個相關聯的服務。服務和所有 VPCs 都是在與服務網路相同的帳戶中建立。這兩個服務都設定為允許來自服務網路的流量。



Service 1 是在 VPC 1 中向目標群組 1 註冊的一組執行個體上執行的計費應用程式。Service 2 是在 VPC 2 中向目標群組 2 註冊的一組執行個體上執行的付款應用程式。VPC 3 位於相同 帳戶中，且具有用戶端，但沒有 服務。資源 1 是在 VPC 4 中具有客戶資料的資料庫。

下列清單依序說明 VPC Lattice 的一般任務工作流程。

1. 建立服務網路

   服務網路擁有者會建立服務網路。

2. 建立服務

   服務擁有者會建立各自的服務，即服務 1 和服務 2。在建立期間，服務擁有者會新增接聽程式，並定義將請求路由到每個服務目標群組的規則。

3. 定義路由

   服務擁有者會為每個服務建立目標群組（目標群組 1 和目標群組 2)。他們透過指定服務執行所在的目標執行個體來執行此操作。它們也會指定這些目標所在的 VPCs。

   在上圖中，實心箭頭代表將流量路由至目標群組的服務，以及路由至資源的資源組態。

   VPC Lattice 支援路由流量的可用區域 (AZ) 親和性。當用戶端將請求傳送至 VPC Lattice 時，VPC Lattice 會回應來自與用戶端相同 AZ 的服務或資源 IP 地址。如果該 AZ 無法使用，VPC Lattice 會

以來自其他 AZs IP 地址回應。從 VPC Lattice 到目標，路由到目標，可能分散到 AZs。此外，VPC Lattice 中沒有跨可用區域資料傳輸費用。

4. 將服務與服務網路建立關聯

   服務網路擁有者或服務擁有者會將服務與服務網路建立關聯。關聯會顯示為箭頭，其核取記號指向服務網路。當您將服務與服務網路建立關聯時，該服務可供與服務網路相關聯的其他 服務以及連線至服務網路之 VPCs中的用戶端探索。

   服務網路和目標群組之間的虛線箭頭會顯示建立連線的方向。使用服務網路將流量傳回用戶端。此圖表中不包含代表傳回流量的箭頭。

5. 建立資源閘道

   資源擁有者會在 VPC 4 中建立資源閘道，以便能夠從用戶端連線至資源 1。

6. 建立資源組態

   資源擁有者會建立資源組態來代表資源 1，並指定資源 1 的資源閘道。

7. 將資源組態與服務網路建立關聯

   服務網路擁有者或資源擁有者會將資源組態與服務網路建立關聯。關聯會顯示為箭頭，其中核取記號指向資源組態中的服務網路。當您將資源組態與服務網路建立關聯時，該資源組態可供與服務網路相關聯的其他 服務以及連線至服務網路之 VPCs中的用戶端探索。

   從服務網路到資源的破折號箭頭代表從用戶端接收請求的資源。使用服務網路將流量傳回用戶端。此圖表中不包含代表傳回流量的箭頭。

8. 將 VPCs與服務網路連線

   VPCs可以透過兩種方式與服務網路連線：將 VPC 與服務網路建立關聯，或建立 VPC 端點。在這裡，服務網路擁有者會將 VPC 1 和 VPC 3 與服務網路建立關聯。使用指向服務網路的核取記號箭頭來顯示關聯。透過這些關聯，VPC 中的任何資源都可以充當用戶端，並且可以向服務網路中的服務提出請求。VPC 1 和服務網路之間的虛線箭頭會顯示建立連線的方向。服務網路只會對服務 1 目標群組鎖定的資源啟動連線。VPC 1 中的任何資源都可以充當用戶端，並啟動與服務網路服務和資源的連線。

   VPC 2 沒有代表關聯的箭頭或核取記號。這表示服務網路擁有者或服務擁有者尚未將 VPC 2 與服務網路建立關聯。這是因為在此範例中，服務 2 只需要接收請求並使用相同的請求傳送回應。換句話說，服務 2 的目標不是用戶端，也不需要向服務網路中的其他服務提出請求。

同樣地，VPC 4 沒有代表關聯的箭頭或核取記號。這表示服務網路擁有者或資源擁有者尚未將 VPC 4 與服務網路建立關聯。這是因為資源 1 只會接收請求，並使用相同的請求傳送回應。它無法向服務網路中的其他 服務和資源提出請求。

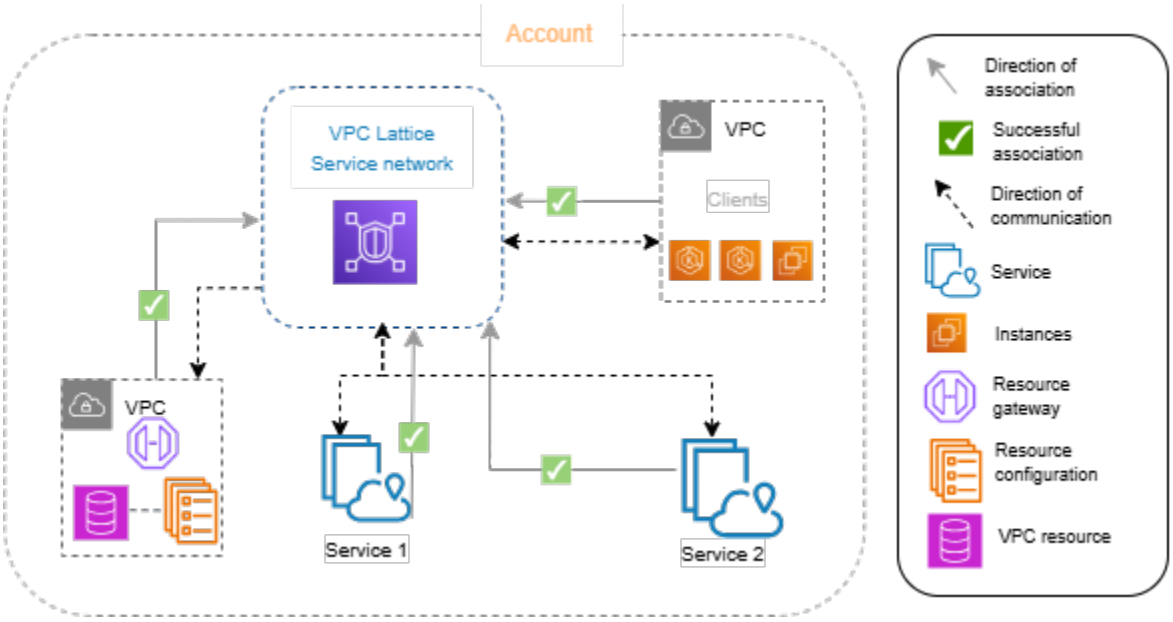
總而言之，流程圖顯示下列案例：

- 僅具有從 VPCs Lattice 到其資源之輸入連線的 VPC。VPC 2 和 VPC 4 代表這些案例。
- 僅具有從資源輸出至 VPC Lattice 之連線的 VPC。VPC 3 代表此案例。
- VPC 具有從 VPC Lattice 到其資源的輸入連線，以及從其資源到 VPC Lattice 的輸出連線。VPC 1 代表此案例。

VPC Lattice 中的服務網路

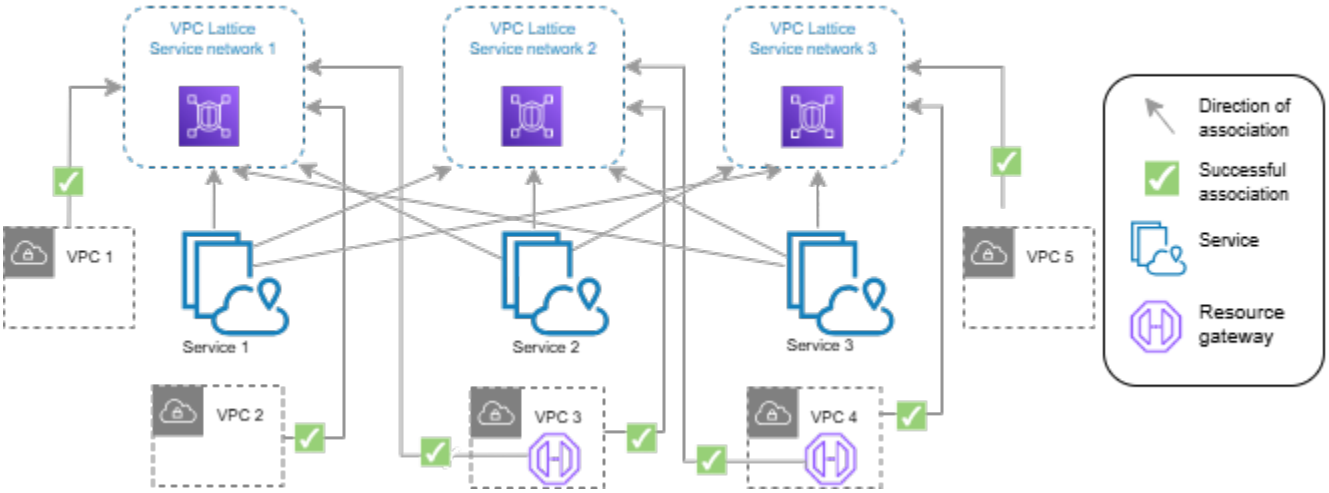
服務網路是服務和資源組態集合的邏輯界限。與網路相關聯的服務和資源組態可以授權用於探索、連線能力、可存取性和可觀測性。若要對網路中的服務和資源組態提出請求，您的服務或用戶端必須位於透過 關聯或透過 VPC 端點連線至服務網路的 VPC 中。

下圖顯示 Amazon VPC Lattice 中典型服務網路的關鍵元件。箭頭上的核取記號表示服務和 VPC 與服務網路相關聯。與服務網路相關聯的 VPC 中的用戶端可以透過服務網路與這兩個服務通訊。



您可以將一或多個服務和資源組態與多個服務網路建立關聯。您也可以使用一個服務網路連接多個 VPCs。您只能透過 關聯將 VPC 連線至一個服務網路。若要將 VPC 連線至多個服務網路，您可以使用服務網路類型的 VPC 端點。如需類型服務網路之 VPC 端點的詳細資訊，請參閱 AWS PrivateLink 使用者指南。

在下圖中，箭頭代表 服務與服務網路之間的關聯，以及 VPCs與服務網路之間的關聯。您可以看到多個服務與多個服務網路相關聯，而多個 VPCs與每個服務網路相關聯。每個 VPC 剛好有一個與服務網路的關聯。不過，VPC 3 和 VPC 4 會連線到兩個服務網路。VPC 3 會透過 VPC 端點連線至服務網路 1。同樣地，VPC 4 會透過 VPC 端點連線至服務網路 2。



如需詳細資訊，請參閱Amazon VPC Lattice 配額。

目錄

建立 VPC Lattice 服務網路

使用 主控台建立服務網路，並選擇性地使用服務、關聯、存取設定和存取日誌進行設定。

使用主控台建立服務網路

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務網路。
3. 選擇建立服務網路。
4. 對於識別符，輸入名稱、選用描述和選用標籤。名稱必須介於 3 到 63 個字元之間。您可以使用小寫字母、數字和連字號。名稱必須以字母或數字開頭和結尾。請勿使用連續連字號。描述最多可有 256 個字元。若要新增標籤，請選擇新增標籤並指定標籤索引鍵和標籤值。

5. （選用）若要建立服務關聯，請從服務關聯、服務中選擇服務。此清單包含您帳戶中的服務，以及從不同帳戶與您共用的任何服務。如果清單中沒有任何服務，您可以選擇建立 VPC Lattice 服務來建立服務。

   或者，若要在建立服務網路之後建立服務關聯，請參閱 the section called “管理服務網路服務關聯”。

6. （選用）若要關聯資源組態，請從資源組態關聯、資源組態中選擇資源組態服務。此清單包含您帳戶中的資源組態，以及從不同帳戶與您共用的任何資源組態。如果清單中沒有任何資源組態，您可以選擇建立 Amazon VPC Lattice 資源組態來建立資源組態。

   或者，若要在建立服務網路之後關聯資源組態，請參閱 the section called “管理服務網路資源關聯”。

7. （選用）若要建立 VPC 的關聯，請選擇新增 VPC 關聯。選取要從 VPC 建立關聯的 VPC，然後從安全群組中選取最多五個安全群組。若要建立安全群組，請選擇建立新安全群組。

   或者，您可以略過此步驟，並使用 VPC 端點（由 提供支援）將 VPC 連接到服務網路 AWS PrivateLink。如需詳細資訊，請參閱《 AWS PrivateLink 使用者指南》中的存取服務網路。

8. 建立服務網路時，您必須決定是否要與其他帳戶共用服務網路。您的選擇是不可變的，而且在您建立服務網路之後無法變更。如果您選擇允許共用，可以透過 與其他帳戶共用服務網路 AWS Resource Access Manager。

   若要與其他帳戶共用您的服務網路，請從 AWS RAM 資源共用中選擇資源共用。

   若要建立資源共享，請前往 AWS RAM 主控台，然後選擇建立資源共享。

9. 對於網路存取，如果您希望相關聯 VPCs 中的用戶端存取此服務網路中的服務，則可以保留預設身分驗證類型 None。若要套用身分驗證政策來控制對服務的存取，請選擇 AWS IAM，並針對身分驗證政策執行下列其中一項動作：

   - 在輸入欄位中輸入政策。如需您可以複製和貼上的政策範例，請選擇政策範例。
   - 選擇套用政策範本，然後選取允許已驗證和未驗證的存取範本。此範本允許來自另一個帳戶的用戶端透過簽署請求（表示已驗證）或以匿名方式（表示未驗證）存取服務。
   - 選擇套用政策範本，然後選取僅允許已驗證的存取範本。此範本僅允許來自另一個帳戶的用戶端透過簽署請求來存取服務（表示已驗證）。

10. （選用）若要開啟存取日誌，請選取存取日誌切換開關，並指定存取日誌的目的地，如下所示：

    - 選取 CloudWatch Log 群組，然後選擇 CloudWatch Log 群組。若要建立日誌群組，請選擇在 CloudWatch 中建立日誌群組。

- 選取 S3 儲存貯體並輸入 S3 儲存貯體路徑，包括任何字首。若要搜尋 S3 儲存貯體，請選擇瀏覽 S3。
- 選取 Kinesis Data Firehose 交付串流，然後選擇交付串流。若要建立交付串流，請選擇在 Kinesis 中建立交付串流。

11. （選用）若要與其他帳戶共用您的服務網路，請從 AWS RAM 資源共用中選擇資源共用。若要建立資源共享，請選擇在 RAM 主控台中建立資源共享。
12. 在摘要區段中檢閱您的組態，然後選擇建立服務網路。

使用 建立服務網路 AWS CLI

使用 create-service-network 命令。此命令只會建立基本服務網路。若要建立功能完整的服務網路，您還必須使用建立服務關聯、VPC 關聯和存取設定的命令。

管理 VPC Lattice 服務網路的關聯

當您將服務或資源組態與服務網路建立關聯時，它可讓連線至服務網路VPCs 中的用戶端向服務和資源組態提出請求。當您將 VPC 與服務網路連線時，它會讓該 VPC 中的所有目標成為用戶端，並與服務網路中的其他服務和資源組態通訊。

服務網路資源關聯的私有 DNS 啟用屬性會覆寫服務網路端點的私有 DNS 啟用屬性和服務網路 VPC 關聯。

如果服務網路擁有者建立服務網路資源關聯，但未啟用私有 DNS，即使服務網路端點或服務網路 VPC 關聯上已啟用私有 DNS，VPC Lattice 也不會在服務網路連線的任何 VPCs 中為該資源組態佈建私有託管區域。

目錄

管理服務網路服務關聯

您可以關聯位於您帳戶中的服務，或從不同帳戶與您共用的服務。這是建立服務網路時的選用步驟。不過，在您建立服務關聯之前，服務網路無法完全運作。如果服務擁有者的帳戶具有必要的存取權，則其服務可以與服務網路建立關聯。如需詳細資訊，請參閱VPC Lattice 的身分型政策範例。

當您刪除服務關聯時，服務就無法再連線到服務網路中的其他服務。

使用主控台管理服務關聯

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務網路。
3. 選取服務網路的名稱以開啟其詳細資訊頁面。
4. 選擇服務關聯索引標籤。
5. 若要建立關聯，請執行下列動作：
   a. 選擇建立關聯。
   b. 從服務中選取服務。若要建立服務，請選擇建立 Amazon VPC Lattice 服務。
   c. （選用）若要新增標籤，請展開服務關聯標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
   d. 選擇儲存變更。
6. 若要刪除關聯，請選取關聯的核取方塊，然後選擇動作、刪除服務關聯。出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 建立服務關聯 AWS CLI

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

使用 刪除服務關聯 AWS CLI

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

管理服務網路資源關聯

資源組態是代表單一資源或一組資源的邏輯物件。您可以關聯位於您帳戶中的資源組態，或從不同帳戶與您共用的資源組態。這是建立服務網路時的選用步驟。如果資源組態擁有者的帳戶具有必要的存取權，則可以將其資源組態與服務網路建立關聯。如需詳細資訊，請參閱 VPC Lattice 的身分型政策範例。

管理服務網路與資源組態之間的關聯

您可以建立或刪除服務網路與資源組態之間的關聯。

使用主控台管理資源組態關聯

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 PrivateLink 和 Lattice 下，選擇服務網路。
3. 選取服務網路的名稱以開啟其詳細資訊頁面。
4. 選擇資源組態關聯索引標籤。
5. 若要建立關聯，請執行下列動作：
   a. 選擇建立關聯。
   b. 針對資源組態，選取資源組態。
   c. 針對 DNS 名稱，選取啟用私有 DNS，以允許 VPC Lattice 根據資源組態的網域名稱，為您的資源組態關聯佈建私有託管區域。
   d. （選用）若要新增標籤，請展開服務關聯標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
   e. 選擇儲存變更。
6. 若要刪除關聯，請選取關聯的核取方塊，然後選擇動作、刪除。出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 建立資源組態關聯 AWS CLI

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

使用 刪除資源組態關聯 AWS CLI

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

管理服務網路 VPC 關聯

如果用戶端位於與服務網路相關聯的 VPCs 中，用戶端可以將請求傳送至與服務網路相關聯的資源組態中指定的服務和資源。流經 VPC 互連連線或傳輸閘道的用戶端流量，只能透過使用類型為服務網路的 VPC 端點的服務網路。

建立 VPC 關聯是建立服務網路時的選用步驟。如果 VPCs的帳戶具有必要的存取權，則網路擁有者可以將 VPC 與服務網路建立關聯。如需詳細資訊，請參閱VPC Lattice 的身分型政策範例。

當您建立與資源組態的 VPC 關聯時，您可以指定私有 DNS 偏好設定。此偏好設定允許 VPC Lattice 代表資源取用者佈建私有託管區域。如需詳細資訊，請參閱the section called “資源提供者的自訂網域名稱”。

當您刪除 VPC 關聯時，VPCs中的用戶端無法再連線到服務網路中的服務。

使用主控台管理 VPC 關聯

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務網路。
3. 選取服務網路的名稱以開啟其詳細資訊頁面。
4. 選擇 VPC 關聯標籤。
5. 若要建立 VPC 關聯，請執行下列動作：

   a. 選擇建立 VPC 關聯。

   b. 選擇新增 VPC 關聯。

   c. 從 VPC 選取 VPC，並從安全群組選取最多五個安全群組。若要建立安全群組，請選擇建立新安全群組。

   d. （選用）若要允許 VPC Lattice 根據資源組態的網域名稱佈建私有託管區域，請針對 DNS 名稱選取啟用 DNS 名稱，然後執行下列動作：

      i. 針對私有 DNS 偏好設定，選取偏好設定。

         如果您選擇所有網域，VPC Lattice 會為資源組態的任何自訂網域名稱佈建私有託管區域。

      ii. （選用）如果您選擇已驗證和指定的網域或指定的網域，請輸入您希望 VPC Lattice 為其佈建託管區域的以逗號分隔的網域清單。VPC Lattice 只有在與您的私有網域清單相符時，才會佈建託管區域。您可以使用萬用字元比對。

   e. （選用）若要新增標籤，請展開 VPC 關聯標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。

   f. 選擇儲存變更。

6. 若要編輯關聯的安全群組，請選取關聯的核取方塊，然後選擇動作、編輯安全群組。視需要新增和移除安全群組。
7. 若要刪除關聯，請選取關聯的核取方塊，然後選擇動作、刪除 VPC 關聯。出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 建立 VPC 關聯 AWS CLI

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

使用 更新 VPC 關聯的安全群組 AWS CLI

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

使用 刪除 VPC 關聯 AWS CLI

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

管理服務網路 VPC 端點關聯

用戶端可以透過其 VPC 中的 VPC 端點（由 提供支援 AWS PrivateLink)，將請求傳送至資源組態中指定的服務和資源。服務網路類型的 VPC 端點會將 VPC 連線至服務網路。從 VPC 外部透過 VPC 對等互連的用戶端流量，Transit Gateway、Direct Connect 或 VPN 可以使用 VPC 端點來連接 服務和資源組態。使用 VPC 端點，您可以將 VPC 連線到多個服務網路。當您在 VPC 中建立 VPC 端點時，來自 VPC 的 IP 地址（而不是來自受管字首清單的 IP 地址）會用來建立與服務網路的連線。

當您建立與資源組態的 VPC 關聯時，您可以指定私有 DNS 偏好設定。此偏好設定允許 VPC Lattice 代表資源取用者佈建私有託管區域。如需詳細資訊，請參閱the section called “資源提供者的自訂網域名稱”。

使用主控台管理 VPC 端點關聯

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務網路。
3. 選取服務網路的名稱以開啟其詳細資訊頁面。
4. 選擇端點關聯索引標籤，以檢視連接到您服務網路的 VPC 端點。
5. 選取 VPC 端點的端點 ID 以開啟其詳細資訊頁面。然後修改或刪除 VPC 端點關聯。

使用主控台建立新的 VPC 端點關聯

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇端點。
3. 選擇建立端點。
4. 針對類型，選擇服務網路。

5. 選取您要連線至 VPC 的服務網路。
6. 選取 VPC、子網路和安全群組。
7. （選用）若要啟用私有 DNS，請選擇啟用私有 DNS。
8. （選用）若要新增標籤，請展開 VPC 關聯標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
9. 選擇建立端點。

若要進一步了解如何連線至服務網路的 VPC 端點，請參閱《 AWS PrivateLink 使用者指南》中的存取服務網路。

編輯 VPC Lattice 服務網路的存取設定

存取設定可讓您設定和管理服務網路的用戶端存取。存取設定包括身分驗證類型和身分驗證政策。驗證政策可協助您驗證和授權流向 VPC Lattice 內服務的流量。服務網路的存取設定不適用於與服務網路相關聯的資源組態。

您可以在服務網路層級、服務層級或兩者套用身分驗證政策。一般而言，身分驗證政策會由網路擁有者或雲端管理員套用。他們可以實作粗略精細的授權，例如，允許來自組織內的已驗證呼叫，或允許符合特定條件的匿名 GET 請求。在服務層級，服務擁有者可以套用精細的控制項，這可能更嚴格。如需詳細資訊，請參閱使用身分驗證政策控制對 VPC Lattice 服務的存取。

使用主控台新增或更新存取政策

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務網路。
3. 選取服務網路的名稱以開啟其詳細資訊頁面。
4. 選擇存取索引標籤以檢查目前的存取設定。
5. 若要更新存取設定，請選擇編輯存取設定。
6. 如果您希望關聯 VPCs中的用戶端存取此服務網路中的服務，請選擇無驗證類型。
7. 若要將資源政策套用至服務網路，請針對驗證類型選擇 AWS IAM，並針對驗證政策執行下列其中一項：
   - 在輸入欄位中輸入政策。例如，您可以複製和貼上的政策，請選擇政策範例。
   - 選擇套用政策範本，然後選取允許已驗證和未驗證的存取範本。此範本允許來自另一個帳戶的用戶端透過簽署請求（表示已驗證）或以匿名方式（表示未驗證）存取服務。
   - 選擇套用政策範本，然後選取僅允許已驗證的存取範本。此範本僅允許來自另一個帳戶的用戶端透過簽署請求來存取服務（表示已驗證）。

8. 選擇儲存變更。

使用 新增或更新存取政策 AWS CLI

使用 put-auth-policy 命令。

編輯 VPC Lattice 服務網路的監控詳細資訊

VPC Lattice 會為每個請求和回應產生指標和日誌，從而更有效地監控和疑難排解應用程式。

您可以啟用存取日誌，並指定日誌的目的地資源。VPC Lattice 可以將日誌傳送至下列資源：CloudWatch Log 群組、Firehose 交付串流和 S3 儲存貯體。

使用主控台啟用存取日誌或更新日誌目的地

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務網路。
3. 選取服務網路的名稱以開啟其詳細資訊頁面。
4. 選擇 Monitoring (監控) 索引標籤。檢查存取日誌，查看是否已啟用存取日誌。
5. 若要啟用或停用存取日誌，請選擇編輯存取日誌，然後開啟或關閉存取日誌切換開關。
6. 啟用存取日誌時，您必須選取交付目的地的類型，然後建立或選擇存取日誌的目的地。您也可以隨時變更交付目的地。例如：
   - 選取 CloudWatch Log 群組，然後選擇 CloudWatch Log 群組。若要建立日誌群組，請選擇在 CloudWatch 中建立日誌群組。
   - 選取 S3 儲存貯體並輸入 S3 儲存貯體路徑，包括任何字首。若要搜尋 S3 儲存貯體，請選擇瀏覽 S3。
   - 選取 Kinesis Data Firehose 交付串流，然後選擇交付串流。若要建立交付串流，請選擇在 Kinesis 中建立交付串流。
7. 選擇儲存變更。

使用 啟用存取日誌 AWS CLI

使用 create-access-log-subscription 命令。

使用 更新日誌目的地 AWS CLI

使用 update-access-log-subscription 命令。

使用 停用存取日誌 AWS CLI

使用 delete-access-log-subscription 命令。

管理 VPC Lattice 服務網路的標籤

標籤可協助您以不同的方式分類服務網路，例如，依用途、擁有者或環境。

您可以將多個標籤新增至每個服務網路。每個服務網路的標籤索引鍵必須是唯一的。如果您使用與服務網路相關聯的金鑰新增標籤，則會更新該標籤的值。您可以使用字母、空格、數字 (UTF-8) 等字元，以及下列特殊字元：+ - = 。_ : / @。不可使用結尾或前方空格。標籤值區分大小寫。

使用主控台新增或刪除標籤

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務網路。
3. 選取服務網路的名稱以開啟其詳細資訊頁面。
4. 選擇 Tags (標籤) 索引標籤。
5. 若要新增標籤，請選擇新增標籤，然後輸入標籤索引鍵和標籤值。若要新增另一個標籤，請再次選擇新增標籤。當您完成新增標籤的作業時，請選擇 Save changes (儲存變更)。
6. 若要刪除標籤，請選取標籤的核取方塊，然後選擇刪除。出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 新增或刪除標籤 AWS CLI

使用 tag-resource 和 untag-resource 命令。

刪除 VPC Lattice 服務網路

您必須先刪除服務網路與任何服務、資源組態、VPC 或 VPC 端點可能具有的所有關聯，才能刪除服務網路。當您刪除服務網路時，我們也會刪除與服務網路相關的所有資源，例如資源政策、身分驗證政策和存取日誌訂閱。

使用主控台刪除服務網路

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務網路。

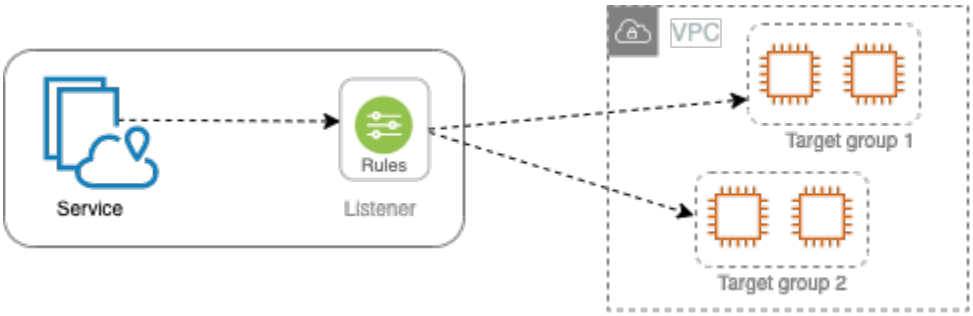
3. 選取服務網路的核取方塊，然後選擇動作、刪除服務網路。
4. 出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 刪除服務網路 AWS CLI

使用 delete-service-network 命令。

VPC Lattice 中的服務

VPC Lattice 內的服務是可獨立部署的軟體單位，可提供特定的任務或函數。服務可以在執行個體、容器或 帳戶或虛擬私有雲端 (VPC) 內的無伺服器函數上執行。服務有一個接聽程式，使用稱為接聽程式規則的規則，您可以設定這些規則來協助將流量路由到目標。支援的目標類型包括 EC2 執行個體、IP 地址、Lambda 函數、Application Load Balancer、Amazon ECS 任務和 Kubernetes Pod。如需詳細資訊，請參閱VPC Lattice 中的目標群組。您可以將服務與多個服務網路建立關聯。下圖顯示 VPC Lattice 中典型服務的關鍵元件。



您可以透過提供服務名稱和描述來建立服務。不過，若要控制和監控您服務的流量，請務必包含存取設定和監控詳細資訊。若要將流量從服務傳送到目標，您必須設定接聽程式並設定規則。若要允許流量從服務網路流向您的服務，您必須將服務與服務網路建立關聯。

與目標的連線有閒置逾時和整體連線逾時。閒置連線逾時為 1 分鐘，之後我們會關閉連線。最長持續時間為 10 分鐘，之後我們不允許透過連線進行新的串流，並開始關閉現有串流的程序。

任務

- 步驟 1：建立 VPC Lattice 服務
- 步驟 2：定義路由
- 步驟 3：建立網路關聯
- 步驟 4：檢閱和建立
- 管理 VPC Lattice 服務的關聯
- 編輯 VPC Lattice 服務的存取設定
- 編輯 VPC Lattice 服務的監控詳細資訊
- 管理 VPC Lattice 服務的標籤
- 為您的 VPC Lattice 服務設定自訂網域名稱
- 為 VPC Lattice 自備憑證 (BYOC)
- 刪除 VPC Lattice 服務

步驟 1：建立 VPC Lattice 服務

使用存取設定和監控詳細資訊建立基本 VPC Lattice 服務。不過，除非您定義其路由組態並將其與服務網路建立關聯，否則服務將無法完全運作。

使用主控台建立基本服務

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 選擇 Create service (建立服務)。
4. 對於識別符，請執行下列動作：

   a. 輸入服務的名稱。名稱必須介於 3-40 個字元之間，並使用小寫字母、數字和連字號。它必須以字母或數字開頭和結尾。請勿使用雙連字號。

   b. （選用）輸入服務網路的描述。您可以在建立期間或之後設定或變更描述。描述最多可有 256 個字元。
5. 若要為您的服務指定自訂網域名稱，請選取指定自訂網域組態，然後輸入自訂網域名稱。

   對於 HTTPS 接聽程式，您可以選擇 VPC Lattice 用來執行 TLS 終止的憑證。如果您現在未選取憑證，您可以在為服務建立 HTTPS 接聽程式時選取憑證。

   對於 TCP 接聽程式，您必須為您的服務指定自訂網域名稱。如果您指定憑證，則不會使用該憑證。反之，您可以在應用程式中執行 TLS 終止。
6. 對於服務存取，如果您希望與服務網路相關聯的 VPCs中的用戶端存取您的服務，請選擇無。若要套用身分驗證政策來控制對服務的存取，請選擇 AWS IAM。若要將資源政策套用至服務，請對驗證政策執行下列其中一項操作：
   - 在輸入欄位中輸入政策。例如，您可以複製和貼上的政策，請選擇政策範例。
   - 選擇套用政策範本，然後選取允許已驗證和未驗證的存取範本。此範本允許來自另一個帳戶的用戶端透過簽署請求（表示已驗證）或以匿名方式（表示未驗證）存取服務。
   - 選擇套用政策範本，然後選取僅允許已驗證的存取範本。此範本僅允許來自另一個帳戶的用戶端透過簽署請求來存取服務（表示已驗證）。
7. （選用）若要啟用存取日誌，請開啟存取日誌切換開關，並指定存取日誌的目的地，如下所示：
   - 選取 CloudWatch Log 群組，然後選擇 CloudWatch Log 群組。若要建立日誌群組，請選擇在 CloudWatch 中建立日誌群組。
   - 選取 S3 儲存貯體並輸入 S3 儲存貯體路徑，包括任何字首。若要搜尋 S3 儲存貯體，請選擇瀏覽 S3。

- 選取 Kinesis Data Firehose 交付串流，然後選擇交付串流。若要建立交付串流，請選擇在 Kinesis 中建立交付串流。

8. （選用）若要與其他 帳戶共用您的服務，請從 AWS RAM 資源共用中選擇資源共用。若要建立資源共享，請選擇在 RAM 主控台中建立資源共享。
9. 若要檢閱您的組態並建立服務，請選擇略過以檢閱和建立。否則，請選擇下一步來定義服務的路由組態。

步驟 2：定義路由

使用接聽程式定義路由組態，讓您的服務可以將流量傳送到您指定的目標。

先決條件

您必須先建立 VPC Lattice 目標群組，才能新增接聽程式。如需詳細資訊，請參閱the section called "建立目標群組"。

使用主控台定義服務的路由

1. 選擇 Add listener (新增接聽程式)。
2. 對於接聽程式名稱，您可以提供自訂接聽程式名稱，或使用接聽程式的通訊協定和連接埠做為接聽程式名稱。您指定的自訂名稱最多可有 63 個字元，且您帳戶中的每個服務都必須是唯一的。有效字元為 a-z、0-9 和連字號 (-)。您不能使用連字號做為第一個或最後一個字元，或緊接在另一個連字號之後。您無法在建立接聽程式之後變更接聽程式的名稱。
3. 選擇通訊協定，然後輸入連接埠號碼。
4. 針對預設動作，選擇 VPC Lattice 目標群組以接收流量，然後選擇要指派給此目標群組的權重。您可以選擇性地為預設動作新增另一個目標群組。選擇新增動作，然後選擇另一個目標群組並指定其權重。
5. （選用）若要新增另一個規則，請選擇新增規則，然後輸入名稱、優先順序、條件和規則的動作。

   您可以為每個規則提供介於 1 到 100 之間的優先順序數字。接聽程式不能擁有多個優先順序相同的規則。依優先順序評估規則，從最低值到最高值。預設規則最後評估。

   針對條件，輸入路徑比對條件的路徑模式。每個字串的大小上限為 200 個字元。比較不區分大小寫。
6. （選用）若要新增標籤，請展開接聽程式標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。

7. 若要檢閱您的組態並建立服務，請選擇略過以檢閱和建立。否則，請選擇下一步，將您的服務與服務網路建立關聯。

步驟 3：建立網路關聯

將您的服務與服務網路建立關聯，以便用戶端可以與其通訊。

使用主控台將服務與服務網路建立關聯

1. 對於 VPC Lattice 服務網路，選取服務網路。若要建立服務網路，請選擇建立 VPC Lattice 網路。您可以將服務與多個服務網路建立關聯。
2. （選用）若要新增標籤，請展開服務網路關聯標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
3. 選擇下一步。

步驟 4：檢閱和建立

使用主控台檢閱組態並建立服務

1. 檢閱您服務的組態。
2. 如果您需要修改服務組態的任何部分，請選擇編輯。
3. 檢閱或編輯組態完成後，請選擇建立 VPC Lattice 服務。
4. 如果您為服務指定了自訂網域名稱，則必須在建立服務之後設定 DNS 路由。如需詳細資訊，請參閱the section called “設定自訂網域名稱”。

管理 VPC Lattice 服務的關聯

當您將服務與服務網路建立關聯時，它可讓用戶端（與服務網路相關聯的 VPC 中的資源）向此服務提出請求。您可以關聯您帳戶中的服務，或從不同帳戶與您共用的服務。建立服務時，此步驟是選用的。不過，在建立之後，除非您將服務與服務網路建立關聯，否則服務無法與其他 服務通訊。如果服務擁有者的帳戶具有必要的存取權，則可以將其服務與服務網路建立關聯。如需詳細資訊，請參閱VPC Lattice 的運作方式。

使用主控台管理服務網路關聯

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。

2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 選取服務的名稱以開啟其詳細資訊頁面。
4. 選擇服務網路關聯索引標籤。
5. 若要建立關聯，請執行下列動作：

   a. 選擇建立關聯。
   b. 從 VPC Lattice 服務網路中選取服務網路。若要建立服務網路，請選擇建立 VPC Lattice 網路。
   c. （選用）若要新增標籤，請展開服務關聯標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
   d. 選擇儲存變更。
6. 若要刪除關聯，請選取關聯的核取方塊，然後選擇動作、刪除網路關聯。出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 建立服務網路關聯 AWS CLI

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

使用 刪除服務網路關聯 AWS CLI

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

編輯 VPC Lattice 服務的存取設定

存取設定可讓您設定和管理 服務的用戶端存取。存取設定包括身分驗證類型和身分驗證政策。驗證政策可協助您驗證和授權流向 VPC Lattice 內服務的流量。

您可以在服務網路層級、服務層級或兩者套用身分驗證政策。在服務層級，服務擁有者可以套用精細的控制項，這可能更嚴格。一般而言，身分驗證政策會由網路擁有者或雲端管理員套用。他們可以實作課程層級的授權，例如，允許來自組織內部的已驗證呼叫，或允許符合特定條件的匿名 GET 請求。如需詳細資訊，請參閱使用身分驗證政策控制對 VPC Lattice 服務的存取。

使用主控台新增或更新存取政策

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 選取服務的名稱以開啟其詳細資訊頁面。

4. 選擇存取索引標籤以檢查目前的存取設定。
5. 若要更新存取設定，請選擇編輯存取設定。
6. 如果您希望關聯服務網路中 VPCs中的用戶端存取您的服務，請針對驗證類型選擇無。
7. 若要套用資源政策來控制對服務的存取，請選擇 AWS IAM for Auth 類型，然後對 Auth 政策執行下列其中一項操作：
   - 在輸入欄位中輸入政策。如需您可以複製和貼上的政策範例，請選擇政策範例。
   - 選擇套用政策範本，然後選取允許已驗證和未驗證的存取範本。此範本允許來自另一個帳戶的用戶端透過簽署請求（表示已驗證）或以匿名方式（表示未驗證）存取服務。
   - 選擇套用政策範本，然後選取僅允許已驗證的存取範本。此範本僅允許來自另一個帳戶的用戶端透過簽署請求來存取服務（表示已驗證）。
8. 選擇儲存變更。

使用 新增或更新存取政策 AWS CLI

使用 put-auth-policy 命令。

編輯 VPC Lattice 服務的監控詳細資訊

VPC Lattice 會為每個請求和回應產生指標和日誌，讓應用程式監控和故障診斷更有效率。

您可以啟用存取日誌，並指定日誌的目的地資源。VPC Lattice 可以將日誌傳送至下列資源：CloudWatch Log 群組、Firehose 交付串流和 S3 儲存貯體。

使用主控台啟用存取日誌或更新日誌目的地

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 選取服務的名稱以開啟其詳細資訊頁面。
4. 選擇監控索引標籤，然後選擇日誌。檢查存取日誌，查看是否已啟用存取日誌。
5. 若要啟用或停用存取日誌，請選擇編輯存取日誌，然後開啟或關閉存取日誌切換開關。
6. 啟用存取日誌時，您必須選取交付目的地的類型，然後建立或選擇存取日誌的目的地。您也可以隨時變更交付目的地。例如：
   - 選取 CloudWatch Log 群組，然後選擇 CloudWatch Log 群組。若要建立日誌群組，請選擇在 CloudWatch 中建立日誌群組。

- 選取 S3 儲存貯體並輸入 S3 儲存貯體路徑，包括任何字首。若要搜尋 S3 儲存貯體，請選擇瀏覽 S3。
- 選取 Kinesis Data Firehose 交付串流，然後選擇交付串流。若要建立交付串流，請選擇在 Kinesis 中建立交付串流。

7. 選擇儲存變更。

使用 啟用存取日誌 AWS CLI

使用 create-access-log-subscription 命令。

使用 更新日誌目的地 AWS CLI

使用 update-access-log-subscription 命令。

使用 停用存取日誌 AWS CLI

使用 delete-access-log-subscription 命令。

管理 VPC Lattice 服務的標籤

標籤可協助您以不同的方式分類服務，例如，依用途、擁有者或環境。

您可以為每個服務新增多個標籤。每個服務的標籤索引鍵必須是唯一的。如果您使用已與服務建立關聯的金鑰新增標籤，則會更新該標籤的值。您可以使用字母、空格、數字 (UTF-8) 和下列特殊字元：+ - = 。_ : / @。不可使用結尾或前方空格。標籤值區分大小寫。

使用主控台新增或刪除標籤

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 選取服務的名稱以開啟其詳細資訊頁面。
4. 選擇 Tags (標籤) 索引標籤。
5. 若要新增標籤，請選擇新增標籤，然後輸入標籤索引鍵和標籤值。若要新增另一個標籤，請再次選擇新增標籤。當您完成新增標籤的作業時，請選擇 Save changes (儲存變更)。
6. 若要刪除標籤，請選取標籤的核取方塊，然後選擇刪除。出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 新增或刪除標籤 AWS CLI

使用 tag-resource 和 untag-resource 命令。

為您的 VPC Lattice 服務設定自訂網域名稱

當您建立新的服務時，VPC Lattice 會使用下列語法為服務產生唯一的完整網域名稱 (FQDN)。

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

不過，VPC Lattice 提供的網域名稱不容易讓使用者記住。自訂網域名稱是更簡單且更直覺URLs，您可以提供給使用者。如果您偏好為您的服務使用自訂網域名稱，例如www.parking.example.com，而不是 VPC Lattice 產生的 DNS 名稱，您可以在建立 VPC Lattice 服務時對其進行設定。當用戶端使用您的自訂網域名稱提出請求時，DNS 伺服器會將其解析為 VPC Lattice 產生的網域名稱。

先決條件

- 您必須擁有服務的註冊網域名稱。如果您還沒有已註冊的網域名稱，您可以透過 Amazon Route 53 或任何其他商業註冊商註冊。
- 若要接收 HTTPS 請求，您必須在 中提供自己的憑證 AWS Certificate Manager。VPC Lattice 不支援預設憑證做為備用憑證。因此，如果您未提供與自訂網域名稱對應的 SSL/TLS 憑證，則與自訂網域名稱的所有 HTTPS 連線都會失敗。如需詳細資訊，請參閱為 VPC Lattice 自備憑證 (BYOC)。

限制和考量事項

- 您無法為服務擁有多個自訂網域名稱。
- 建立服務之後，您無法修改自訂網域名稱。
- 自訂網域名稱對於服務網路必須是唯一的。這表示無法使用相同服務網路中已存在的自訂網域名稱（適用於其他服務）來建立服務。

下列程序說明如何為您的服務設定自訂網域名稱。

AWS 管理主控台

為您的服務設定自訂網域名稱

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務。

3. 選擇建立服務。您已導覽至步驟 1：建立服務。
4. 在自訂網域組態區段中，選擇指定自訂網域組態。
5. 輸入您的自訂網域名稱。
6. 若要提供 HTTPS 請求，請在自訂 SSL/TLS 憑證中選取符合您自訂網域名稱的 SSL/TLS 憑證。如果您還沒有憑證，或不想立即新增憑證，您可以在建立 HTTPS 接聽程式時新增憑證。不過，如果沒有憑證，您的自訂網域名稱將無法提供 HTTPS 請求。如需詳細資訊，請參閱新增 HTTPS 接聽程式。
7. 當您完成新增所有其他資訊以建立服務時，請選擇建立。

AWS CLI

為您的服務設定自訂網域名稱

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
```

在上述命令中，針對 `--name`輸入服務的名稱。針對 `--custom-domain-name`，輸入服務的網域名稱，例如 `parking.example.com`。在 ACM `--certificate-arn` 中輸入憑證的 ARN。憑證 ARN 可在您的 帳戶中使用 AWS Certificate Manager。

將自訂網域名稱與您的服務建立關聯

首先，如果您尚未這麼做，請註冊您的自訂網域名稱。網際網路名稱和數字指派公司 (ICANN) 負責管理網際網路上的網域名稱。您可以使用網域名稱註冊商註冊網域名稱，這是一家 ICANN 認可的組織，專門管理網域名稱的註冊。您的網站註冊商網站將為註冊您的網域名稱提供詳細指示和定價資訊。如需詳細資訊，請參閱下列資源：

- 若要使用 Amazon Route 53 註冊網域名稱，請參閱 Amazon Route 53 開發人員指南中的使用 Route 53 註冊網域名稱。
- 如需這類註冊機構的清單，請參閱認可的註冊機構目錄。

接著，使用您的 DNS 服務，例如網域註冊商，建立記錄以將查詢路由到您的服務。如需詳細資訊，請參閱您的 DNS 服務文件。或者，您可以使用 Route 53 做為您的 DNS 服務。

如果您使用的是 Route 53，您可以使用別名記錄或 CNAME 記錄，將查詢路由到您的服務。我們建議您使用別名記錄，因為您可以在 DNS 命名空間的頂端節點建立別名記錄，也稱為區域頂點。

如果您使用的是 Route 53，您必須先建立託管區域，其中包含如何在網際網路上為網域路由流量的相關資訊。建立私有或公有託管區域之後，請建立記錄，讓您的自訂網域名稱 例如 `parking.example.com`對應至 VPC Lattice 自動產生的網域名稱，例如 `my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws`。如果沒有此映射，您的自訂網域名稱將無法在 VPC Lattice 中運作。

下列程序說明如何使用 Route 53 建立私有或公有託管區域

AWS 管理主控台

若要建立別名記錄，以使用 Route 53 將查詢路由到服務，請參閱將流量路由到 Amazon VPC Lattice 服務網域端點。

為您的服務使用 VPC Lattice 產生的網域名稱，例如 `my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws` 值。您可以在服務頁面上的 VPC Lattice 主控台中找到此自動產生的網域名稱。

AWS CLI

在託管區域中建立別名記錄

1. 取得您服務的 VPC Lattice 產生的網域名稱 （ 例如 `my-service-02031c045478f6ddf1.7d67968.vpc-lattice-svcs.us-west-2.on.aws`)。
2. 若要設定別名，請使用下列命令。

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

對於 `change-set.json` 檔案，使用下列 JSON 範例中的內容建立 JSON 檔案，並將其儲存在本機電腦上。將上述命令中的 *file#//~/Desktop/change-set.json* 取代為本機機器中儲存的 JSON 檔案路徑。請注意，下列 JSON 中的「類型」可以是 A 或 AAAA 記錄類型。

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

為 VPC Lattice 自備憑證 (BYOC)

若要提供 HTTPS 請求，您必須先在 (ACM) 中 AWS Certificate Manager 備妥自己的 SSL/TLS 憑證，才能設定自訂網域名稱。這些憑證必須具有符合您服務的自訂網域名稱的主體別名 (SAN) 或通用名稱 (CN)。如果 SAN 存在，我們只會在 SAN 清單中檢查相符項目。如果 SAN 不存在，我們會檢查 CN 中的相符項目。

VPC Lattice 使用伺服器名稱指示 (SNI) 提供 HTTPS 請求。DNS 會根據自訂網域名稱和符合此網域名稱的憑證，將 HTTPS 請求路由到您的 VPC Lattice 服務。若要請求 ACM 中網域名稱的 SSL/TLS 憑證或將憑證匯入 ACM，請參閱AWS Certificate Manager 《 使用者指南》中的發行和管理憑證和匯入憑證。如果您無法在 ACM 中請求或匯入自己的憑證，請使用 VPC Lattice 產生的網域名稱和憑證。

VPC Lattice 每個服務只接受一個自訂憑證。不過，您可以針對多個自訂網域使用自訂憑證。這表示您可以針對使用自訂網域名稱建立的所有 VPC Lattice 服務使用相同的憑證。

若要使用 ACM 主控台檢視您的憑證，請開啟憑證，然後選取憑證 ID。您應該會在關聯的資源下看到與該憑證相關聯的 VPC Lattice 服務。

限制及考量

- VPC Lattice 允許在關聯憑證的主體別名 (SAN) 或通用名稱 (CN) 深一級的萬用字元比對。例如，如果您使用自訂網域名稱建立服務，`parking.example.com`並將您自己的憑證與 SAN 建立關聯`*.example.com`。當 的請求傳入時`parking.example.com`，VPC Lattice 會將 SAN 與具有頂點網域 的任何網域名稱相符`example.com`。不過，如果您有自訂網域，`parking.different.example.com`且憑證有 SAN`*.example.com`，則請求會失敗。

- VPC Lattice 支援單一層級的萬用字元網域比對。這表示萬用字元只能用作第一層子網域，而且只能保護一個子網域層級。例如，如果您憑證的 SAN 是 `*.example.com`，則`parking.*.example.com`不支援 。
- VPC Lattice 支援每個網域名稱一個萬用字元。這表示 `*.*.example.com` 是無效的。如需詳細資訊，請參閱AWS Certificate Manager 《 使用者指南》中的請求公有憑證。
- VPC Lattice 僅支援具有 2048 位元 RSA 金鑰的憑證。
- ACM 中的 SSL/TLS 憑證必須與您與其建立關聯的 VPC Lattice 服務位於相同的區域。

保護憑證的私有金鑰

當您使用 ACM 請求 SSL/TLS 憑證時，ACM 會產生公有/私有金鑰對。當您匯入憑證時，會產生金鑰對。公有金鑰會成為憑證的一部分。為了安全存放私有金鑰，ACM 會使用 AWS KMS稱為 KMS 金鑰的另一個金鑰與別名 aws/acm。 AWS KMS 使用此金鑰來加密憑證的私有金鑰。如需詳細資訊，請參閱《 AWS Certificate Manager 使用者指南》中的 AWS Certificate Manager中的資料保護。

VPC Lattice AWS 使用 TLS Connection Manager，此服務只能存取 AWS 服務，以保護和使用憑證的私有金鑰。當您使用 ACM 憑證建立 VPC Lattice 服務時，VPC Lattice 會將您的憑證與 AWS TLS Connection Manager 建立關聯。我們會針對您的 AWS 受 AWS KMS 管金鑰在 中建立授予，藉此達成此目的。此授權允許 TLS Connection Manager 使用 AWS KMS 解密憑證的私有金鑰。TLS Connection Manager 使用憑證和解密的 （純文字） 私有金鑰，與 VPC Lattice 服務的用戶端建立安全連線 (SSL/TLS 工作階段）。當憑證與 VPC Lattice 服務取消關聯時，授權會淘汰。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的授權。

如需詳細資訊，請參閱靜態加密。

刪除 VPC Lattice 服務

若要刪除 VPC Lattice 服務，您必須先刪除服務可能與任何服務網路的所有關聯。如果您刪除服務，也會刪除與服務相關的所有資源，例如資源政策、身分驗證政策、接聽程式、接聽程式規則和存取日誌訂閱。

使用主控台刪除服務

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 在服務頁面上，選取您要刪除的服務，然後選擇動作、刪除服務。

4. 出現確認提示時，請選擇刪除。

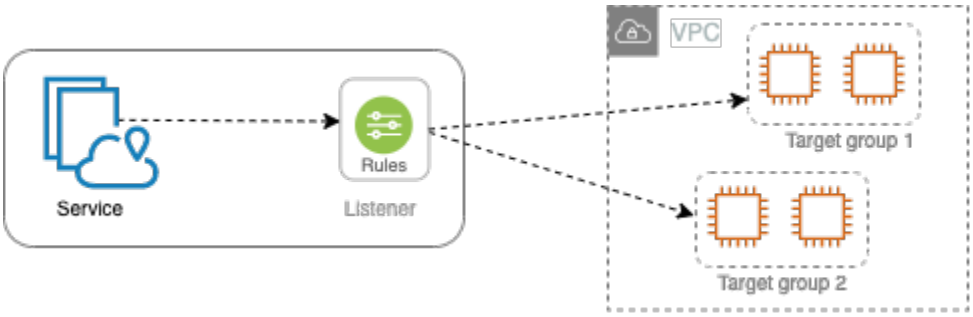
使用 刪除服務 AWS CLI

使用 delete-service 命令。

VPC Lattice 中的目標群組

VPC Lattice 目標群組是執行應用程式或服務的目標或運算資源集合。支援的目標類型包括 EC2 執行個體、IP 地址、Lambda 函數、Application Load Balancer、Amazon ECS 任務和 Kubernetes Pod。您也可以將現有的服務連接到目標群組。如需搭配 VPC Lattice 使用 Kubernetes 的詳細資訊，請參閱AWS 閘道 API 控制器使用者指南。

每個目標群組會用來將請求轉送到一個或多個註冊的目標。當您建立接聽程式規則時，您可以指定目標群組和條件。規則的條件符合時，會將流量轉送到對應的目標群組。您可以針對不同類型的請求，建立不同的目標群組。例如，為一般請求建立一個目標群組，為包含特定規則條件的請求建立其他目標群組，例如路徑或標頭值。



您可以為每個目標群組定義服務的運作狀態檢查設定。除非您在建立目標群組時覆寫這些設定，或是在之後修改設定，否則每個目標群組都會使用預設的運作狀態檢查設定。在您為接聽程式在規則中指定目標群組之後，服務會持續監控向目標群組註冊的所有目標的運作狀態。服務會將請求路由至運作狀態良好的已註冊目標。

若要在服務接聽程式的規則中指定目標群組，目標群組必須與服務位於相同的帳戶中。

VPC Lattice 目標群組類似於 Elastic Load Balancing 提供的目標群組，但無法互換。

目錄

- VPC Lattice 中的 HTTP 目標
- Lambda 函數作為 VPC Lattice 中的目標
- Application Load Balancer 作為 VPC Lattice 中的目標
- 通訊協定版本
- VPC Lattice 目標群組的標籤
- 刪除 VPC Lattice 目標群組

建立 VPC Lattice 目標群組

您會向目標群組註冊您的目標。根據預設，VPC Lattice 服務會使用您為目標群組指定的連接埠和通訊協定，將請求傳送至已註冊的目標。在透過目標群組來註冊每個目標時，您可以覆寫此埠號。

若要將流量路由到目標群組中的目標，請在建立接聽程式或為接聽程式建立規則時，於動作中指定目標群組。如需詳細資訊，請參閱VPC Lattice 服務的接聽程式規則。您可以在多個接聽程式中指定相同的目標群組，但這些接聽程式必須屬於相同的服務。若要搭配服務使用目標群組，您必須驗證目標群組並未由任何其他服務的接聽程式使用。

您可以隨時從目標群組新增或移除目標。如需詳細資訊，請參閱向 VPC Lattice 目標群組註冊目標。您也可以修改目標群組的運作狀態檢查設定。如需詳細資訊，請參閱VPC Lattice 目標群組的運作狀態檢查。

建立目標群組

您可以建立目標群組並選擇性地註冊目標，如下所示。

使用主控台來建立目標群組

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇目標群組。
3. 選擇 Create target group (建立目標群組)。
4. 對於選擇目標類型，請執行下列其中一項操作：
    - 選擇執行個體，依執行個體 ID 註冊目標。
    - 選擇 IP 地址，依 IP 地址註冊目標。
    - 選擇 Lambda 函數將 Lambda 函數註冊為目標。
    - 選擇 Application Load Balancer，將 Application Load Balancer 註冊為目標。

5. 針對 Target group name (目標群組名稱)，輸入目標群組的名稱。此名稱對於每個 AWS 區域中的帳戶必須是唯一的，最多可有 32 個字元，只能包含英數字元或連字號，且開頭或結尾不得為連字號。
6. 對於通訊協定和連接埠，您可以視需要修改預設值。預設通訊協定為 HTTPS，預設連接埠為 443。

   如果目標類型是 Lambda 函數，則無法指定通訊協定或連接埠。
7. 針對 IP 地址類型，選擇 IPv4 以 IPv4 地址註冊目標，或選擇 IPv6 以 IPv6 地址註冊目標。建立目標群組後，您無法變更此設定。

   只有在目標類型為 IP 地址時，才能使用此選項。
8. 針對 VPC (VPC) 選擇虛擬私有雲端 (VPC)。

   如果目標類型為 Lambda 函數，則無法使用此選項。
9. 對於通訊協定版本，視需要修改預設值。預設值為 HTTP1。

   如果目標類型為 Lambda 函數，則無法使用此選項。
10. 對於運作狀態檢查，視需要修改預設設定。如需詳細資訊，請參閱VPC Lattice 目標群組的運作狀態檢查。

    如果目標類型為 Lambda 函數，則無法使用運作狀態檢查。
11. 針對 Lambda 事件結構版本，選擇版本。如需詳細資訊，請參閱the section called “從 VPC Lattice 服務接收事件”。

    只有在目標類型為 Lambda 函數時，才能使用此選項
12. （選用）若要新增標籤，請展開標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
13. 選擇下一步。
14. 對於註冊目標，您可以略過此步驟或新增目標，如下所示：
    - 如果目標類型為執行個體，請選取執行個體，輸入連接埠，然後選擇包含為以下待定的項目。
    - 如果目標類型是 IP 地址，請執行下列動作：
      a. 對於選擇網路，請保留您為目標群組選取的 VPC，或選擇其他私有 IP 地址。
      b. 針對指定 IPs和定義連接埠，輸入 IP 地址並輸入連接埠。預設連接埠是目標群組連接埠。
      c. 選擇包含為下方待處理項目。

- 如果目標類型是 Lambda 函數，請選擇 Lambda 函數。若要建立 Lambda 函數，請選擇建立新的 Lambda 函數。
- 如果目標類型是 Application Load Balancer，請選擇 Application Load Balancer。若要建立 Application Load Balancer，請選擇建立 Application Load Balancer。

15. 選擇 Create target group (建立目標群組)。

    VPC Lattice 可能需要幾分鐘的時間來註冊目標。如需詳細資訊，請參閱 為什麼我的 DNS 變更在 Route 53 和公有解析程式中傳播需要這麼長的時間？

使用 建立目標群組 AWS CLI

使用 create-target-group 命令來建立目標群組，並使用 register-targets 命令來新增目標。

共用子網路

參與者可以在共用 VPC 中建立 VPC Lattice 目標群組。下列規則適用於共用子網路：

- VPC Lattice 服務的所有部分，例如接聽程式、目標群組和目標，都必須由相同的帳戶建立。它們可以在 VPC Lattice 服務擁有者擁有或共用的子網路中建立。
- 向目標群組註冊的目標必須由與目標群組相同的帳戶建立。
- 只有 VPC 擁有者可以將 VPC 與服務網路建立關聯。與服務網路相關聯的共用 VPC 中的參與者資源可以將請求傳送至與服務網路相關聯的服務。不過，管理員可以使用安全群組、網路 ACLs 或身分驗證政策來防止這種情況。

如需 VPC Lattice 可共用資源的詳細資訊，請參閱 共用 VPC Lattice 實體。

向 VPC Lattice 目標群組註冊目標

您的服務可做為用戶端的單一聯絡點，並將傳入流量分配到運作狀態良好的已註冊目標。您可以利用一個或多個群組來登錄每個目標。

如果應用程式的需求增加，您可以向一或多個目標群組註冊其他目標，以處理需求。一旦註冊程序完成且目標通過初始運作狀態檢查，服務就會開始將請求路由到新註冊的目標。

如果對您應用程式的需求減少，或者您需要為目標提供服務，可以從目標群組取消目標的登錄。取消目標的登錄，會將該目標從目標群組中移除，但不會影響到目標。一旦取消註冊，服務就會停止將請求路

由到目標。目標會進入 DRAINING 狀態，直到處理中的請求已完成。當您準備讓目標再繼續接收請求時，可以將目標註冊到目標群組。

目標群組的目標類型會決定您向該目標群組註冊目標的方式。如需詳細資訊，請參閱Target type (目標類型)。

使用下列主控台程序來註冊或取消註冊目標。或者，從 使用 register-targets 和 deregister-targets 命令 AWS CLI。

目錄

根據執行個體 ID 來登記或取消登記目標

目標執行個體必須位於您為目標群組指定的虛擬私有雲端 (VPC) 中。在註冊時，執行個體也必須處於 running 狀態。

當您依執行個體 ID 註冊目標時，您可以將服務與 Auto Scaling 群組搭配使用。將目標群組連接至 Auto Scaling 群組且群組向外擴展後，Auto Scaling 群組啟動的執行個體會自動向目標群組註冊。如果分離目標群組與 Auto Scaling 群組的連結，會自動從該目標群組中取消註冊執行個體。如需詳細資訊，請參閱《Amazon EC2 Auto Scaling 使用者指南》中的使用 VPC Lattice 目標群組將流量路由到您的 Auto Scaling 群組。 Amazon EC2 Auto Scaling

使用主控台根據執行個體 ID 來註冊或取消註冊目標

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇目標群組。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 選擇 Targets (目標) 標籤。
5. 若要註冊執行個體，請選擇註冊目標。選取執行個體，輸入執行個體連接埠，然後選擇包含為以下待定項目。新增執行個體完成後，請選擇註冊目標。
6. 若要取消註冊執行個體，請選取執行個體，然後選擇取消註冊。

根據 IP 地址來登記或取消登記目標

目標 IP 地址必須來自您為目標群組指定的 VPC 子網路。您無法在相同的 VPC 中註冊其他服務的 IP 地址。您無法註冊 VPC 端點或可公開路由的 IP 地址。

使用主控台根據 IP 地址來註冊或取消註冊目標

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇目標群組。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 選擇 Targets (目標) 標籤。
5. 若要註冊 IP 地址，請選擇註冊目標。為每個 IP 地址選取網路，輸入 IP 地址和連接埠，然後選擇包含為下方待處理項目。完成指定地址後，請選擇註冊目標。
6. 若要取消註冊 IP 地址，請選取 IP 地址，然後選擇取消註冊。

註冊或取消註冊 Lambda 函數

您可以向目標群組註冊單一 Lambda 函數。如果您不再需要將流量傳送到您的 Lambda 函數，則可以將它取消註冊。取消註冊 Lambda 函數之後，傳輸中的請求會失敗，出現 HTTP 5XX 錯誤。最好建立新的目標群組，而不是取代目標群組的 Lambda 函數。

使用主控台註冊或取消註冊 Lambda 函數

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇目標群組。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 選擇 Targets (目標) 標籤。
5. 如果沒有註冊 Lambda 函數，請選擇註冊目標。選取 Lambda 函數，然後選擇註冊目標。
6. 若要取消註冊 Lambda 函數，請選擇 Deregister (取消註冊)。出現確認提示時，請輸入 **`confirm`**，然後選擇取消註冊。

註冊或取消註冊 Application Load Balancer

您可以向每個目標群組註冊單一 Application Load Balancer。如果您不再需要將流量傳送到負載平衡器，您可以取消註冊。取消註冊負載平衡器之後，傳輸中的請求會失敗並出現 HTTP 5XX 錯誤。最好建立新的目標群組，而不是取代目標群組的 Application Load Balancer。

使用主控台註冊或取消註冊 Application Load Balancer

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇目標群組。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 選擇 Targets (目標) 標籤。
5. 如果沒有註冊 Application Load Balancer，請選擇註冊目標。選取 Application Load Balancer，然後選擇註冊目標。
6. 若要取消註冊 Application Load Balancer，請選擇取消註冊。出現確認提示時，請輸入 **`confirm`**，然後選擇取消註冊。

VPC Lattice 目標群組的運作狀態檢查

您的服務會定期將請求傳送至其註冊的目標，以測試其狀態。這些測試稱為運作狀況檢查。

每個 VPC Lattice 服務只會將請求路由至運作狀態良好的目標。每個服務會使用目標註冊的目標群組的運作狀態檢查設定，來檢查每個目標的運作狀態。目標註冊後，必須通過一次運作狀態檢查，才算運作狀態良好。在每次運作狀態檢查完成後，服務會關閉為運作狀態檢查建立的連線。

限制和考量事項

- 當目標群組通訊協定版本為 HTTP1 時，依預設會啟用運作狀態檢查。
- 當目標群組通訊協定版本為 HTTP2 時，根據預設不會啟用運作狀態檢查。不過，您可以啟用運作狀態檢查，並手動將通訊協定版本設定為 HTTP1 或 HTTP2。
- 運作狀態檢查不支援 gRPC 目標群組通訊協定版本。不過，如果您啟用運作狀態檢查，則必須將運作狀態檢查通訊協定版本指定為 HTTP1 或 HTTP2。
- 運作狀態檢查不支援 Lambda 目標群組。
- 運作狀態檢查不支援 Application Load Balancer 目標群組。不過，您可以使用 Elastic Load Balancing 啟用 Application Load Balancer 目標的運作狀態檢查。如需詳細資訊，請參閱《Application Load Balancer 使用者指南》中的目標群組運作狀態檢查。

運作狀態檢查設定

您需要按下表中的描述為目標群組中的目標設定運作狀態檢查。表中使用的設定名稱是 API 中使用的名稱。服務會使用指定的連接埠、通訊協定和 ping 路徑，每隔 HealthCheckIntervalSeconds

秒將運作狀態檢查請求傳送至每個已註冊的目標。每個運作狀態檢查請求是各自獨立，且在整個間隔內持續保持此結果。目標回應所花的時間不影響下次運作狀態檢查請求的間隔。如果運作狀態檢查超過 UnhealthyThresholdCount 連續失敗，服務會將目標停止服務。當運作狀態檢查超過 HealthyThresholdCount 連續成功時，服務會將目標恢復服務。

設定	Description
HealthCheckProtocol	服務在對目標執行運作狀態檢查時使用的通訊協定。可能的通訊協定是 HTTP 和 HTTPS。預設為 HTTP 通訊協定。
HealthCheckPort	服務在對目標執行運作狀態檢查時使用的連接埠。預設值是使用每個目標接收來自服務的流量的連接埠。
HealthCheckPath	目標上運作狀態檢查的目的地。 如果通訊協定版本為 HTTP1 或 HTTP2，請指定有效的 URI (/path？query)。預設為 /.
HealthCheckTimeoutSeconds	以秒為單位的時間量，若目標在此期間內毫無回應即表示運作狀態檢查失敗。範圍為 1–120 秒。如果目標類型為 INSTANCE或 ，則預設值為 5 秒IP。指定 0 將此設定重設為其預設值。
HealthCheckIntervalSeconds	個別目標每次執行運作狀態檢查的大約間隔時間量，以秒為單位。範圍介於 5–300 秒之間。如果目標類型為 INSTANCE或 ，則預設值為 30 秒IP。指定 0 將此設定重設為其預設值。
HealthyThresholdCount	在運作狀態不佳的目標視為正常之前，所需的連續成功運作狀態檢查次數。範圍介於 2–10 之間。預設值為 5。指定 0 將此設定重設為其預設值。
UnhealthyThresholdCount	在將目標視為運作狀態不良前，必要的連續運作狀態檢查失敗次數。範圍介於 2–10 之間。預設為 2。指定 0 將此設定重設為其預設值。

設定	Description
Matcher	檢查是否收到來自目標的成功回應時所使用的代碼。這些在主控台中稱為成功代碼。 如果通訊協定版本為 HTTP1 或 HTTP2，則可能的值為 200 到 499。您可以指定多個值 (例如，"200,202") 或值範圍 (例如，"200-299")。預設值為 200。 目前不支援 gRPC 的運作狀態檢查通訊協定版本。不過，如果您的目標群組通訊協定版本是 gRPC，您可以在運作狀態檢查組態中指定 HTTP1 或 HTTP2 通訊協定版本。

檢查目標的運作狀態

您可以檢查已向目標群組註冊的各個目標的運作狀態。

使用主控台檢查目標的運作狀態

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇目標群組。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在 Targets (目標) 標籤，Health status (運作狀態) 欄指出各目標的狀態。如果狀態是 以外的任何值`Healthy`，運作狀態詳細資訊欄會包含更多資訊。

使用 檢查目標的運作狀態 AWS CLI

使用 list-targets 命令。此命令的輸出包含目標的運作狀態。如果狀態為 `Healthy` 以外的任何值，則輸出也會包含原因代碼。

接收有關狀態不良目標的電子郵件通知

使用 CloudWatch 警示啟動 Lambda 函數，以傳送運作狀態不佳目標的詳細資訊。

修改運作狀態檢查設定

您可以隨時修改目標群組的運作狀態檢查設定。

使用主控台修改運作狀態檢查設定

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇目標群組。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在運作狀態檢查索引標籤上的運作狀態檢查設定區段中，選擇編輯。
5. 視需要修改運作狀態檢查設定。
6. 選擇儲存變更。

使用 修改運作狀態檢查設定 AWS CLI

使用 update-target-group 命令。

路由組態

根據預設，服務會使用您在建立目標群組時指定的通訊協定和連接埠號碼，將請求路由至其目標。或者，您可以在使用目標群組來登錄目標時，覆寫用來將流量轉傳到目標的連接埠。

目標群組支援下列的通訊協定和連接埠：

- 通訊協定：HTTP、HTTPS、TCP
- Ports (連接埠)：1-65535

如果使用 HTTPS 通訊協定設定目標群組或使用 HTTPS 運作狀態檢查，目標的 TLS 連線會使用接聽程式的安全政策。VPC Lattice 會使用您在目標上安裝的憑證，與目標建立 TLS 連線。VPC Lattice 不會驗證這些憑證。因此，您可以使用自我簽署的憑證或已過期的憑證。VPC Lattice 與目標之間的流量會在封包層級進行驗證，因此即使目標上的憑證無效，也不會有man-in-the-middle攻擊或詐騙的風險。

只有 TLS 接聽程式才支援 TCP 目標群組。

路由演算法

根據預設，循環配置路由演算法會用來將請求路由至運作狀態良好的目標。

當 VPC Lattice 服務收到請求時，會使用下列程序：

1. 以優先順序評估接聽程序的規則，以決定要套用哪個規則。
2. 使用預設循環配置演算法，從規則動作的目標群組中選取目標。即使一個目標向多個目標群組註冊，每個目標群組的路由都是獨立運作。

如果目標群組只包含運作狀態不佳的目標，則請求會路由至所有目標，無論其運作狀態為何。這表示如果所有目標同時未通過運作狀態檢查，VPC Lattice 服務會無法開啟。故障開啟的效果是根據循環配置演算法，允許所有目標的流量，無論其運作狀態為何。

VPC Lattice 支援路由流量的可用區域 (AZ) 親和性。當用戶端將請求傳送至 VPC Lattice 時，VPC Lattice 會使用與用戶端相同 AZ 的服務或資源 IP 地址來回應。如果無法使用該 AZ，VPC Lattice 會以來自其他 AZs IP 地址回應。從 VPC Lattice 到目標，路由到目標，可能分散到 AZs。此外，VPC Lattice 中沒有跨可用區域資料傳輸費用。

Target type (目標類型)

建立目標群組時，您會指定其目標類型，這會決定您對此目標群組註冊目標時指定的目標類型。在建立目標群組之後，您無法變更其目標類型。

下列是可能的目標類型：

`INSTANCE`

以執行個體 ID 來指定目標。

`IP`

目標為 IP 地址。

`LAMBDA`

目標是 Lambda 函數。

`ALB`

目標是 Application Load Balancer。

考量事項

- 當目標類型為 時IP，您必須從目標群組的 VPC 子網路指定 IP 地址。如果您需要從此 VPC 外部註冊 IP 地址，請建立 類型的目標群組，ALB並向 Application Load Balancer 註冊 IP 地址。
- 當目標類型為 時IP，您無法註冊 VPC 端點或可公開路由的 IP 地址。
- 當目標類型為 時LAMBDA，您可以註冊單一 Lambda 函數。當服務收到 Lambda 函數的請求時，它會叫用 Lambda 函數。如果您想要將多個 Lambda 函數註冊到服務，則需要使用多個目標群組。
- 當目標類型為 時ALB，您可以將單一內部 Application Load Balancer 註冊為最多兩個 VPC Lattice 服務的目標。若要這樣做，請使用兩個不同的目標群組註冊 Application Load Balancer，供兩個不同的 VPC Lattice 服務使用。此外，目標 Application Load Balancer 必須至少有一個接聽程式，其連接埠符合目標群組連接埠。
- 您可以在啟動時自動向 VPC Lattice 目標群組註冊 ECS 任務。目標群組必須擁有 IP 的目標類型。如需詳細資訊，請參閱《Amazon Elastic Container Service 開發人員指南》中的搭配 Amazon ECS 服務使用 VPC Lattice。

  或者，使用類型為 的 VPC Lattice 目標群組註冊 Amazon ECS 服務的 Application Load BalancerALB。如需詳細資訊，請參閱《Amazon Elastic Container Service 開發人員指南》中的使用負載平衡來分配 Amazon ECS 服務流量。
- 若要將 EKS Pod 註冊為目標，請使用AWS 閘道 API 控制器，從 Kubernetes 服務取得 IP 地址。
- 如果目標群組通訊協定是 TCP，則唯一支援的目標類型為 INSTANCE、 IP或 ALB。

IP 地址類型

當您建立目標類型為 的目標群組時IP，您可以指定目標群組的 IP 地址類型。這會指定負載平衡器用來將請求和運作狀態檢查傳送至目標的地址類型。可能的值為 IPv4 和 IPv6。預設值為 IPV4。

考量事項

- 如果您使用 IP 地址類型 建立目標群組IPv6，則您為目標群組指定的 VPC 必須具有 IPv6 地址範圍。
- 您向目標群組註冊的 IP 地址必須符合目標群組的 IP 地址類型。例如，如果 IPv6 地址的 IP 地址類型為 ，則無法向目標群組註冊 IPv6 地址IPv4。
- 您向目標群組註冊的 IP 地址必須在您為目標群組指定的 VPC 的 IP 地址範圍內。

VPC Lattice 中的 HTTP 目標

HTTP 請求和 HTTP 回應使用標頭欄位來傳送有關 HTTP 訊息的資訊。HTTP 標頭會自動新增。標頭欄位是以冒號分隔的名稱值組，以歸位字元 (CR) 和換行 (LF) 分隔。一組以 RFC 2616 定義的標準 HTTP 標頭欄位，訊息標頭。也有應用程式廣泛採用的非標準 HTTP 標頭可用 (而且會自動新增)。例如，有 `x-forwarded`字首為 的非標準 HTTP 標頭。

x-forwarded 標頭

Amazon VPC Lattice 新增了下列`x-forwarded`標頭：

`x-forwarded-for`

來源 IP 地址。

`x-forwarded-port`

目的地連接埠。

`x-forwarded-proto`

連線通訊協定 (`http` | `https`)。

來電者身分標頭

Amazon VPC Lattice 新增了下列呼叫者身分標頭：

`x-amzn-lattice-identity`

身分資訊。如果 AWS 身分驗證成功，則會顯示下列欄位。

- `Principal` – 已驗證的委託人。
- `PrincipalOrgID` – 已驗證主體的組織 ID。
- `PrincipalOrgPath` – 已驗證主體的組織路徑。
- `SessionName` – 已驗證工作階段的名稱。

如果使用 Roles Anywhere 登入資料且身分驗證成功，則會顯示下列欄位。

- `X509Issuer/OU` – 發行者 (OU)。
- `X509SAN/DNS` – 主體替代名稱 (DNS)。
- `X509SAN/NameCN` – 發行者替代名稱 （ 名稱/CN)。

- X509SAN/URI – 主體替代名稱 (URI)。
- X509Subject/CN – 主體名稱 (CN)。

`x-amzn-lattice-identity-tags`

主體 ID 和任何主體標籤。格式如下所示。

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

VPC Lattice 會以反斜線 (\) 逸出值中的任何分號 (;)。

`x-amzn-lattice-network`

VPC。格式如下所示。

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

`x-amzn-lattice-target`

目標。格式如下所示。

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

如需 VPC Lattice 資源 ARNs 的相關資訊，請參閱 Amazon VPC Lattice 定義的資源類型。

來電者身分標頭無法欺騙。VPC Lattice 會從任何傳入請求中去除這些標頭。這些身分標頭表示支援使用下列格式的空白值的映射。剖析時，您不應依賴這些標頭中 KEYs 的特定順序，您應該預期可以隨時新增新的 KEYs，並且應該準備好處理空值。

格式如下所示。

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

Lambda 函數作為 VPC Lattice 中的目標

您可以使用 VPC Lattice 目標群組將 Lambda 函數註冊為目標，並設定接聽程式規則，將請求轉送至 Lambda 函數的目標群組。當服務將請求轉送至具有 Lambda 函數做為目標的目標群組時，它會叫用您的 Lambda 函數，並以 JSON 格式將請求的內容傳遞給 Lambda 函數。

限制

- Lambda 函數和目標群組必須在相同的帳戶中，且在相同的區域內。
- 您可以傳送到 Lambda 函數的請求內文大小上限為 6 MB。
- Lambda 函數可傳送的回應 JSON 大小上限為 6 MB。
- 通訊協定必須是 HTTP 或 HTTPS。

準備 Lambda 函數

如果您使用 Lambda 函數搭配 VPC Lattice 服務，則適用下列建議。

調用 Lambda 函數的許可

當您使用 AWS 管理主控台 或 建立目標群組並註冊 Lambda 函數時 AWS CLI，VPC Lattice 會代表您將必要的許可新增至 Lambda 函數政策。

您也可以使用下列 API 呼叫自行新增許可：

```
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 函數版本控制

您可以為每個目標群組註冊一個 Lambda 函數。為了確保您可以變更 Lambda 函數，且 VPC Lattice 服務一律叫用 Lambda 函數的目前版本，請在向 VPC Lattice 服務註冊 Lambda 函數時，建立函數別名，並將別名包含在函數 ARN 中。如需詳細資訊，請參閱《 AWS Lambda 開發人員指南》中的 Lambda 函數版本和建立 Lambda 函數的別名。

為 Lambda 函數建立目標群組

建立目標群組以用於請求路由。如果請求內容符合接聽程式規則和轉送到此目標群組的動作，VPC Lattice 服務會叫用已註冊的 Lambda 函數。

使用主控台建立目標群組並註冊 Lambda 函數

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。

2. 在導覽窗格的 VPC Lattice 下，選擇目標群組。
3. 選擇 Create target group (建立目標群組)。
4. 在選取目標類型中，選取 Lambda 函數。
5. 針對 Target group name (目標群組名稱)，輸入目標群組的名稱。
6. 針對 Lambda 事件結構版本，選擇版本。如需詳細資訊，請參閱the section called “從 VPC Lattice 服務接收事件”。
7. （選用）若要新增標籤，請展開標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
8. 選擇下一步。
9. 對於 Lambda function (Lambda 函數)，請執行以下其中一項：
   - 選取現有的 Lambda 函數。
   - 建立新的 Lambda 函數並選取它。
   - 稍後註冊 Lambda 函數。
10. 選擇 Create target group (建立目標群組)。

使用 建立目標群組並註冊 Lambda 函數 AWS CLI

使用 create-target-group 和 register-targets 命令。

從 VPC Lattice 服務接收事件

VPC Lattice 服務支援透過 HTTP 和 HTTPS 對請求進行 Lambda 調用。服務會以 JSON 格式傳送事件，並將 `X-Forwarded-For`標頭新增至每個請求。

Base64 編碼

如果`content-encoding`標頭存在且內容類型不是下列其中一項，則服務 Base64 會編碼內文：

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

如果 `content-encoding` 標頭不存在，則 Base64 編碼取決於內容類型。對於上述內容類型，服務會依原樣傳送內文，無需 Base64 編碼。

事件結構格式

當您建立或更新類型 的目標群組時LAMBDA，您可以指定 Lambda 函數接收的事件結構版本。可能的版本為 V1和 V2。

Example範例事件：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`

請求的本文。只有在通訊協定為 HTTP、HTTPS 或 gRPC 時才會顯示。

`headers`

請求的 HTTP 標頭。只有在通訊協定為 HTTP、HTTPS 或 gRPC 時才會顯示。

`identity`

身分資訊。以下是可能的欄位。

- `principal` – 已驗證的委託人。只有在 AWS 身分驗證成功時才會顯示。
- `principalOrgID` – 已驗證主體的組織 ID。只有在 AWS 身分驗證成功時才會顯示。
- `sessionName` – 已驗證工作階段的名稱。只有在 AWS 身分驗證成功時才會顯示。
- `sourceVpcArn` – 發出請求之 VPC 的 ARN。只有在可以識別來源 VPC 時才會顯示。
- `type` – 如果使用身分驗證政策且 AWS 身分驗證成功`AWS_IAM`，則值為 。

如果使用 Roles Anywhere 登入資料且身分驗證成功，下列是可能的欄位。

- `x509IssuerOu` – 發行者 (OU)。
- `x509SanDns` – 主體替代名稱 (DNS)。
- `x509SanNameCn` – 發行者替代名稱 （ 名稱/CN)。
- `x509SanUri` – 主體替代名稱 (URI)。
- `x509SubjectCn` – 主體名稱 (CN)。

`isBase64Encoded`

指出內文是否為 base64 編碼。只有在通訊協定為 HTTP、HTTPS 或 gRPC，且請求內文尚未是字串時，才會顯示。

`method`

請求的 HTTP 方法。只有在通訊協定為 HTTP、HTTPS 或 gRPC 時才會顯示。

`path`

來自用戶端的請求路徑，其中包含查詢字串參數。只有在通訊協定為 HTTP、HTTPS 或 gRPC 時才會顯示。

`queryStringParameters`

HTTP 查詢字串參數。只有在通訊協定為 HTTP、HTTPS 或 gRPC 時才會顯示。

`serviceArn`

接收請求之服務的 ARN。

`serviceNetworkArn`

交付請求之服務網路的 ARN。

`targetGroupArn`

接收請求之目標群組的 ARN。

`timeEpoch`

時間，以微秒為單位。

Example範例事件：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
}
```

回應 VPC Lattice 服務

來自 Lambda 函數的回應必須包含 Base64 編碼狀態、狀態碼、狀態描述和標頭。您可以省略內文。

若要在回應的內文中包含二進位內容，您必須將內容以 Base64 編碼，並將 `isBase64Encoded` 設定為 `true`。服務會解碼內容以擷取二進位內容，並將其傳送至 HTTP 回應內文中的用戶端。

VPC Lattice 服務不遵守hop-by-hop標頭，例如 `Connection`或 `Transfer-Encoding`。您可以省略 `Content-Length`標頭，因為服務會在傳送回應給用戶端之前進行運算。

以下是來自 Lambda 函數的範例回應：

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

多值標頭

VPC Lattice 支援來自用戶端的請求或來自 Lambda 函數的回應，其中包含具有多個值或多次包含相同標頭的標頭。VPC Lattice 會將所有值傳遞至目標。

在下列範例中，有兩個名為 的標頭header1具有不同的值。

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

使用 V2 事件結構時，VPC Lattice 會傳送清單中的值。例如：

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

使用 V1 事件結構時，VPC Lattice 會將值合併為單一字串。例如：

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

多值查詢字串參數

VPC Lattice 支援具有相同索引鍵多個值的查詢參數。

在下列範例中，有兩個名為 的參數QS1具有不同的值。

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

使用 V2 事件結構時，VPC Lattice 會傳送清單中的值。例如：

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

使用 V1 事件結構時，VPC Lattice 會使用傳遞的最後一個值。例如：

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

取消註冊 Lambda 函數

如果您不再需要將流量傳送到您的 Lambda 函數，則可以將它取消註冊。取消註冊 Lambda 函數之後，傳輸中的請求會失敗，出現 HTTP 5XX 錯誤。

若要取代 Lambda 函數，建議您建立新的目標群組、向新目標群組註冊新函數，並更新接聽程式規則以使用新的目標群組，而非現有的目標群組。

使用主控台取消註冊 Lambda 函數

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇目標群組。
3. 選擇目標群組的名稱，以開啟其詳細資訊頁面。
4. 在 Targets (目標) 索引標籤上，選擇 Deregister (取消註冊)。
5. 出現確認提示時，請輸入 **`confirm`**，然後選擇取消註冊。

使用 取消註冊 Lambda 函數 AWS CLI

使用 deregister-targets 命令。

Application Load Balancer 作為 VPC Lattice 中的目標

您可以建立 VPC Lattice 目標群組、將單一內部 Application Load Balancer 註冊為目標，以及設定 VPC Lattice 服務將流量轉送至此目標群組。在此案例中，Application Load Balancer 會在流量到達路由決策時立即接管路由決策。此組態可讓您使用 Application Load Balancer 的第 7 層請求型路由功能，以及 VPC Lattice 支援的功能，例如 IAM 身分驗證和授權，以及跨 VPCs和帳戶的連線。

限制

- 您可以在類型為 的 VPC Lattice 目標群組中，將單一內部 Application Load Balancer 註冊為目標ALB。
- 您可以將 Application Load Balancer 註冊為最多兩個 VPC Lattice 目標群組的目標，供兩個不同的 VPC Lattice 服務使用。

- VPC Lattice 不提供ALB類型目標群組的運作狀態檢查。不過，您可以在負載平衡器層級為 Elastic Load Balancing 中的目標獨立設定運作狀態檢查。如需詳細資訊，請參閱《Application Load Balancer 使用者指南》中的目標群組運作狀態檢查

先決條件

建立 Application Load Balancer，以向 VPC Lattice 目標群組註冊為目標。負載平衡器必須符合下列條件：

- 負載平衡器方案為內部。
- Application Load Balancer 必須位於與 VPC Lattice 目標群組相同的帳戶中，且必須處於作用中狀態。
- Application Load Balancer 必須與 VPC Lattice 目標群組位於相同的 VPC 中。
- 您可以在 Application Load Balancer 上使用 HTTPS 接聽程式來終止 TLS，但前提是 VPC Lattice 服務使用與負載平衡器相同的 SSL/TLS 憑證。
- 若要在`X-Forwarded-For`請求標頭中保留 VPC Lattice 服務的用戶端 IP，您必須將 Application Load Balancer 的 屬性設定為 `routing.http.xff_header_processing.mode` `Preserve`。如果值為 `Preserve`，負載平衡器會在 HTTP 請求中保留 `X-Forwarded-For`標頭，並將其傳送至目標，而不會進行任何變更。

如需詳細資訊，請參閱《Application Load Balancer 使用者指南》中的建立 Application Load Balancer。

步驟 1：建立類型為 ALB 的目標群組

使用下列程序來建立目標群組。請注意，VPC Lattice 不支援ALB目標群組的運作狀態檢查。不過，您可以為 Application Load Balancer 的目標群組設定運作狀態檢查。如需詳細資訊，請參閱《Application Load Balancer 使用者指南》中的目標群組運作狀態檢查。

建立目標群組

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇目標群組。
3. 選擇 Create target group (建立目標群組)。
4. 在指定目標群組詳細資訊頁面的基本組態下，選擇 Application Load Balancer 作為目標類型。
5. 針對 Target group name (目標群組名稱)，輸入目標群組的名稱。

6. 針對通訊協定，選擇 **HTTP**、 **HTTPS**或 **TCP**。目標群組通訊協定必須符合內部 Application Load Balancer 接聽程式的通訊協定。
7. 針對連接埠，指定目標群組的連接埠。此連接埠必須符合內部 Application Load Balancer 的接聽程式連接埠。您也可以在內部 Application Load Balancer 上新增接聽程式連接埠，以符合您在此處指定的目標群組連接埠。
8. 針對 VPC，選取您在建立內部 Application Load Balancer 時選取的相同虛擬私有雲端 (VPC)。這應該是包含 VPC Lattice 資源的 VPC。
9. 針對通訊協定版本，選擇 Application Load Balancer 支援的通訊協定版本。
10. （選用）新增任何必要的標籤。
11. 選擇下一步。

步驟 2：將 Application Load Balancer 註冊為目標

您可以立即或稍後將負載平衡器註冊為目標。

將 Application Load Balancer 註冊為目標

1. 選擇立即註冊。
2. 針對 Application Load Balancer，選擇您的內部 Application Load Balancer。
3. 對於連接埠，請保留預設值或視需要指定不同的連接埠。此連接埠必須符合 Application Load Balancer 上現有的接聽程式連接埠。如果您在沒有相符連接埠的情況下繼續，流量將無法到達 Application Load Balancer。
4. 選擇 Create target group (建立目標群組)。

通訊協定版本

根據預設， 服務會使用 HTTP/1.1 將請求傳送至目標。您可以使用通訊協定版本，使用 HTTP/2 或 gRPC 將請求傳送至目標。

下表摘要說明請求通訊協定與目標群組通訊協定版本組合的結果。

請求通訊協定	通訊協定版本	結果
HTTP/1.1	HTTP/1.1	成功
HTTP/2	HTTP/1.1	成功

請求通訊協定	通訊協定版本	結果
gRPC	HTTP/1.1	錯誤
HTTP/1.1	HTTP/2	錯誤
HTTP/2	HTTP/2	成功
gRPC	HTTP/2	如果目標支援 gRPC，則成功
HTTP/1.1	gRPC	錯誤
HTTP/2	gRPC	如果是 POST 請求，則成功
gRPC	gRPC	成功

gRPC 通訊協定版本的考量事項

- 唯一支援的接聽程式通訊協定是 HTTPS。
- 支援的目標類型僅為 `INSTANCE` 和 `IP`。
- 服務會剖析 gRPC 請求，並根據套件、服務和方法，將 gRPC 呼叫路由至適當的目標群組。
- 您無法使用 Lambda 函數做為目標。

HTTP/2 通訊協定版本的考量事項

- 唯一支援的接聽程式通訊協定是 HTTPS。您可以為目標群組通訊協定選擇 HTTP 或 HTTPS。
- 唯一支援的接聽程式規則是轉送和固定回應。
- 支援的目標類型僅為 `INSTANCE` 和 `IP`。
- 服務支援從用戶端串流。服務不支援串流至目標。

VPC Lattice 目標群組的標籤

標籤可幫助您以不同的方式來將目標群組分類，例如，根據目的、擁有者或環境。

您可以在每個目標群組中加入多個標籤。每個目標群組的標籤索引鍵必須是唯一的。如果所新增的標籤，其索引鍵已經和目標群組具有關聯，則此動作會更新該標籤的值。

當您使用完標籤之後，可以將其移除。

限制

- 每一資源標籤數上限：50
- 索引鍵長度上限：127 個 Unicode 字元
- 數值長度上限：255 個 Unicode 字元
- 標籤鍵與值皆區分大小寫。允許的字元包括可用 UTF-8 表示的英文字母、空格和數字，還有以下特殊字元：+ - = . _ : / @。不可使用結尾或前方空格。
- 請勿在標籤名稱或值中使用 `aws:`字首，因為它保留供 AWS 使用。您不可編輯或刪除具此字首的標籤名稱或值。具此字首的標籤，不算在受資源限制的標籤計數內。

使用主控台來更新目標群組的標籤

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇目標群組。
3. 選取目標群組的名稱，以開啟其詳細資訊頁面。
4. 選擇 Tags (標籤) 索引標籤。
5. 若要新增標籤，請選擇新增標籤，然後輸入標籤索引鍵和標籤值。若要新增另一個標籤，請再次選擇新增標籤。當您完成新增標籤的作業時，請選擇 Save changes (儲存變更)。
6. 若要刪除標籤，請選取標籤的核取方塊，然後選擇刪除。出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 更新目標群組的標籤 AWS CLI

使用 tag-resource 和 untag-resource 命令。

刪除 VPC Lattice 目標群組

如果沒有任何接聽程式規則的轉送動作參照某目標群組，即可刪除該目標群組。刪除目標群組不會影響透過該目標群組登錄的目標。如果不再需要註冊的 EC2 執行個體，則可以停止或終止它。

使用主控台來刪除目標群組

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中，選擇目標群組。

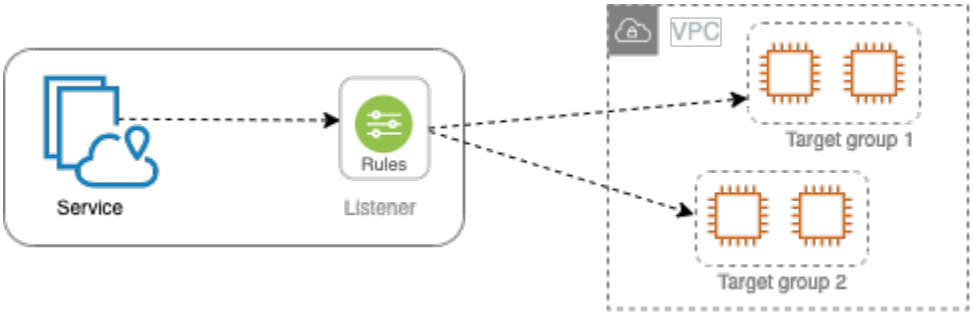
3. 選取目標群組的核取方塊，然後選擇動作、刪除。
4. 出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 刪除目標群組 AWS CLI

使用 delete-target-group 指令。

VPC Lattice 服務的接聽程式

開始使用 VPC Lattice 服務之前，您必須新增接聽程式。接聽程式是檢查連線請求的程序，必須使用您已設定的通訊協定與連接埠。您為接聽程式定義的規則會決定服務如何將請求路由到其已註冊的目標。



目錄

接聽程式組態

接聽程式支援下列通訊協定與連接埠：

- 通訊協定：HTTP、HTTPS、TLS
- Ports (連接埠)：1-65535

如果接聽程式通訊協定是 HTTPS，VPC Lattice 會佈建和管理與 VPC Lattice 產生的 FQDN 相關聯的 TLS 憑證。VPC Lattice 支援 HTTP/1.1 和 HTTP/2 上的 TLS。當您使用 HTTPS 接聽程式設定服務時，VPC Lattice 會使用應用程式層通訊協定交涉 (ALPN) 自動判斷 HTTP 通訊協定。如果沒有 ALPN，VPC Lattice 會預設為 HTTP/1.1。如需詳細資訊，請參閱HTTPS 接聽程式。

VPC Lattice 可以在 HTTP、HTTPS、HTTP/1.1 和 HTTP/2 上接聽，並與這些通訊協定和版本中的任何目標通訊。我們不需要接聽程式和目標群組通訊協定相符。VPC Lattice 會管理通訊協定和版本之間升級和降級的整個程序。如需詳細資訊，請參閱通訊協定版本。

您可以建立 TLS 接聽程式，以確保您的應用程式解密加密的流量，而不是 VPC Lattice。如需詳細資訊，請參閱TLS 接聽程式。

VPC Lattice 原生不支援 WebSockets 不過，您仍然可以使用 TLS 接聽程式或透過 VPC Lattice 資源路由來連線至 Websocket 型服務。

VPC Lattice 服務的 HTTP 接聽程式

接聽程式是檢查連線請求的程序。您可以在建立 VPC Lattice 服務時定義接聽程式。您可以隨時將接聽程式新增至您的服務。

此頁面上的資訊可協助您為服務建立 HTTP 接聽程式。如需建立使用其他通訊協定的接聽程式的相關資訊，請參閱 HTTPS 接聽程式和 TLS 接聽程式。

先決條件

- 若要將轉送動作新增至預設接聽程式規則，您必須指定可用的 VPC Lattice 目標群組。如需詳細資訊，請參閱建立 VPC Lattice 目標群組。
- 您可以在多個接聽程式中指定相同的目標群組，但這些接聽程式必須屬於相同的服務。若要搭配 VPC Lattice 服務使用目標群組，您必須確認接聽程式未將其用於任何其他 VPC Lattice 服務。

新增 HTTP 接聽程式

您可以隨時將接聽程式和規則新增至您的服務。您可以使用通訊協定和從用戶端到 服務的連線連接埠，以及預設接聽程式規則的 VPC Lattice 目標群組來設定接聽程式。如需詳細資訊，請參閱接聽程式組態。

使用主控台新增 HTTP 接聽程式

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 選取服務的名稱以開啟其詳細資訊頁面。
4. 在路由索引標籤上，選擇新增接聽程式。
5. 對於接聽程式名稱，您可以提供自訂接聽程式名稱，或使用接聽程式的通訊協定和連接埠做為接聽程式名稱。您指定的自訂名稱最多可有 63 個字元，且您帳戶中的每個服務都必須是唯一的。有效字元為 a-z、0-9 和連字號 (-)。您不能使用連字號做為第一個或最後一個字元，或緊接在另一個連字號之後。您無法在建立之後變更名稱。

6. 針對通訊協定 ：連接埠，選擇 HTTP 並輸入連接埠號碼。
7. 針對預設動作，選擇 VPC Lattice 目標群組以接收流量，然後選擇要指派給此目標群組的權重。您指派給目標群組的權重會將優先順序設定為接收流量。例如，如果兩個目標群組具有相同的權重，則每個目標群組都會接收一半的流量。如果您只指定一個目標群組，則 100% 的流量會傳送到一個目標群組。

   您可以選擇性地為預設動作新增另一個目標群組。選擇新增動作，然後選擇目標群組並指定其權重。
8. （選用）若要新增另一個規則，請選擇新增規則，然後輸入規則的名稱、優先順序、條件和動作。

   您可以為每個規則提供介於 1 到 100 之間的優先順序數字。接聽程式不能擁有多個優先順序相同的規則。依優先順序評估規則，從最低值到最高值。預設規則最後評估。如需詳細資訊，請參閱接聽程式規則。
9. （選用）若要新增標籤，請展開接聽程式標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
10. 檢閱您的組態，然後選擇新增。

使用 新增 HTTP 接聽程式 AWS CLI

使用 create-listener 命令來建立具有預設規則的接聽程式，並使用 create-rule 命令來建立其他接聽程式規則。

VPC Lattice 服務的 HTTPS 接聽程式

接聽程式是檢查連線請求的程序。您可以在建立服務時定義接聽程式。您可以隨時在 VPC Lattice 中將接聽程式新增至服務。

您可以建立 HTTPS 接聽程式，該接聽程式使用 TLS 1.2 版或 TLS 1.3 版直接終止與 VPC Lattice 的 HTTPS 連線。VPC Lattice 將佈建和管理與 VPC Lattice 產生的完整網域名稱 (FQDN) 相關聯的 TLS 憑證。VPC Lattice 支援 HTTP/1.1 和 HTTP/2 上的 TLS。當您使用 HTTPS 接聽程式設定服務時，VPC Lattice 會透過應用程式層通訊協定交涉 (ALPN) 自動判斷 HTTP 通訊協定。如果沒有 ALPN，VPC Lattice 會預設為 HTTP/1.1。

VPC Lattice 使用多租用戶架構，這表示它可以在相同的端點上託管多個服務。VPC Lattice 會針對每個用戶端請求使用 TLS 搭配伺服器名稱指示 (SNI)。不支援加密的用戶端 Hello (ECH) 和加密的伺服器名稱指示 (ESNI)。

VPC Lattice 可以在 HTTP、HTTPS、HTTP/1.1 和 HTTP/2 上接聽，並與這些通訊協定和版本中的任何目標通訊。這些接聽程式和目標群組組態不需要相符。VPC Lattice 會管理通訊協定和版本之間升級和降級的整個程序。如需詳細資訊，請參閱通訊協定版本。

為了確保您的應用程式解密流量，請改為建立 TLS 接聽程式。透過 TLS 傳遞，VPC Lattice 不會終止 TLS。如需詳細資訊，請參閱TLS 接聽程式。

內容

安全政策

VPC Lattice 使用安全政策，該政策是 TLSv1.2 通訊協定和 SSL/TLS 密碼清單的組合。通訊協定會在用戶端與伺服器之間建立安全連線，並協助確保用戶端與 VPC Lattice 中的服務之間傳遞的所有資料皆為私有。密碼是一種加密演算法，使用加密金鑰來建立編碼的訊息。通訊協定使用多個密碼來加密資料。在連線交涉程序期間，用戶端和 VPC Lattice 會依偏好順序，提供他們各自支援的加密和通訊協定清單。在預設情況下，將針對安全連線選取伺服器清單上符合任何用戶端加密的第一個加密。

VPC Lattice 依此偏好順序使用下列 TLS 1.2 SSL/TLS 密碼：

- `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 也會依此偏好順序使用下列 TLS 1.3 SSL/TLS 密碼：

- `TLS_AES_128_GCM_SHA256`
- `TLS_AES_256_GCM_SHA384`

- TLS_CHACHA20_POLY1305_SHA256

ALPN 政策

應用程式層通訊協定交涉 (ALPN) 是在初始 TLS 交握 hello 訊息上傳送的 TLS 延伸。ALPN 使應用程式層能夠協商哪些通訊協定的使用透過安全的連接 (如 HTTP/1 和 HTTP/2) 來進行。

當用戶端啟動 ALPN 連線時，VPC Lattice 服務會將用戶端 ALPN 偏好設定清單與其 ALPN 政策進行比較。如果用戶端支援來自 ALPN 政策的通訊協定，VPC Lattice 服務會根據 ALPN 政策的偏好設定清單建立連線。否則，服務不會使用 ALPN。

VPC Lattice 支援下列 ALPN 政策：

`HTTP2Preferred`

偏好 HTTP/2 over HTTP/1.1。ALPN 偏好設定清單為 h2、http/1.1。

新增 HTTPS 接聽程式

您可以使用通訊協定和連接埠來設定接聽程式，以便從用戶端連線至 服務，並為預設接聽程式規則設定目標群組。如需詳細資訊，請參閱接聽程式組態。

先決條件

- 若要將轉送動作新增至預設接聽程式規則，您必須指定可用的 VPC Lattice 目標群組。如需詳細資訊，請參閱建立 VPC Lattice 目標群組。
- 您可以在多個接聽程式中指定相同的目標群組，但這些接聽程式必須屬於相同的 VPC Lattice 服務。若要搭配 VPC Lattice 服務使用目標群組，您必須確認接聽程式未將其用於任何其他 VPC Lattice 服務。
- 您可以使用 VPC Lattice 提供的憑證，或將您自己的憑證匯入至其中 AWS Certificate Manager。如需詳細資訊，請參閱the section called “BYOC”。

使用主控台新增 HTTPS 接聽程式

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 選取服務的名稱以開啟其詳細資訊頁面。
4. 在路由索引標籤上，選擇新增接聽程式。

5. 對於接聽程式名稱，您可以提供自訂接聽程式名稱，或使用接聽程式的通訊協定和連接埠做為接聽程式名稱。您指定的自訂名稱最多可有 63 個字元，且您帳戶中的每個服務都必須是唯一的。有效字元為 a-z、0-9 和連字號 (-)。您不能使用連字號做為第一個或最後一個字元，或緊接在另一個連字號之後。您無法在建立接聽程式之後變更接聽程式的名稱。
6. 針對通訊協定：連接埠，選擇 HTTPS，然後輸入連接埠號碼。
7. 針對預設動作，選擇 VPC Lattice 目標群組以接收流量，然後選擇要指派給此目標群組的權重。您指派給目標群組的權重會將優先順序設定為接收流量。例如，如果兩個目標群組具有相同的權重，則每個目標群組都會接收一半的流量。如果您只指定一個目標群組，則 100% 的流量會傳送到一個目標群組。

   您可以選擇性地為預設動作新增另一個目標群組。選擇新增動作，然後選擇目標群組並指定其權重。
8. （選用）若要新增另一個規則，請選擇新增規則，然後輸入規則的名稱、優先順序、條件和動作。

   您可以為每個規則提供介於 1 到 100 之間的優先順序數字。接聽程式不能擁有多個優先順序相同的規則。依優先順序評估規則，從最低值到最高值。預設規則最後評估。如需詳細資訊，請參閱接聽程式規則。
9. （選用）若要新增標籤，請展開接聽程式標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
10. 對於 HTTPS 接聽程式憑證設定，如果您未在建立服務時指定自訂網域名稱，VPC Lattice 會自動產生 TLS 憑證，以保護流經接聽程式的流量。

    如果您使用自訂網域名稱建立服務，但未指定相符的憑證，則現在可以從自訂 SSL/TLS 憑證中選擇憑證來執行此操作。否則，您在建立服務時指定的憑證已選擇。
11. 檢閱您的組態，然後選擇新增。

使用 新增 HTTPS 接聽程式 AWS CLI

使用 create-listener 命令來建立具有預設規則的接聽程式，並使用 create-rule 命令來建立其他接聽程式規則。

VPC Lattice 服務的 TLS 接聽程式

接聽程式是檢查連線請求的程序。您可以在建立 VPC Lattice 服務時定義接聽程式。您可以隨時將接聽程式新增至您的服務。

您可以建立 TLS 接聽程式，讓 VPC Lattice 將加密的流量傳遞至您的應用程式，而不會解密。

如果您希望 VPC Lattice 解密加密的流量，並將未加密的流量傳送到您的應用程式，請改為建立 HTTPS 接聽程式。如需詳細資訊，請參閱HTTPS 接聽程式。

考量事項

下列考量適用於 TLS 接聽程式：

- VPC Lattice 服務必須具有自訂網域名稱。服務自訂網域名稱會用作服務名稱指示 (SNI) 比對。如果您在建立服務時指定憑證，則不會使用該憑證。
- TLS 接聽程式允許的唯一規則是預設規則。
- TLS 接聽程式的預設動作必須是轉送動作至 TCP 目標群組。
- 預設會停用 TCP 目標群組的運作狀態檢查。如果您啟用 TCP 目標群組的運作狀態檢查，則必須指定通訊協定和通訊協定版本。
- TLS 接聽程式會使用 client-hello 訊息的 SNI 欄位路由請求。如果相符條件與 client-hello 完全相符，您可以在目標上使用萬用字元和 SAN 憑證。
- 由於從用戶端到目標的所有流量都會保持加密，VPC Lattice 無法讀取 HTTP 標頭，也無法插入或移除 HTTP 標頭。因此，使用 TLS 接聽程式時，存在下列限制：
  - 連線持續時間限制為 10 分鐘
  - 驗證政策僅限於匿名主體
  - 不支援 Lambda 目標
- Websocket 連線可以使用 TLS 接聽程式連線至 、VPC Lattice 服務。存在下列限制：
  - 連線持續時間限制為 10 分鐘
  - 驗證政策僅限於匿名主體
  - 不支援 Lambda 目標
- 不支援加密的用戶端 Hello (ECH)。
- 不支援加密的伺服器名稱指示 (ESNI)。

新增 TLS 接聽程式

您可以使用通訊協定和連接埠來設定接聽程式，以便從用戶端連線至 服務，並為預設接聽程式規則設定目標群組。如需詳細資訊，請參閱接聽程式組態。

使用主控台新增 TLS 接聽程式

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。

2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 選取服務的名稱以開啟其詳細資訊頁面。
4. 在路由索引標籤上，選擇新增接聽程式。
5. 對於接聽程式名稱，您可以提供自訂接聽程式名稱，或使用接聽程式的通訊協定和連接埠做為接聽程式名稱。您指定的自訂名稱最多可有 63 個字元，且您帳戶中的每個服務都必須是唯一的。有效字元為 a-z、0-9 和連字號 (-)。您不能使用連字號做為第一個或最後一個字元，或緊接在另一個連字號之後。您無法在建立接聽程式之後變更接聽程式的名稱。
6. 針對通訊協定，選擇 TLS。在連接埠中，輸入連接埠號碼。
7. 針對轉送至目標群組，選擇使用 TCP 通訊協定接收流量的 VPC Lattice 目標群組，然後選擇要指派給此目標群組的權重。您可以選擇性地新增另一個目標群組。選擇新增目標群組，然後選擇目標群組，然後輸入其權重。
8. （選用）若要新增標籤，請展開接聽程式標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
9. 檢閱您的組態，然後選擇新增。

使用 新增 TLS 接聽程式 AWS CLI

使用 create-listener 命令建立具有預設規則的接聽程式。指定 TLS_PASSTHROUGH 通訊協定。

VPC Lattice 服務的接聽程式規則

每個接聽程式都有您可以定義的預設規則和其他規則。每個規則由優先順序、一或多個動作及一或多個條件組成。您可以隨時新增或編輯規則。

目錄

預設規則

建立接聽程式時，您會定義預設規則的預設動作。預設規則不能有條件。如果沒有符合任何接聽程式規則的條件，則會執行預設規則的動作。

規則優先順序

每個規則具有優先順序。依優先順序評估規則，從最低值到最高值。預設規則最後評估。您可以隨時變更非預設規則的優先順序。您無法變更預設規則的優先順序。

規則動作

VPC Lattice 服務的接聽程式支援轉送動作和固定回應動作。

轉送動作

您可以使用forward動作將請求路由到一或多個 VPC Lattice 目標群組。如果您為一個 forward 動作指定多個目標群組，則必須為每個目標群組指定加權。每個目標群組權重為介於 0 到 999 之間的值。符合加權目標群組之監聽程式規則的請求，會根據其權重分配到這些目標群組。例如，如果您指定兩個目標群組，每個目標群組的權重為 10，則每個目標群組都會收到一半的請求。如果您指定兩個目標群組，一個權重為 10，另一個權重為 20，則權重為 20 的目標群組接收的請求數量是另一個目標群組的兩倍。

固定回應動作

您可以使用 fixed-response 動作來捨棄用戶端請求，並傳回自訂 HTTP 回應。您可以使用此動作傳回 404 或 500 回應碼。

Example的固定回應動作範例 AWS CLI

您可以在建立或更新規則時指定 動作。下列動作會傳送具有指定狀態碼的固定回應。

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

規則條件

每個規則條件具有類型和組態資訊。滿足規則的條件時，即會執行它的動作。

以下是規則支援的相符條件：

標頭比對

路由是以每個請求的 HTTP 標頭為基礎。您可以使用 HTTP 標頭條件來設定規則，以根據請求的 HTTP 標頭來路由傳送請求。您可以指定標準或自訂 HTTP 標頭欄位的名稱。標頭名稱和相符項目評估不區分大小寫。您可以開啟區分大小寫功能來變更此設定。標頭名稱中不支援萬用字元。標頭比對支援字首、精確且包含比對。

方法比對

路由是以每個請求的 HTTP 請求方法為基礎。

您可以使用 HTTP 請求方法條件來設定規則，以根據請求的 HTTP 請求方法來路由傳送請求。您可以指定標準或自訂 HTTP 方法。方法比對區分大小寫。方法名稱必須完全相符。不支援萬用字元。

路徑比對

路由是以符合請求 URLs中的路徑模式為基礎。

您可以使用路徑條件來定義規則，以根據請求中的 URL 路由請求。不支援萬用字元。支援路徑上的字首和完全相符。

新增規則

您可以隨時新增接聽程式規則。

使用主控台新增接聽程式規則

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務。
3. 選取服務的名稱以開啟其詳細資訊頁面。
4. 在路由索引標籤上，選擇編輯接聽程式。
5. 展開接聽程式規則，然後選擇新增規則。
6. 針對 Rule name (規則名稱)，輸入規則的名稱。
7. 針對優先順序，輸入介於 1 到 100 之間的優先順序。依優先順序評估規則，從最低值到最高值。預設規則最後評估。
8. 針對條件，輸入路徑比對條件的路徑模式。每個字串的大小上限為 200 個字元。比較不區分大小寫。不支援萬用字元。

若要新增標頭比對或方法比對規則條件，請使用 AWS CLI 或 AWS SDK。

9. 針對動作，選擇 VPC Lattice 目標群組。
10. 選擇儲存變更。

使用 新增規則 AWS CLI

使用 create-rule 命令。

更新規則

您可以隨時更新接聽程式規則。您可以修改其優先順序、條件、目標群組和每個目標群組的權重。您無法修改規則的名稱。

使用主控台更新接聽程式規則

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 選取服務的名稱以開啟其詳細資訊頁面。
4. 在路由索引標籤上，選擇編輯接聽程式。
5. 視需要修改規則優先順序、條件和動作。
6. 檢閱您的更新，然後選擇儲存變更。

使用 更新規則 AWS CLI

使用 update-rule 命令。

刪除規則

您可以隨時刪除接聽程式的非預設規則。您無法刪除接聽程式的預設規則。當您刪除接聽程式時，會刪除其所有規則。

使用主控台刪除接聽程式規則

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 選取服務的名稱以開啟其詳細資訊頁面。

4. 在路由索引標籤上，選擇編輯接聽程式。
5. 尋找規則，然後選擇移除。
6. 選擇儲存變更。

使用 刪除規則 AWS CLI

使用 delete-rule 命令。

刪除 VPC Lattice 服務的接聽程式

您可隨時刪除接聽程式。當您刪除接聽程式時，其所有規則都會自動刪除。

使用主控台刪除接聽程式

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格中的 VPC Lattice 下，選擇服務。
3. 選取服務的名稱以開啟其詳細資訊頁面。
4. 在路由索引標籤上，選擇刪除接聽程式。
5. 出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 刪除接聽程式 AWS CLI

使用 delete-listener 命令。

Amazon VPC Lattice 中的 VPC 資源

您可以與組織中的其他團隊或外部獨立軟體廠商 (ISV) 合作夥伴共用 VPC 資源。VPC 資源可以是 AWS原生資源，例如 Amazon RDS 資料庫、網域名稱或 IP 地址。資源可以位於您的 VPC 或內部部署網路中，不需要負載平衡。您可以使用 AWS RAM 來指定可存取資源的委託人。您可以建立資源閘道，以透過該閘道存取您的資源。您也可以建立資源組態，代表您要共用的資源或資源群組。

您共用資源的主體可以使用 VPC 端點私下存取這些資源。他們可以使用資源 VPC 端點存取 VPC Lattice 服務網路中的一個資源或集區多個資源，並使用服務網路 VPC 端點存取服務網路。

下列各節說明如何在 VPC Lattice 中建立和管理 VPC 資源：

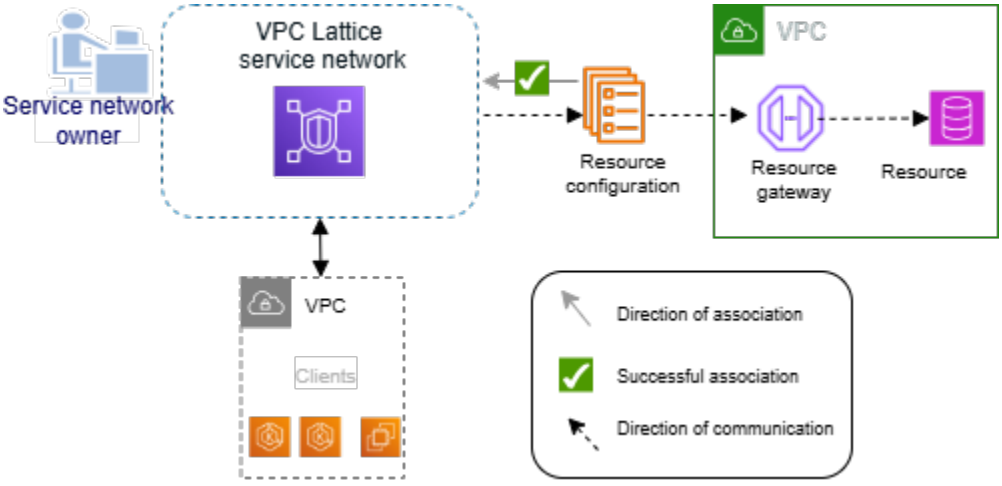
主題

VPC Lattice 中的資源閘道

資源閘道是將流量接收到資源所在 VPC 的點。它跨越多個可用區域。

如果您打算讓 VPC 內的資源可從其他 VPCs或帳戶存取，VPC 必須具有資源閘道。您共用的每個資源都與資源閘道相關聯。當其他 VPCs或帳戶中的用戶端存取 VPC 中的資源時，資源會看到來自該 VPC 中資源閘道的本機流量。流量的來源 IP 地址是可用區域中資源閘道的 IP 地址。多個資源組態，每個都具有多個資源，可以連接到資源閘道。

下圖顯示用戶端如何透過資源閘道存取資源：



目錄

考量事項

下列考量適用於資源閘道：

- 若要從所有可用區域存取您的資源，您應該建立資源閘道，以盡可能跨越多個可用區域。
- VPC 端點和資源閘道的至少一個可用區域必須重疊。
- VPC 最多可以有 100 個資源閘道。如需詳細資訊，請參閱 VPC Lattice 的配額。
- VPC Lattice 可能會將新的 ENIs 新增至您的資源閘道。
- 具有共用 VPC 子網路的資源閘道：
  - 資源閘道只能由擁有 VPC 的帳戶部署到共用 VPC 子網路。
  - 資源閘道的資源組態只能由擁有資源閘道的帳戶建立。

Security groups (安全群組)

您可以將安全群組連接到資源閘道。資源閘道的安全群組規則會控制從資源閘道到資源的傳出流量。

從資源閘道流向資料庫資源之流量的建議傳出規則

若要讓流量從資源閘道流向資源，您必須為資源接受的接聽程式通訊協定和連接埠範圍建立傳出規則。

目標	通訊協定	連接埠範圍	Comment
### CIDR ##	TCP	3306	允許從資源閘道到資料庫的流量。

IP 地址類型

資源閘道可以有 IPv4、IPv6 或雙堆疊地址。資源閘道的 IP 地址類型必須與資源閘道的子網路和資源的 IP 地址類型相容，如下所述：

- IPv4 – 將 IPv4 地址指派給資源閘道網路介面。只有在所有選取的子網路都具有 IPv4 地址範圍，且資源也具有 IPv4 地址時，才支援此選項。使用此選項時，您可以設定每個資源閘道 ENI 的 IPv4 地址數量。
- IPv6 – 將 IPv6 地址指派給資源閘道網路介面。只有在所有選取的子網路都是僅限 IPv6 的子網路，且資源也具有 IPv6 地址時，才支援此選項。當您使用此選項時，IPv6 地址會自動指派，不需要管理。
- Dualstack – 將 IPv4 和 IPv6 地址指派給資源閘道網路介面。只有在所有選取的子網路同時具有 IPv4 和 IPv6 地址範圍，且資源具有 IPv4 或 IPv6 地址時，才支援此選項。使用此選項時，您可以設定每個資源閘道 ENI 的 IPv4 地址數量。

資源閘道的 IP 地址類型與用戶端的 IP 地址類型或存取資源的 VPC 端點無關。

每個 ENI 的 IPv4 位址

如果您的資源閘道具有 IPv4 或雙堆疊 IP 地址類型，您可以設定指派給資源閘道每個 ENI 的 IPv4 地址數目。建立資源閘道時，您可以選擇 1 到 62 個 IPv4 地址。設定 IPv4 地址的數量後，就無法變更該值。

IPv4 地址用於網路地址轉譯，並判斷與資源並行 IPv4 連線的數量上限。每個 IPv4 地址每個目的地 IP 最多可支援 55，000 個同時連線。根據預設，所有資源閘道都會為每個 ENI 指派 16 個 IPv4 地址。

如果您的資源閘道使用 IPv6 地址類型，資源閘道會自動接收每個 ENI 的 /80 CIDR。此值無法變更。每個連線的最大傳輸單位 (MTU) 為 8500 位元組。

資源組態 DNS 解析

您可以指定資源閘道如何針對做為網域名稱目標的資源組態進行 DNS 解析。這個屬性是不可改變的。您可以選擇：

- PUBLIC （預設） - 使用公有 DNS 解析程式解析網域名稱。
- IN_VPC - 網域名稱會使用資源閘道所在之 VPC 的 DHCP 選項集中設定的 DNS 伺服器來解析。如果您使用的是私有 DNS 伺服器，或您的網域名稱目標位於 Route53 私有託管區域中，您應該使用此項目。

如果 DNS 解析為 IN_VPC，則無法將 ARN 定義的資源組態連接到資源閘道。如果資源閘道使用IPv6-only 的子網路，則無法將 DNS 解析設定為 IN_VPC。

在 VPC Lattice 中建立資源閘道

使用 主控台建立資源閘道。

使用主控台建立資源閘道

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 PrivateLink 和 Lattice 下，選擇資源閘道。
3. 選擇建立資源閘道。
4. 針對資源閘道名稱，輸入您 AWS 帳戶中唯一的名稱。
5. 針對 IP 地址類型，選擇資源閘道的 IP 地址類型。
   - 如果您為 IP 地址類型選取 IPv4 或 Dualstack，則可以為資源閘道輸入每個 ENI 的 IPv4 地址數量。

     預設為每個 ENI 16 個 IPv4 地址。這是與您的後端資源建立連線的適當 IPs 數量。
6. 對於 VPC，請選擇要在其中建立資源閘道的 VPC 和子網路。
7. 針對安全群組，選擇最多五個安全群組，以控制從 VPC 到服務網路的傳入流量。
8. 針對資源組態 DNS 解析，選擇您要如何解析網域名稱目標的 DNS。
   - 如果您使用私有 DNS 伺服器或網域名稱目標位於 Route53 私有託管區域中，請將 設定為 IN_VPC
9. (選用) 若要新增標籤，請選擇 Add new tag (新增標籤)，然後輸入標籤的鍵和值。
10. 選擇建立資源閘道。

使用 建立資源閘道 AWS CLI

使用 create-resource-gateway 命令。

刪除 VPC Lattice 中的資源閘道

使用 主控台刪除資源閘道。

使用主控台刪除資源閘道

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 PrivateLink 和 Lattice 下，選擇資源閘道。
3. 選取您要刪除之資源閘道的核取方塊，然後選擇動作、刪除。出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 刪除資源閘道 AWS CLI

使用 delete-resource-gateway 命令。

VPC 資源的資源組態

資源組態代表您要讓其他 VPCs和帳戶中的用戶端存取的資源或資源群組。透過定義資源組態，您可以從其他 VPC 和帳戶中的用戶端允許私有、安全、單向網路連線至 VPCs中的資源。資源組態與其接收流量的資源閘道相關聯。若要從另一個 VPC 存取資源，它需要有資源組態。

目錄

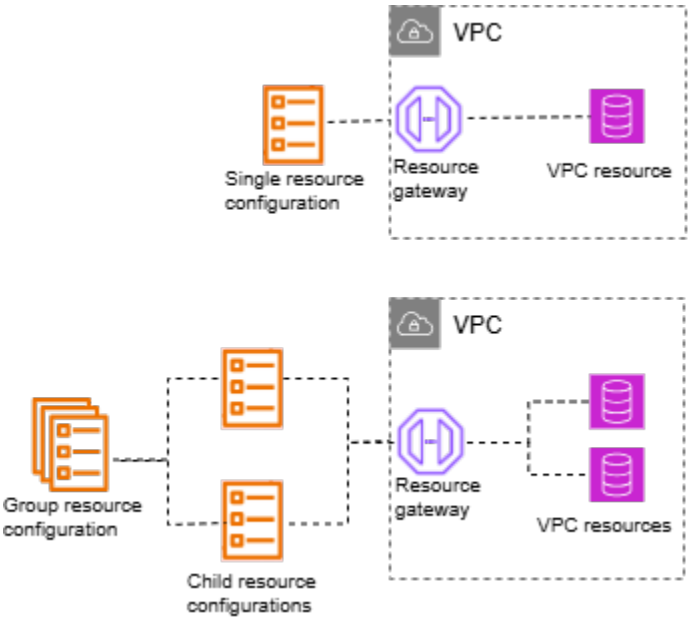
- 管理 VPC Lattice 資源組態的關聯

資源組態的類型

資源組態可以有數種類型。不同類型的協助代表不同類型的資源。類型為：

- 單一資源組態：代表 IP 地址或網域名稱。它可以獨立共用。
- 群組資源組態：它是子資源組態的集合。它可以用來代表 DNS 和 IP 地址端點的群組。
- 子資源組態：它是群組資源組態的成員。它代表 IP 地址或網域名稱。它無法獨立共用，只能做為群組的一部分共用。它可以從群組中新增和移除。新增時，其會自動供可存取 群組的人員存取。
- ARN 資源組態：代表由 AWS 服務佈建的支援資源類型。任何群組-子關係都會自動處理。

下圖顯示單一、子和群組資源組態：



通訊協定

建立資源組態時，您可以定義資源將支援的通訊協定。目前僅支援 TCP 通訊協定。

資源閘道

資源組態與資源閘道相關聯。資源閘道是一組 ENIs，可做為資源所在的 VPC 傳入點。多個資源組態可以與相同的資源閘道相關聯。當其他 VPCs或帳戶中的用戶端存取 VPC 中的資源時，資源會看到來自該 VPC 中資源閘道 IP 地址的本機流量。

資源提供者的自訂網域名稱

資源提供者可以將自訂網域名稱連接到資源組態，例如 `example.com`，取用者可以使用該資源來存取資源組態。自訂網域名稱可以由資源提供者擁有和驗證，也可以是第三方或 AWS 網域。資源提供者可以使用資源組態來共用快取叢集和 Kafka 叢集、TLS 型應用程式或其他 AWS 資源。

下列考量適用於資源組態提供者：

- 資源組態只能有一個自訂網域。
- 資源組態的自訂網域名稱無法變更。
- 所有資源組態取用者都可看見自訂網域名稱。
- 您可以使用 VPC Lattice 中的網域名稱驗證程序來驗證自訂網域名稱。如需詳細資訊，請參閱 the section called “建立和驗證網域”。
- 對於類型群組和子項的資源組態，您必須先在群組資源組態上指定群組網域。之後，子資源組態可以具有屬於群組網域子網域的自訂網域。如果群組沒有群組網域，您可以為子系使用任何自訂網域名稱，但 VPC Lattice 不會為資源取用者 VPC 中的子系網域名稱佈建任何託管區域。

資源取用者的自訂網域名稱

當資源取用者啟用具有自訂網域名稱的資源組態連線時，他們可以允許 VPC Lattice 在其 VPC 中管理 Route 53 私有託管區域。資源取用者有精細的選項，他們想要允許 VPC Lattice 管理私有託管區域的網域。

當透過資源端點、服務網路端點或服務網路 VPC 關聯啟用與資源組態的連線時，資源取用者可以設定 `private-dns-enabled` 參數。除了 `private-dns-enabled` 參數之外，消費者還可以使用 DNS 選項來指定他們希望 VPC Lattice 管理私有託管區域的網域。消費者可以選擇下列私有 DNS 偏好設定：

`ALL_DOMAINS`

VPC Lattice 為所有自訂網域名稱佈建私有託管區域。

`VERIFIED_DOMAINS_ONLY`

只有在供應商已驗證自訂網域名稱時，VPC Lattice 才會佈建私有託管區域。

`VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS`

VPC Lattice 會為所有已驗證的自訂網域名稱和資源取用者指定的其他網域名稱佈建私有託管區域。資源取用者會在 `private DNS specified domains` 參數中指定網域名稱。

SPECIFIED_DOMAINS_ONLY

VPC Lattice 會為資源取用者指定的網域名稱佈建私有託管區域。資源取用者會在 `private DNS specified domains` 參數中指定網域名稱。

當您啟用私有 DNS 時，VPC Lattice 會在 VPC 中為與資源組態相關聯的自訂網域名稱建立私有託管區域。根據預設，私有 DNS 偏好設定會設為 VERIFIED_DOMAINS_ONLY。這表示只有在資源提供者已驗證自訂網域名稱時，才會建立私有託管區域。如果您將私有 DNS 偏好設定設定為 ALL_DOMAINS或 SPECIFIED_DOMAINS_ONLY，則無論自訂網域名稱的驗證狀態為何，VPC Lattice 都會建立私有託管區域。為指定網域建立私有託管區域時，從 VPC 到該網域的所有流量都會透過 VPC Lattice 路由。建議您只在希望這些自訂網域名稱的流量通過 VPC Lattice ALL_DOMAINS時使用 VERIFIED_DOMAINS_AND_SPECIFIED_DOMAINS、 或 SPECIFIED_DOMAINS_ONLY偏好設定。

我們建議資源消費者將其私有 DNS 偏好設定設為 VERIFIED_DOMAINS_ONLY。這可讓消費者透過僅允許 VPC Lattice 為資源消費者帳戶中已驗證的網域佈建私有託管區域，來加強其安全周邊。

若要選取私有 DNS 指定網域中的網域，資源取用者可以輸入完整網域名稱，例如 `my.example.com`，或使用萬用字元，例如 `*.example.com`。

下列考量適用於資源組態的取用者：

- 私有 DNS 啟用參數無法變更。
- 應在服務網路資源關聯上啟用私有 DNS，以便在 VPC 中建立私有託管。對於資源組態，服務網路資源關聯的私有 DNS 啟用狀態會覆寫服務網路端點或服務網路 VPC 關聯的私有 DNS 啟用狀態。

對於屬於網域名稱目標的資源組態，如果符合下列條件，則不會建立私有託管區域項目：

- 資源閘道與服務網路 VPC 端點/服務網路 VPC 關聯位於相同的 VPC 中。
- DNS 解析在資源閘道上設定為 IN_VPC。
- 自訂網域名稱或群組網域是網域名稱目標的相同或更高層級的網域。

服務網路擁有者的自訂網域名稱

服務網路資源關聯的私有 DNS 啟用屬性會覆寫服務網路端點的私有 DNS 啟用屬性和服務網路 VPC 關聯。

如果服務網路擁有者建立服務網路資源關聯，但未啟用私有 DNS，即使已在服務網路端點或服務網路 VPC 關聯上啟用私有 DNS，VPC Lattice 也不會在服務網路連線的任何 VPCs 中為該資源組態佈建私有託管區域。

對於 ARN 類型的資源組態，私有 DNS 旗標為 true 且不變。

資源定義

在資源組態中，以下列其中一種方式識別資源：

- 透過 Amazon Resource Name (ARN)：由 AWS 服務佈建的支援資源類型可由其 ARN 識別。僅支援 Amazon RDS 資料庫。您無法為可公開存取的叢集建立資源組態。
- 依網域名稱目標：您可以使用任何網域名稱。如果您使用私有 DNS 伺服器或網域位於 Route53 私有託管區域，則資源閘道必須將 DNS 解析設定為 IN_VPC。如果您的網域名稱指向 VPC 外部的 IP，則必須在 VPC 中具有 NAT 閘道。
- 依 IP 地址：針對 IPv4，從下列範圍指定私有 IP：10.0.0.0/8、100.64.0.0/10、172.16.0.0/12、192.168.0.0/16。針對 IPv6，從 VPC 指定 IP。不支援公IPs。

連接埠範圍

當您建立資源組態時，您可以定義其將接受請求的連接埠。不允許在其他連接埠上存取用戶端。

存取 資源

消費者可以使用 VPC 端點或透過服務網路直接從其 VPC 存取資源組態。身為消費者，您可以從 VPC 存取您帳戶中的資源組態，或透過其他帳戶與您共用的資源組態 AWS RAM。

- 直接存取資源組態

  您可以在 AWS PrivateLink VPC 中建立類型資源的 VPC 端點（資源端點），以從 VPC 私下存取資源組態。如需如何建立資源端點的詳細資訊，請參閱AWS PrivateLink《使用者指南》中的存取 VPC 資源。
- 透過服務網路存取資源組態

  您可以將資源組態與服務網路建立關聯，並將 VPC 連接到服務網路。您可以透過 關聯或使用服務網路 VPC 端點，將 VPC 連線至 AWS PrivateLink 服務網路。

  如需服務網路關聯的詳細資訊，請參閱管理 VPC Lattice 服務網路的關聯。

如需服務網路 VPC 端點的詳細資訊，請參閱AWS PrivateLink 《 使用者指南》中的存取服務網路。

為您的 VPC 啟用私有 DNS 時，您無法為相同的資源組態建立資源端點和服務網路端點。

與服務網路類型的關聯

當您與取用者帳戶共用資源組態時，例如 Account-B，透過 AWS RAM，Account-B 可以直接透過資源 VPC 端點或透過服務網路存取資源組態。

若要透過服務網路存取資源組態，Account-B 必須將資源組態與服務網路建立關聯。服務網路可在帳戶之間共用。因此，Account-B 可以與 Account-C 共用其服務網路 （ 與資源組態相關聯的 ），讓您的資源可從 Account-C 存取。

為了防止這類暫時性共用，您可以指定資源組態無法新增至可在帳戶之間共用的服務網路。如果您指定此選項，則 Account-B 將無法將您的資源組態新增至已共用或未來可與其他帳戶共用的服務網路。

服務網路的類型

當您透過 與其他帳戶共用資源組態時， AWS RAM Account-B 可以透過下列三種方式之一存取資源組態中指定的資源：

- 使用類型資源的 VPC 端點 （ 資源 VPC 端點 ）。
- 使用類型為服務網路的 VPC 端點 （ 服務網路 VPC 端點 ）。
- 使用服務網路 VPC 關聯。

  當您使用服務網路關聯時，每個資源都會從 129.224.0.0/17 區塊指派給每個子網路的 IP，這是 AWS 擁有且不可路由的。這是 VPC Lattice 用來透過 VPC Lattice 網路將流量路由至 服務的受管字首清單以外的項目。這兩個 IPs都會更新到您的 VPC 路由表。

對於服務網路 VPC 端點和服務網路 VPC 關聯，資源組態必須與 Account-B 中的服務網路相關聯。 服務網路可在帳戶之間共用。因此，Account-B 可以與 Account-C 共用其服務網路 （ 包含資源組態 ），讓您的資源可從 Account-C 存取。 為了防止這類暫時性共用，您可以不允許將資源組態新增至可在帳戶之間共用的服務網路。如果您不允許這麼做，則 Account-B 將無法將您的資源組態新增至共用或可以與其他帳戶共用的服務網路。

透過 共用資源組態 AWS RAM

資源組態已與 整合 AWS Resource Access Manager。您可以透過 與其他 帳戶共用資源組態 AWS RAM。當您與 AWS 帳戶共用資源組態時，該帳戶中的用戶端可以私下存取資源。您可以使用 中的資源共用來共用資源組態 AWS RAM。

使用 AWS RAM 主控台來檢視您新增的資源共用、您可以存取的共用資源，以及與您共用資源 AWS 的帳戶。如需詳細資訊，請參閱AWS RAM 《 使用者指南》中的與您共用的資源。

若要從與資源組態相同的帳戶中的另一個 VPC 存取資源，您不需要透過 共用資源組態 AWS RAM。

監控

您可以在資源組態上啟用監控日誌。您可以選擇要傳送日誌的目的地。

建立和驗證網域

網域名稱驗證是可讓您證明擁有指定網域的實體。身為資源提供者，您可以使用 網域及其子網域做為資源組態的自訂網域名稱。資源取用者在描述資源組態時，可以看到自訂網域名稱的驗證狀態。

開始網域驗證

您可以使用 VPC Lattice 開始網域名稱驗證，然後使用 DNS 區域來完成程序。

AWS 管理主控台

開始網域名稱驗證

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 PrivateLink 和 Lattice 下，選擇網域驗證
3. 選擇開始網域驗證。
4. 針對網域名稱，輸入您擁有的網域名稱。
5. (選用) 若要新增標籤，請選擇 Add new tag (新增標籤)，然後輸入標籤的鍵和值。
6. 選擇開始網域名稱驗證。

網域名稱驗證成功開始後，VPC Lattice 會傳回 `Id`和 `txtMethodConfig`。您可以使用 `txtMethodConfig`來完成網域名稱的驗證。

AWS CLI

下列start-domain-verification命令會啟動網域名稱驗證：

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

輸出看起來如下：

```
{
    "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 會傳回 Id和 txtMethodConfig。您可以使用 txtMethodConfig來完成網域名稱的驗證。在此範例中， txtMethodConfig如下：

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

完成網域名稱驗證

若要完成網域名稱驗證，請在 DNS 區域中新增 TXT 記錄。如果您使用 Route 53，請使用網域名稱的託管區域。當您驗證網域名稱時，也會驗證任何子網域。例如，如果您驗證 example.com，您可以將資源組態與 alpha.example.com 建立關聯，beta.example.com而無需執行任何其他驗證。

若要使用 建立 TXT 記錄 AWS 管理主控台，請參閱使用 Amazon Route 53 主控台建立記錄。

使用 AWS CLI 適用於 Route 53 的 建立 TXT 記錄

1. 使用 change-resource-record-sets 命令搭配下列範例TXT-record.json檔案：

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

2. 使用下列 AWS CLI 命令，將上一個步驟的 TXT 記錄新增至 Route 53 託管區域：

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

將 取代`hosted-zone-id`為您帳戶中託管區域的 Route 53 託管區域 ID。change-batch 參數值指向資料夾 (path/to/your) 中的 JSON 檔案 (TXT-record.json)。

若要檢查網域名稱的驗證狀態，您可以使用 VPC Lattice 主控台或 `get-domain-verification`命令。

驗證網域名稱後，它會保持驗證狀態，直到您將其刪除為止。如果您從 DNS 區域刪除 TXT 記錄，VPC Lattice 會刪除 ，`verification-id`而且您需要重新驗證網域名稱。如果您刪除 DNS 區域中的 TXT 記錄，VPC Lattice 會將您的網域名稱驗證狀態設定為 UNVERIFIED。這不會影響任何現有的資源端點、服務網路端點或服務網路 VPC 與您的資源組態的關聯。若要重新驗證您的網域名稱，請重新開始網域名稱驗證程序。

在 VPC Lattice 中建立資源組態

建立資源組態。

AWS 管理主控台

使用主控台建立資源組態

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 PrivateLink 和 Lattice 下，選擇資源組態。
3. 選擇建立資源組態。
4. 輸入您 AWS 帳戶中唯一的名稱。您無法在建立資源組態後變更此名稱。
5. 針對組態類型，選擇單一或子資源的資源，或子資源群組的資源群組。
6. 選擇您先前建立的資源閘道，或立即建立資源閘道。
7. （選用）若要輸入自訂網域名稱，請執行下列其中一項操作：
   - 如果您有單一類型的資源組態，您可以輸入自訂網域名稱。資源取用者可以使用此網域名稱來存取您的資源組態。
   - 如果您有類型群組和子項的資源組態，您必須先在群組資源組態上指定群組網域。接下來，子資源組態可以具有屬於群組網域子網域的自訂網域。
8. （選用）輸入驗證 ID。

   如果您想要驗證網域名稱，請提供驗證 ID。這可讓資源消費者知道您擁有網域名稱。
9. 選擇您希望此資源組態代表的資源識別符。
10. 選擇您要共用資源的連接埠範圍。
11. 針對關聯設定，指定此資源組態是否可以與可共用的服務網路建立關聯。
12. 針對共用資源組態，選擇可識別可存取此資源之主體的資源共用。
13. （選用）對於監控，如果您想要監控對資源組態的請求和回應，請啟用資源存取日誌和交付目的地。
14. (選用) 若要新增標籤，請選擇 Add new tag (新增標籤)，然後輸入標籤的鍵和值。
15. 選擇建立資源組態。

AWS CLI

下列 create-resource-configuration 命令會建立單一資源組態，並將其與自訂網域名稱 建立關聯`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
```

下列 create-resource-configuration 命令會建立群組資源組態，並將其與自訂網域名稱 建立關聯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
```

下列 create-resource-configuration 命令會建立子資源組態，並將其與自訂網域名稱 建立關聯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 資源組態的關聯

與您帳戶中的 和用戶端共用資源組態的消費者帳戶可以直接使用類型資源的 VPC 端點或透過類型服務網路的 VPC 端點存取資源組態。因此，您的資源組態將具有端點關聯和服務網路關聯。

管理服務網路資源關聯

建立或刪除服務網路關聯。

> **Note**
>
> 如果您在建立服務網路與資源組態之間的關聯時收到存取遭拒訊息，請檢查您的 AWS RAM 政策版本，並確保其為版本 2。如需詳細資訊，請參閱 AWS RAM 使用者指南。

使用主控台管理服務網路關聯

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 PrivateLink 和 Lattice 下，選擇資源組態。
3. 選取資源組態的名稱以開啟其詳細資訊頁面。
4. 選取服務網路關聯索引標籤。
5. 選擇建立關聯。
6. 從 VPC Lattice 服務網路選取服務網路。若要建立服務網路，請選擇建立 VPC Lattice 網路。
7. （選用）若要新增標籤，請展開服務關聯標籤，選擇新增標籤，然後輸入標籤索引鍵和標籤值。
8. （選用）若要啟用此服務網路資源關聯的私有 DNS 名稱，請選擇啟用私有 DNS 名稱。如需詳細資訊，請參閱the section called “服務網路擁有者的自訂網域名稱”。
9. 選擇 Save changes (儲存變更)。
10. 若要刪除關聯，請選取關聯的核取方塊，然後選擇動作、刪除。出現確認提示時，請輸入 **`confirm`**，然後選擇 Delete (刪除)。

使用 建立服務網路關聯 AWS CLI

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

使用 刪除服務網路關聯 AWS CLI

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

管理資源 VPC 端點關聯

可存取您資源組態或您帳戶中用戶端的消費者帳戶可以使用資源 VPC 端點來存取資源組態。如果您的資源組態具有自訂網域名稱，您可以使用啟用私有 DNS 來允許 VPC Lattice 為您的資源端點或服務網路端點佈建私有託管區域。這樣，用戶端可以直接控制網域名稱來存取資源組態。如需詳細資訊，請參閱the section called “資源取用者的自訂網域名稱”。

AWS 管理主控台

1. 若要建立新的端點關聯，請前往左側導覽窗格中的 PrivateLink 和 Lattice，然後選擇端點。
2. 選擇建立端點。
3. 選取您要連線至 VPC 的資源組態。
4. 選取 VPC、子網路和安全群組。

5. （選用）若要開啟私有 DNS 並設定 DNS 選項，請選取啟用私有 DNS 名稱。
6. （選用）若要標記 VPC 端點，請選擇新增標籤，然後輸入標籤索引鍵和標籤值。
7. 選擇建立端點。

AWS CLI

下列 create-vpc-endpoint 命令會建立使用私有 DNS 的 VPC 端點。私有 DNS 偏好設定設定為 ，VERIFIED_AND_SELECTED而選取的網域為 example.com和 example.org。VPC Lattice 只會為任何已驗證的網域或 example.com或 佈建私有託管區域example.org。

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

使用 建立 VPC 端點關聯 AWS CLI

使用 create-vpc-endpoint 命令。

使用 刪除 VPC 端點關聯 AWS CLI

使用 delete-vpc-endpoint 命令。

分享您的 VPC Lattice 實體

Amazon VPC Lattice 與 AWS Resource Access Manager (AWS RAM) 整合，以啟用共享服務、資源組態和服務網路。 AWS RAM 是一種服務，可讓您與其他 AWS 帳戶 或透過 共享一些 VPC Lattice 實體 AWS Organizations。與 AWS RAM共用您透過建立資源共用擁有的實體。資源共用會指定要共用的實體，以及要與其共用的消費者。消費者可包括：

- 組織 AWS 帳戶 內部或外部的特定 AWS Organizations。
- AWS Organizations中組織內的組織單位。
- 中的整個組織 AWS Organizations。

如需 的詳細資訊 AWS RAM，請參閱AWS RAM 《 使用者指南》 。

目錄

共用 VPC Lattice 實體的先決條件

- 若要共用實體，您必須在 中擁有該實體 AWS 帳戶。這表示必須在您的帳戶中配置或佈建實體。您無法共用已與您共用的實體。
- 若要與組織或 中的組織單位共用實體 AWS Organizations，您必須啟用與 共用 AWS Organizations。如需詳細資訊，請參閱《AWS RAM 使用者指南》中的透過 AWS Organizations啟用共用。

共用 VPC Lattice 實體

若要共用實體，請先使用 建立資源共用 AWS Resource Access Manager。資源共用會指定要共用的實體、與其共用的消費者，以及主體可以執行的動作。

當您與其他人共用您擁有的 VPC Lattice 實體時 AWS 帳戶，您可以讓這些帳戶將其實體與您帳戶中的實體建立關聯。當您針對共用實體建立關聯時，我們會在實體擁有者帳戶和建立關聯的帳戶中產生 Amazon Resource Name (ARN)。因此，實體擁有者和建立關聯的帳戶都可以刪除關聯。

如果您是 中組織的一部分， AWS Organizations 且已啟用組織內的共用，則組織中的消費者會自動獲得共用實體的存取權。否則，消費者會收到加入資源共享的邀請，並在接受邀請後授予共用實體的存取權。

考量事項

- 您可以共用三種類型的 VPC Lattice 實體：服務網路、服務和資源組態。
- 您可以與任何 共用 VPC Lattice 實體 AWS 帳戶。
- 您無法與個別 IAM 使用者和角色共用 VPC Lattice 實體。
- VPC Lattice 支援服務、資源組態和服務網路的客戶受管許可。

使用 VPC Lattice 主控台共用您擁有的實體

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務、服務網路或資源組態。
3. 選擇實體的名稱以開啟其詳細資訊頁面，然後從共用索引標籤中選擇共用服務、共用服務網路或共用資源組態。
4. 從 AWS RAM 資源共用中選擇資源共用。若要建立資源共享，請選擇在 RAM 主控台中建立資源共享。
5. 選擇共用服務、共用服務網路或共用資源組態。

使用 AWS RAM 主控台共用您擁有的實體

使用 AWS RAM 使用者指南中建立資源共享所述的程序。

使用 共享您擁有的實體 AWS CLI

使用 associate-resource-share 命令。

停止共用 VPC Lattice 實體

若要停止共用您擁有的 VPC Lattice 實體，您必須將其從資源共用中移除。在您停止共用實體之後，現有的關聯會持續存在。不允許與先前共用實體的新關聯。當實體擁有者或關聯擁有者刪除關聯時，會從

兩個帳戶中刪除該關聯。如果帳戶擁有者想要離開資源共用，則必須要求資源共用的擁有者從此資源共用的帳戶清單中移除其帳戶。

使用 VPC Lattice 主控台停止共用您擁有的實體

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務、服務網路或資源組態。
3. 選擇實體的名稱以開啟其詳細資訊頁面。
4. 在共用索引標籤上，選取資源共用的核取方塊，然後選擇移除。

使用 AWS RAM 主控台停止共用您擁有的實體

請參閱《AWS RAM 使用者指南》中的更新資源共用。

使用 停止共用您擁有的實體 AWS CLI

使用 disassociate-resource-share 命令。

責任和許可

使用共用 VPC Lattice 實體時，適用下列責任和許可。

實體擁有者

- 服務網路擁有者無法修改消費者建立的服務。
- 服務網路擁有者無法刪除消費者建立的服務。
- 服務網路擁有者可以描述服務網路的所有服務關聯。
- 無論誰建立關聯，服務網路擁有者都可以取消與服務網路關聯的任何服務。
- 服務網路擁有者可以描述服務網路的所有 VPC 關聯。
- 服務網路擁有者可以取消消費者與服務網路關聯的任何 VPC 的關聯。
- 服務網路擁有者可以描述服務網路的所有資源組態關聯。
- 無論誰建立關聯，服務網路擁有者都可以取消與服務網路關聯的任何資源組態的關聯。
- 服務網路擁有者可以描述服務網路的所有端點關聯。
- 服務網路擁有者可以取消與服務網路關聯的任何端點的關聯，無論關聯是由誰建立。
- 服務擁有者可以描述與服務的所有服務網路關聯。

- 服務擁有者可以將服務與其關聯的任何服務網路取消關聯。
- 資源組態擁有者可以描述與資源組態的所有網路關聯。
- 資源組態擁有者可以取消資源組態與其關聯的任何服務網路的關聯。
- VPC 端點擁有者可以描述與其相關聯的服務網路。
- VPC 端點擁有者可以取消端點與服務網路的關聯。
- 只有建立關聯的帳戶才能更新服務網路和 VPC 之間的關聯。

實體消費者

- 消費者無法刪除他們未建立的服務或資源組態。
- 消費者只能取消與服務網路相關聯的服務或資源組態的關聯。
- 消費者和網路擁有者可以描述服務網路與服務或資源組態之間的所有關聯。
- 消費者無法擷取其未擁有之資源組態的服務或資源組態資訊。
- 消費者可以描述所有服務關聯和資源組態與共用服務網路的關聯。
- 消費者可以將服務或資源組態與共用服務網路建立關聯。
- 消費者可以看到與共用服務網路的所有 VPC 關聯。
- 消費者可以將 VPC 與共用服務網路建立關聯。
- 消費者只能取消與服務網路相關聯之 VPCs的關聯。
- 消費者可以建立服務網路 VPC 端點，將其 VPC 連線至共用服務網路。
- 消費者只能刪除他們為了將 VPC 連線到共用服務網路而建立的服務網路 VPC 端點。
- 共用服務的取用者無法將服務與其未擁有的服務網路建立關聯。
- 共用服務網路的取用者無法關聯他們不擁有的 VPC 或服務。
- 共用資源組態的取用者無法將資源組態與其未擁有的服務網路建立關聯。
- 共用服務網路的取用者無法關聯他們不擁有的 VPC 或服務或資源組態。
- 消費者可以描述與其共用的服務、服務網路或資源組態。
- 如果兩個實體共用，消費者就無法關聯這兩個實體。

跨帳戶事件

當實體擁有者和取用者對共用實體執行動作時，這些動作會記錄為 中的跨帳戶事件 AWS CloudTrail。

`CreateServiceNetworkResourceAssociationBySharee`

當實體消費者使用共用實體呼叫 CreateServiceNetworkResourceAssociation 時，傳送給實體擁有者。如果發起人擁有資源組態，則事件會傳送至服務網路的擁有者。如果發起人擁有服務網路，則事件會傳送至資源組態的擁有者。

`CreateServiceNetworkServiceAssociationBySharee`

當實體消費者呼叫 CreateServiceNetworkServiceAssociation 與共用實體時，傳送給實體擁有者。如果發起人擁有該服務，則事件會傳送至服務網路的擁有者。如果發起人擁有服務網路，則事件會傳送給服務的擁有者。

`CreateServiceNetworkVpcAssociationBySharee`

當實體消費者使用共用服務網路呼叫 CreateServiceNetworkVpcAssociation 時，傳送給實體擁有者。

`DeleteServiceNetworkResourceAssociationByOwner`

當實體擁有者使用共用實體呼叫 DeleteServiceNetworkResourceAssociation 時，會傳送給關聯擁有者。如果發起人擁有資源組態，則事件會傳送至服務網路關聯的擁有者。如果發起人擁有服務網路，則事件會傳送至資源關聯的擁有者。

`DeleteServiceNetworkResourceAssociationBySharee`

當實體消費者使用共用實體呼叫 DeleteServiceNetworkResourceAssociation 時，傳送給實體擁有者。如果發起人擁有資源組態，則事件會傳送至服務網路的擁有者。如果發起人擁有服務網路，則事件會傳送至資源組態的擁有者。

`DeleteServiceNetworkServiceAssociationByOwner`

當實體擁有者呼叫 DeleteServiceNetworkServiceAssociation 與共用實體時，會傳送給關聯擁有者。如果發起人擁有該服務，則事件會傳送給服務網路關聯的擁有者。如果發起人擁有服務網路，則事件會傳送給服務關聯的擁有者。

`DeleteServiceNetworkServiceAssociationBySharee`

當實體消費者呼叫 DeleteServiceNetworkServiceAssociation 與共用實體時，傳送給實體擁有者。如果發起人擁有該服務，則事件會傳送至服務網路的擁有者。如果發起人擁有服務網路，則事件會傳送給服務的擁有者。

`DeleteServiceNetworkVpcAssociationByOwner`

當實體擁有者使用共用服務網路呼叫 DeleteServiceNetworkVpcAssociation 時，會傳送給關聯擁有者。

`DeleteServiceNetworkVpcAssociationBySharee`

當實體消費者使用共用服務網路呼叫 DeleteServiceNetworkVpcAssociation 時，傳送給實體擁有者。

`GetServiceBySharee`

當實體消費者使用共用服務呼叫 GetService 時，傳送給實體擁有者。

`GetServiceNetworkBySharee`

當實體消費者使用共用服務網路呼叫 GetServiceNetwork 時，傳送給實體擁有者。

`GetServiceNetworkResourceAssociationBySharee`

當實體消費者使用共用實體呼叫 GetServiceNetworkResourceAssociation 時，傳送給實體擁有者。如果發起人擁有資源組態，則事件會傳送至服務網路的擁有者。如果發起人擁有服務網路，則事件會傳送至資源組態的擁有者。

`GetServiceNetworkServiceAssociationBySharee`

當實體消費者呼叫 GetServiceNetworkServiceAssociation 與共用實體時，傳送給實體擁有者。如果發起人擁有該服務，則事件會傳送至服務網路的擁有者。如果發起人擁有服務網路，則事件會傳送給服務的擁有者。

`GetServiceNetworkVpcAssociationBySharee`

當實體消費者使用共用服務網路呼叫 GetServiceNetworkVpcAssociation 時，傳送給實體擁有者。

以下是`CreateServiceNetworkServiceAssociationBySharee`事件的範例項目。

```
{
    "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"
}
```

的 VPC Lattice Oracle Database@AWS

VPC Lattice 支援 Oracle Database@AWS(ODB) 的 AWS 受管服務整合，並為您提供 ODB 網路、AWS VPCs 和內部部署之間的簡化連線。為了支援此連線，VPC Lattice 會代表您佈建下列實體：

預設服務網路

預設服務網路使用命名慣例 `default-odb-network-randomHash`

預設服務網路端點

此 AWS 資源沒有名稱。

資源閘道

資源閘道使用命名慣例 `default-odb-network-randomHash`

VPC Lattice 支援 AWS 受管服務整合，稱為 ODB 網路的受管整合。預設會啟用 Oracle Cloud Infrastructure (OCI) 受管備份至 Amazon S3。您可以選擇啟用 Amazon S3 和零 ETL 的自我管理存取。

建立 ODB 網路後，您可以使用 AWS 管理主控台 或 檢視佈建的資源 AWS CLI。下列範例命令會列出 ODB 網路的預設受管整合，以及您針對此服務網路可能擁有的任何其他資源：

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

考量事項

下列考量適用於 VPC Lattice Oracle Database@AWS：

- 您無法刪除 VPC Lattice 佈建的預設服務網路、服務網路端點、資源閘道或任何 ODB 受管整合。若要刪除這些實體，請刪除您的 ODB 網路或停用受管整合。
- 用戶端只能存取 ODB 網路中的受管整合。ODB 網路以外的用戶端，例如在您的 VPCs中，無法使用這些受管整合來存取 S3 或零 ETL。
- 您無法連線到 VPC Lattice 佈建之 ODB 網路以外的任何受管整合。
- 所有流向 Amazon S3 的流量都會經過預設的服務網路端點，並收取存取資源的標準處理費用。所有零 ETL 流量都會經過資源閘道，而您共用的資源會收取標準資料處理費用。如需詳細資訊，請參閱 VPC Lattice 定價。

- Oracle Database@AWS 受管整合不收取每小時費用。
- 您可以管理 VPC Lattice 佈建的資源，就像任何其他服務網路一樣。您可以與其他 AWS 帳戶 或組織共用預設服務網路，並將新的端點、VPC 關聯、VPC Lattice 服務和資源新增至預設網路。
- VPC Lattice 需要下列許可才能佈建 Oracle Database@AWS 資源：

  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"
     ]
    }
   }
 }
  ]
}
```

若要將 VPC Lattice 用於 Oracle Database@AWS，建議您熟悉 VPC Lattice 中的服務網路、服務網路關聯和資源閘道。

主題

- the section called “Oracle Cloud Infrastructure (OCI) 受管備份至 Amazon S3”
- the section called “Amazon S3 存取”
- the section called “適用於 Amazon Redshift 的零 ETL”
- the section called “存取和共用 VPC Lattice 實體”

Oracle Cloud Infrastructure (OCI) 受管備份至 Amazon S3

當您建立 Oracle Database@AWS 資料庫時，VPC Lattice 會建立名為 的資源組態odb-managed-s3-backup-access。此資源組態代表將資料庫的 OCI 受管備份至 Amazon S3，並且僅啟用與 OCI 所擁有 Amazon S3 儲存貯體的連線。ODB 網路和 S3 之間的流量永遠不會離開 Amazon 網路。

Amazon S3 存取

除了 OCI 受管備份到 Amazon S3 之外，您還可以建立受管整合，以便從 ODB 網路存取 Amazon S3。當您修改 Oracle Database@AWS 網路以啟用 Amazon S3 Access 受管整合時，VPC Lattice 會在預設服務網路odb-s3-access中佈建名為 的資源組態。您可以使用此整合來存取 Amazon S3 以滿足自己的需求，包括自我管理備份或還原。您可以透過提供身分驗證政策來建立周邊控制。

考量事項

以下是 Amazon S3 Access 受管整合的考量：

- 您只能為 ODB 網路建立一個 Amazon S3 Access 受管整合。
- 此受管整合僅允許從 ODB 網路存取 Amazon S3，而不是從預設服務網路中的其他 VPC 關聯或服務網路端點存取。
- 您無法存取不同 AWS 區域中的 S3 儲存貯體。

啟用 Amazon S3 Access 受管整合

使用下列命令來啟用 Amazon S3 Access 受管整合：

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

使用身分驗證政策進行安全存取

您可以使用 ODB API 定義身分驗證政策，以保護對 S3 儲存貯體的存取。下列範例政策會授予特定組織擁有之特定 S3 儲存貯體的存取權。

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 受管整合時`aws:SourceVpce`，S3 儲存貯體政策不支援 `aws:SourceVpc`、 和 `aws:VpcSourceIp`條件金鑰。

適用於 Amazon Redshift 的零 ETL

您可以使用 VPC Lattice 佈建的服務網路來啟用零 ETL。此受管整合會將 ODB 網路資料庫連線至 Amazon Redshift，以協助分析不同資料庫的資料。您可以使用 AWS Glue 整合 APIs 啟動零 ETL 設定，並使用 ODB APIs 開啟受管整合並設定網路路徑。如需詳細資訊，請參閱與 Amazon Redshift 的零 ETL 整合。

考量事項

以下是 受管零 ETL 整合的考量：

- 如果您啟用 受管零 ETL 整合，您只能使用零 ETL 存取 ODB 網路中的執行個體。與您的服務網路相關聯的其他服務和資源會與 Zero-ETL 隔離。

存取和共用 VPC Lattice 實體

您也可以使用 VPCs Lattice 將 ODB 網路連線到 VPC 中的服務、資源和其他用戶端。這些連線選項是透過 VPC Lattice 佈建的預設服務網路、資源閘道和服務網路端點提供支援。

存取 VPC Lattice 服務和資源

若要存取其他實體，請將您擁有或與您共用的 服務或資源關聯至預設服務網路。ODB 網路中的用戶端可以透過預設的服務網路端點存取服務或資源。

考量事項

以下是連接到其他 VPC Lattice 實體的考量事項：

- 您可以將新的服務網路端點、VPC 關聯、VPC Lattice 資源和服務新增至服務網路，但您無法代表 ODB 網路修改 VPC Lattice 佈建的資源。這些必須透過 Oracle Database@AWS APIs管理。

透過 VPC Lattice 共用您的 ODB 網路

您可以與其他 VPCs、帳戶或內部部署中的用戶端共用 ODB 網路資源。若要開始使用，請為您要共用的資源建立資源組態。資源組態必須使用 ODB 網路的預設資源閘道。然後，您可以將資源與您的預設服務網路建立關聯。

在其他 VPCs中或您已與之共用服務網路 AWS 帳戶 的用戶端，可以透過自己的服務網路端點或 VPC 關聯來存取這些資源。如需詳細資訊，請參閱the section called “管理關聯”。

考量事項

以下是共用 ODB 網路的考量：

- 我們建議僅將 ODB 網路執行個體做為 IP 型資源共用。
- VPC Lattice 不支援 OCI 的單一用戶端存取名稱 (SCAN) 接聽程式 DNS。

Amazon VPC Lattice 的安全性

的雲端安全性 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構是為了符合最安全敏感組織的需求而建置。

您負責維護在此基礎設施上託管內容的控制權。共同責任模式將其描述為雲端的安全性，和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 中執行 AWS 服務的基礎設施 AWS 雲端。 AWS 也為您提供可安全使用的服務。在AWS 合規計畫中，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用於 Amazon VPC Lattice 的合規計劃，請參閱AWS 合規計劃的 服務範圍。
- 雲端安全 – 您需負責控制在此基礎設施上託管的內容。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 VPC Lattice 時套用共同責任模型。下列主題說明如何設定 VPC Lattice 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務，以協助您監控和保護 VPC Lattice 服務、服務網路和資源組態。

目錄

管理對 VPC Lattice 服務的存取

根據預設，VPC Lattice 是安全的，因為您必須明確哪些服務和資源組態可提供對哪些 VPCs存取。您可以透過 VPC 關聯或類型為服務網路的 VPC 端點存取服務。對於多帳戶案例，您可以使用 AWS Resource Access Manager 跨帳戶邊界共用服務、資源組態和服務網路。

VPC Lattice 提供架構，可讓您在網路的多層實作defense-in-depth策略。

- 第一層 – 服務、資源、VPC 和 VPC 端點與服務網路的關聯。VPC 可以透過 關聯或透過 VPC 端點連線到服務網路。如果 VPC 未連線至服務網路，VPC 中的用戶端無法存取與服務網路相關聯的服務和資源組態。
- 第二層 – 服務網路的選用網路層級安全保護，例如安全群組和網路 ACLs。透過使用這些項目，您可以允許存取 VPC 中的特定用戶端群組，而不是 VPC 中的所有用戶端。
- 第三層 – 選用的 VPC Lattice 驗證政策。您可以將身分驗證政策套用至服務網路和個別服務。一般而言，服務網路上的身分驗證政策是由網路或雲端管理員操作，而且它們會實作粗略精細的授權。例如，僅允許來自特定組織的已驗證請求 AWS Organizations。對於服務層級的身分驗證政策，通常服務擁有者會設定精細程度控制，這可能比在服務網路層級套用的粗略程度授權更嚴格。

> **Note**
>
> 服務網路上的身分驗證政策不適用於服務網路中的資源組態。

存取控制的方法

- 驗證政策
- Security groups (安全群組)
- 網路 ACL

使用身分驗證政策控制對 VPC Lattice 服務的存取

VPC Lattice 驗證政策是您連接到服務網路或服務的 IAM 政策文件，以控制指定的委託人是否可以存取一組服務或特定服務。您可以針對每個要控制存取權的服務網路或服務，均附加一個驗證政策。

> **Note**
>
> 服務網路上的身分驗證政策不適用於服務網路中的資源組態。

驗證政策與 IAM 身分型政策不同。IAM 身分型政策會連接到 IAM 使用者、群組或角色，並定義這些身分可以對哪些資源執行哪些動作。驗證政策會連接到 服務和 服務網路。若要成功授權，身分驗證政策和身分型政策都需要有明確的允許陳述式。如需詳細資訊，請參閱授權的運作方式。

您可以使用 AWS CLI 和 主控台來檢視、新增、更新或移除 服務和服務網路上的身分驗證政策。當您新增、更新或移除身分驗證政策時，可能需要幾分鐘的時間才能準備就緒。使用 時 AWS CLI，請確定您位於正確的區域。您可以變更設定檔的預設區域，或搭配 命令使用 `--region` 參數。

目錄

若要開始使用身分驗證政策，請依照程序建立套用至服務網路的身分驗證政策。對於您不希望套用至其他服務的更嚴格許可，您可以選擇在個別服務上設定身分驗證政策。

使用身分驗證政策管理對服務網路的存取

下列 AWS CLI 任務說明如何使用身分驗證政策管理對服務網路的存取。如需使用 主控台的指示，請參閱 VPC Lattice 中的服務網路。

任務

將身分驗證政策新增至服務網路

請依照本節中的步驟使用 AWS CLI 來：

- 使用 IAM 在服務網路上啟用存取控制。
- 將身分驗證政策新增至服務網路。如果您未新增身分驗證政策，所有流量都會收到存取遭拒錯誤。

啟用存取控制並將身分驗證政策新增至新的服務網路

1. 若要啟用服務網路上的存取控制，使其可以使用身分驗證政策，請使用 create-service-network命令搭配 --auth-type選項和 值AWS_IAM。

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

如果成功，此命令傳回的輸出會類似如下。

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

2. 使用 put-auth-policy命令，指定您要新增身分驗證政策和您要新增之身分驗證政策的服務網路 ID。

例如，使用下列命令，為 ID 為 的服務網路建立身分驗證政策*sn-0123456789abcdef0*。

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

使用 JSON 建立政策定義。如需詳細資訊，請參閱驗證政策中的常見元素。

如果成功，此命令傳回的輸出會類似如下。

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

啟用存取控制並將身分驗證政策新增至現有的服務網路

1. 若要啟用服務網路上的存取控制，使其可以使用身分驗證政策，請使用 update-service-network命令搭配 --auth-type選項和 值AWS_IAM。

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

如果成功，此命令傳回的輸出會類似如下。

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

2. 使用 put-auth-policy命令，指定您要新增身分驗證政策和您要新增之身分驗證政策的服務網路 ID。

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

使用 JSON 建立政策定義。如需詳細資訊，請參閱驗證政策中的常見元素。

如果成功，此命令傳回的輸出會類似如下。

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

變更服務網路的身分驗證類型

停用服務網路的身分驗證政策

使用 update-service-network命令搭配 `--auth-type`選項和 的值`NONE`。

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

如果您稍後需要再次啟用身分驗證政策，請使用為 `--auth-type`選項`AWS_IAM`指定的 執行此命令。

從服務網路移除身分驗證政策

從服務網路移除身分驗證政策

使用 delete-auth-policy 命令。

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

如果您在將服務網路的身分驗證類型變更為 之前移除身分驗證政策，請求會失敗NONE。

使用身分驗證政策管理對服務的存取

下列 AWS CLI 任務說明如何使用身分驗證政策管理對服務的存取。如需使用 主控台的指示，請參閱 VPC Lattice 中的服務。

任務

- 將身分驗證政策新增至服務
- 變更服務的身分驗證類型
- 從服務移除身分驗證政策

將身分驗證政策新增至服務

請依照下列步驟使用 AWS CLI 來：

- 使用 IAM 在服務上啟用存取控制。
- 將身分驗證政策新增至服務。如果您未新增身分驗證政策，所有流量都會收到存取遭拒錯誤。

啟用存取控制並將身分驗證政策新增至新服務

1. 若要啟用服務的存取控制，使其可以使用身分驗證政策，請使用 create-service命令搭配 --auth-type選項和 值AWS_IAM。

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

   如果成功，此命令傳回的輸出會類似如下。

   ```
   {
      "arn": "arn",
   ```

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

2. 使用 put-auth-policy命令，指定您要新增身分驗證政策的 服務 ID，以及您要新增的身分驗證政策。

   例如，使用下列命令，為 ID 為 *svc-0123456789abcdef0* 的服務建立身分驗證政策。

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

   使用 JSON 建立政策定義。如需詳細資訊，請參閱驗證政策中的常見元素。

   如果成功，此命令傳回的輸出會類似如下。

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

啟用存取控制並將身分驗證政策新增至現有服務

1. 若要啟用服務的存取控制，使其可以使用身分驗證政策，請使用 update-service命令搭配 `--auth-type`選項和 值`AWS_IAM`。

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

   如果成功，此命令傳回的輸出會類似如下。

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

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

2. 使用 put-auth-policy命令，指定您要新增身分驗證政策的 服務 ID，以及您要新增的身分驗證政策。

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

使用 JSON 建立政策定義。如需詳細資訊，請參閱驗證政策中的常見元素。

如果成功，此命令傳回的輸出會類似如下。

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

變更服務的身分驗證類型

停用服務的身分驗證政策

使用 update-service命令搭配 --auth-type選項和 的值NONE。

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

如果您稍後需要再次啟用身分驗證政策，請使用為 --auth-type選項AWS_IAM指定的 執行此命令。

從服務移除身分驗證政策

從服務中移除身分驗證政策

使用 delete-auth-policy 命令。

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

如果您在將服務的身分驗證類型變更為 之前移除身分驗證政策，請求會失敗NONE。

如果您啟用需要對服務進行驗證請求的身分驗證政策，對該服務的任何請求都必須包含使用 Signature 第 4 版 (SigV4) 計算的有效請求簽章。如需詳細資訊，請參閱Amazon VPC Lattice 的 SIGv4 驗證請求。

驗證政策中的常見元素

VPC Lattice 驗證政策是使用與 IAM 政策相同的語法來指定。如需詳細資訊，請參閱《IAM 使用者指南》中的身分型政策和資源型政策。

驗證政策包含下列元素：

- 委託人 – 允許存取 陳述式中動作和資源的人員或應用程式。在身分驗證政策中，委託人是此許可的收件人 IAM 實體。委託人會驗證為 IAM 實體，以向特定資源或資源群組提出請求，就像服務網路中的服務一樣。

  您必須在資源型政策中指定主體。委託人可以包括帳戶、使用者、角色、聯合身分使用者 AWS 或服務。如需詳細資訊，請參閱《IAM 使用者指南》中的 AWS JSON 政策元素：主體。
- 效果 – 指定委託人請求特定動作時的效果。可以是 `Allow` 或 `Deny`。根據預設，當您使用 IAM 在服務或服務網路上啟用存取控制時，主體沒有向服務或服務網路提出請求的許可。
- 動作 – 您要授予或拒絕許可的特定 API 動作。VPC Lattice 支援使用 `vpc-lattice-svcs`字首的動作。如需詳細資訊，請參閱《服務授權參考》中的 Amazon VPC Lattice Services 定義的動作。
- 資源 – 受 動作影響的服務。
- 條件 – 條件為選用。您可以使用它們來控制政策生效的時間。如需詳細資訊，請參閱《服務授權參考》中的 Amazon VPC Lattice Services 的條件金鑰。

當您建立和管理身分驗證政策時，建議您使用 IAM 政策產生器。

需求

JSON 中的政策不得包含換行或空白行。

驗證政策的資源格式

您可以透過建立身分驗證政策來限制對特定資源的存取，該政策使用相符的結構描述搭配`<serviceARN>/<path>`模式和 `Resource`元素的程式碼，如下列範例所示。

通訊協定	範例
HTTP	• "Resource": "arn:aws:vpc-lattice:us-west-2:1234567890:service/svc-0123456789abcdef0/rates" • "Resource": "*/rates" • "Resource": "*/*"
gRPC	• "Resource": "arn:aws:vpc-lattice:us-west-2:1234567890:service/svc-0123456789abcdef0/api.parking/GetRates" • "Resource": "arn:aws:vpc-lattice:us-west-2:1234567890:service/svc-0123456789abcdef0/api.parking/*" • "Resource": "arn:aws:vpc-lattice:us-west-2:1234567890:service/svc-0123456789abcdef0/*"

針對 使用以下 Amazon Resource Name (ARN) 資源格式<serviceARN>：

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

例如：

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

可用於身分驗證政策的條件索引鍵

身分驗證政策的條件元素中的條件索引鍵可以進一步控制存取。根據通訊協定以及請求是使用 Signature 第 4 版 (SigV4) 還是匿名簽署，這些條件金鑰會用於評估。條件金鑰名稱有區分大小寫。

AWS 提供全域條件金鑰，您可以用來控制存取，例如 aws:PrincipalOrgID和 aws:SourceIp。若要查看 AWS 全域條件索引鍵的清單，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

下列故事列出 VPC Lattice 條件索引鍵。如需詳細資訊，請參閱《服務授權參考》中的 Amazon VPC Lattice Services 的條件金鑰。

條件索引鍵	說明	範例	適用於匿名（未驗證）來電者？	適用於 gRPC？
`vpc-lattice-svcs:Port`	依向 提出請求的服務連接埠篩選存取權	80	是	是
`vpc-lattice-svcs:RequestMethod`	依請求的方法來篩選存取權	GET	是	一律 POST
`vpc-lattice-svcs:RequestPath`	依請求 URL 的路徑部分篩選存取權	/path	是	是
`vpc-lattice-svcs:RequestHeader/` *`header-name`* `:` *`value`*	依請求標頭中的標頭名稱/值對來篩選存取權	content-type: application/json	是	是
`vpc-lattice-svcs:RequestQueryString/` *`key-name`*`:` *`value`*	依請求 URL 中的查詢字串鍵值對來篩選存取權	quux: [corge, grault]	是	否
`vpc-lattice-svcs:ServiceNetworkArn`	依接收請求之服務之服務網路的 ARN 篩選存取權	arn:aws:vpc-lattice:us-west-2:123456789012:servicenetwork/sn-0123456789abcdef0	是	是
`vpc-lattice-svcs:ServiceArn`	依接收請求之服務的 ARN 篩選存取權	arn:aws:vpc-lattic	是	是

條件索引鍵	說明	範例	適用於匿名（未驗證）來電者？	適用於 gRPC？
		e:us-west-2:123456789012:service/svc-0123456789abcdef0		
vpc-lattice-svcs:SourceVpc	依提出請求的 VPC 來篩選存取權	vpc-1a2b3c4d	是	是
vpc-lattice-svcs:SourceVpcOwnerAccount	依提出請求的 VPC 所屬帳戶來篩選存取權	123456789012	是	是

資源標籤

標籤是您指派或 AWS 指派給 AWS 資源的中繼資料標籤。每個 標籤都有兩個部分：

- 標籤鍵 (例如，CostCenter、Environment 或 Project)。標籤鍵會區分大小寫。
- 選用欄位，稱為標籤值 (例如 111122223333 或 Production)。忽略標籤值基本上等同於使用空字串。與標籤鍵相同，標籤值會區分大小寫。

如需標記的詳細資訊，請參閱使用標籤控制對 AWS 資源的存取

您可以使用aws:ResourceTag/key AWS 全域條件內容索引鍵，在身分驗證政策中使用標籤。

下列範例政策會授予具有標籤 之 服務的存取權Environment=Gamma。此政策可讓您在沒有硬式編碼服務 ARNs或 IDs 的情況下參考 服務。

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

主體標籤

您可以根據連接到發起人身分的標籤來控制對 服務和資源的存取。VPC Lattice 支援根據使用者、角色或工作階段標籤上使用aws:PrincipalTag/context變數的任何主體標籤進行存取控制。如需詳細資訊，請參閱控制 IAM 主體的存取。

下列範例政策只會將存取權授予具有標籤 的身分Team=Payments。此政策可讓您在沒有硬式編碼帳戶 IDs或角色 ARNs的情況下控制存取。

```
{
  "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 （未驗證） 主體

匿名主體是未使用 Signature 第 4 版 (SigV4) 簽署 AWS 請求的發起人，且位於連接到服務網路的 VPC 內。如果身分驗證政策允許，匿名主體可以對服務網路中的服務提出未經驗證的請求。

驗證政策範例

以下是要求已驗證委託人提出請求的身分驗證政策範例。

所有範例都使用 `us-west-2`區域，並包含虛構的帳戶 IDs。

範例 1：限制特定 AWS 組織對 服務的存取

下列身分驗證政策範例會授予許可給任何已驗證的請求，以存取政策適用的服務網路中的任何服務。不過，請求必須來自屬於 條件中指定 AWS 組織的委託人。

JSON

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

範例 2：限制特定 IAM 角色對服務的存取

下列身分驗證政策範例會授予許可給任何使用 IAM 角色`rates-client`對 `Resource`元素中指定的服務提出 HTTP GET 請求的已驗證請求。`Resource` 元素中的資源與附加政策的服務相同。

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"
            }
         }
      }
   ]
}
```

範例 3：限制特定 VPC 中已驗證主體對 服務的存取

下列身分驗證政策範例僅允許來自 VPC 中 VPC ID 為 之主體的已驗證請求*vpc-1a2b3c4d*。

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"
            }
         }
      }
   ]
}
```

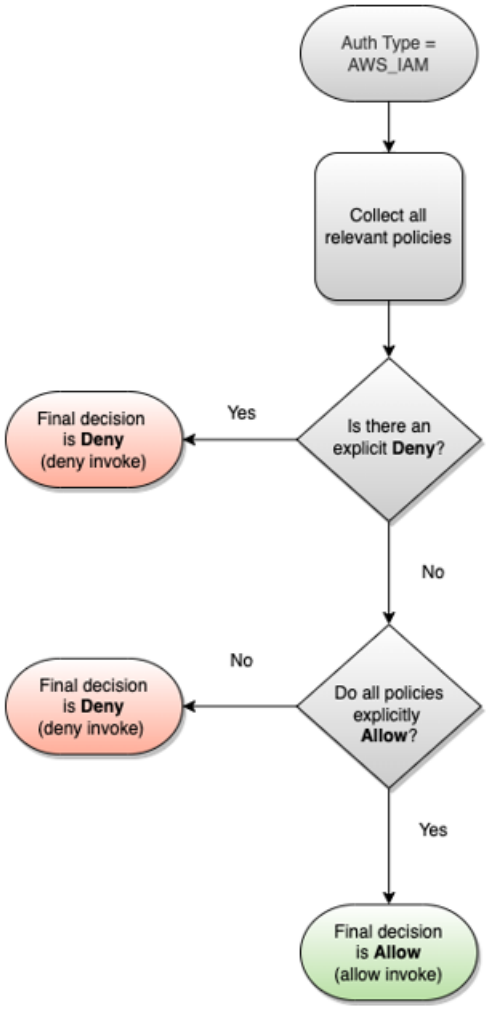
授權的運作方式

當 VPC Lattice 服務收到請求時， AWS 強制執行程式碼會一起評估所有相關許可政策，以決定是否授權或拒絕請求。它會評估授權期間適用於請求內容的所有 IAM 身分型政策和身分驗證政策。根據預設，當身分驗證類型為 時，所有請求都會隱含拒絕AWS_IAM。所有相關政策的明確允許會覆寫預設值。

授權包括：

- 收集所有相關 IAM 身分型政策和身分驗證政策。
- 評估產生的一組政策：
  - 驗證申請者（例如 IAM 使用者或角色）是否具有從申請者所屬帳戶執行操作的許可。如果沒有明確的允許陳述式， AWS 不會授權請求。
  - 驗證服務網路的身分驗證政策是否允許請求。如果已啟用身分驗證政策，但沒有明確的允許陳述式， AWS 不會授權請求。如果有明確的允許陳述式，或身分驗證類型為 NONE，則程式碼會繼續。
  - 驗證 服務的身分驗證政策是否允許請求。如果已啟用身分驗證政策，但沒有明確的允許陳述式， AWS 不會授權請求。如果有明確的允許陳述式，或身分驗證類型為 NONE，則強制執行程式碼會傳回允許的最終決策。
  - 任何政策中的明確拒絕會覆寫任何允許。

圖表顯示授權工作流程。提出請求時，相關政策會允許或拒絕對指定服務的請求存取。



使用安全群組控制 VPC Lattice 中的流量

AWS 安全群組充當虛擬防火牆，控制往返與其相關聯之實體的網路流量。使用 VPC Lattice，您可以建立安全群組並將其指派給 VPC 關聯，以將 VPC 連線至服務網路，為您的服務網路強制執行額外的網路層級安全保護。如果您使用 VPC 端點將 VPC 連線至服務網路，您也可以將安全群組指派給 VPC 端點。同樣地，您可以將安全群組指派給您建立的資源閘道，以啟用對 VPC 中資源的存取。

目錄

受管字首清單

VPC Lattice 提供受管字首清單，其中包含當您使用服務網路關聯將 VPC 連線至使用 VPC 關聯的服務網路時，用於透過 VPC Lattice 網路路由流量的 IP 地址。這些 IPs 可以是私有連結本機 IPs或無法路由的公IPs。

您可以在安全群組規則中參考 VPC Lattice 受管字首清單。這可讓流量從用戶端流經 VPC Lattice 服務網路，以及流向 VPC Lattice 服務目標。

例如，假設您的 EC2 執行個體已在美國西部（奧勒岡）區域 () 註冊為目標`us-west-2`。您可以將規則新增至執行個體安全群組，以允許從 VPC Lattice 受管字首清單傳入 HTTPS 存取，讓此區域中的 VPC Lattice 流量可以到達執行個體。如果您從安全群組移除所有其他傳入規則，您可以防止 VPC Lattice 流量以外的任何流量到達執行個體。

VPC Lattice 的受管字首清單名稱如下所示：

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

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的 AWS受管字首清單。

Windows 和 macOS 用戶端

VPC Lattice 字首清單中的地址是連結本機地址和不可路由的公有地址。如果您從這些用戶端連線至 VPC Lattice，則必須更新其組態，以便將受管字首清單中的 IP 地址轉送到用戶端的主要 IP 地址。以下是更新 Windows 用戶端組態的範例命令，其中 169.254.171.0是受管字首清單中的其中一個地址。

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

以下是更新 macOS 用戶端組態的範例命令，其中 169.254.171.0是受管字首清單中的其中一個地址。

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

為了避免建立靜態路由，建議您在 VPC 中使用服務網路端點來建立連線。如需詳細資訊，請參閱the section called “管理服務網路 VPC 端點關聯”。

安全群組規則

搭配或不搭配安全群組使用 VPC Lattice 不會影響您現有的 VPC 安全群組組態。不過，您可以隨時新增自己的安全群組。

關鍵考量

- 用戶端的安全群組規則會控制 VPC Lattice 的傳出流量。
- 目標的安全群組規則會控制從 VPC Lattice 到目標的傳入流量，包括運作狀態檢查流量。
- 服務網路與 VPC 之間關聯的安全群組規則會控制哪些用戶端可以存取 VPC Lattice 服務網路。
- 資源閘道的安全群組規則會控制從資源閘道到資源的傳出流量。

從資源閘道流向資料庫資源的流量的建議傳出規則

若要讓流量從資源閘道流向資源，您必須為開放連接埠建立傳出規則，並為資源建立接受的接聽程式通訊協定。

目標	通訊協定	連接埠範圍	Comment
### CIDR ##	*TCP*	*3306*	允許從資源閘道到資料庫的流量

服務網路和 VPC 關聯的建議傳入規則

若要讓流量從用戶端 VPCs 流向與服務網路相關聯的服務，您必須為服務的接聽程式連接埠和接聽程式通訊協定建立傳入規則。

來源	通訊協定	連接埠範圍	Comment
VPC CIDR	*listener*	*listener*	允許從用戶端到 VPC Lattice 的流量

建議從用戶端執行個體流向 VPC Lattice 的流量傳出規則

根據預設，安全群組允許所有對外流量。不過，如果您有自訂傳出規則，則必須允許傳出流量到接聽程式連接埠和通訊協定的 VPC Lattice 字首，以便用戶端執行個體可以連線到與 VPC Lattice 服務網路相關聯的所有服務。您可以參考 VPC Lattice 字首清單的 ID，以允許此流量。

目標	通訊協定	連接埠範圍	Comment
VPC Lattice #### # ID	*listener*	*listener*	允許從用戶端到 VPC Lattice 的流量

從 VPC Lattice 流向目標執行個體之流量的建議傳入規則

您無法使用用戶端安全群組做為目標安全群組的來源，因為流量會從 VPC Lattice 流出。您可以參考 VPC Lattice 字首清單的 ID。

來源	通訊協定	連接埠範圍	Comment
VPC Lattice #### # ID	*target*	*target*	允許從 VPC Lattice 到目標的流量
VPC Lattice #### # ID	*health check*	*health check*	允許從 VPC Lattice 到目標的運作狀態檢查流量

管理 VPC 關聯的安全群組

您可以使用 AWS CLI 來檢視、新增或更新 VPC 上的安全群組，以服務網路關聯。使用 時 AWS CLI，請記住您的命令會在為設定檔 AWS 區域 設定的 中執行。如果您想在不同區域中執行命令，則可變更設定檔的預設區域，或搭配 `--region` 參數使用命令。

開始之前，請確認您已在與要新增至服務網路的 VPC 相同的 VPC 中建立安全群組。如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的使用安全群組控制資源的流量

使用主控台建立 VPC 關聯時新增安全群組

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務網路。
3. 選取服務網路的名稱以開啟其詳細資訊頁面。
4. 在 VPC 關聯索引標籤上，選擇建立 VPC 關聯，然後選擇新增 VPC 關聯。
5. 選取 VPC 和最多五個安全群組。

6. 選擇儲存變更。

使用主控台新增或更新現有 VPC 關聯的安全群組

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 在導覽窗格的 VPC Lattice 下，選擇服務網路。
3. 選取服務網路的名稱以開啟其詳細資訊頁面。
4. 在 VPC 關聯索引標籤上，選取關聯的核取方塊，然後選擇動作、編輯安全群組。
5. 視需要新增和移除安全群組。
6. 選擇儲存變更。

使用 新增 VPC 關聯時新增安全群組 AWS CLI

使用 create-service-network-vpc-association 命令，指定 VPC 關聯的 VPC ID 和要新增的安全群組 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，此命令傳回的輸出會類似如下。

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

使用 新增或更新現有 VPC 關聯的安全群組 AWS CLI

使用 update-service-network-vpc-association 命令，指定服務網路的 ID 和安全群組IDs。這些安全群組會覆寫任何先前相關聯的安全群組。更新清單時至少定義一個安全群組。

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

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

> **Warning**
>
> 您無法移除所有安全群組。反之，您必須先刪除 VPC 關聯，然後重新建立不含任何安全群組的 VPC 關聯。刪除 VPC 關聯時請小心。這可防止流量到達該服務網路中的服務。

使用網路 ACLs流量

網路存取控制清單 (ACL) 會允許或拒絕子網層級的特定傳入或傳出流量。預設的網路 ACL 會允許所有外傳和傳入流量。您可以為子網路建立自訂網路 ACLs，以提供額外的安全層。如需詳細資訊，請參閱「Amazon VPC 使用者指南」中的網路 ACL。

目錄

用戶端子網路的網路 ACLs

用戶端子網路的網路 ACLs 必須允許用戶端和 VPC Lattice 之間的流量。您可以從 VPC Lattice 的受管字首清單中，取得要允許的 IP 地址範圍。

以下是傳入規則的範例。

來源	通訊協定	連接埠範圍	Comment
vpc_lattice_cidr_block	TCP	1025-65535	允許從 VPC Lattice 到用戶端的流量

以下是傳出規則範例。

目標	通訊協定	連接埠範圍	Comment
vpc_lattice_cidr_block	*listener*	*listener*	允許從用戶端到 VPC Lattice 的流量

目標子網路的網路 ACLs

目標子網路的網路 ACLs 必須允許目標連接埠和運作狀態檢查連接埠上目標和 VPC Lattice 之間的流量。您可以從 VPC Lattice 的受管字首清單中取得要允許的 IP 地址範圍。

以下是傳入規則的範例。

來源	通訊協定	連接埠範圍	Comment
vpc_lattice_cidr_block	*target*	*target*	允許從 VPC Lattice 到目標的流量
vpc_lattice_cidr_block	*health check*	*health check*	允許從 VPC Lattice 到目標的運作狀態檢查流量

以下是傳出規則範例。

目標	通訊協定	連接埠範圍	Comment
vpc_lattice_cidr_block	*target*	1024-65535	允許從目標到 VPC Lattice 的流量
vpc_lattice_cidr_block	*health check*	1024-65535	允許從目標到 VPC Lattice 的運作狀態檢查流量

Amazon VPC Lattice 的 SIGv4 驗證請求

VPC Lattice 使用 Signature 第 4 版 (SIGv4) 或 Signature 第 4A 版 (SIGv4A) 進行用戶端身分驗證。如需詳細資訊，請參閱《IAM 使用者指南》中的 API 請求的AWS 第 4 版簽署程序。

考量事項

- VPC Lattice 會嘗試驗證使用 SIGv4 或 SIGv4A 簽署的任何請求。請求在沒有身分驗證的情況下失敗。

- VPC Lattice 不支援承載簽署。您必須傳送 `x-amz-content-sha256`標頭，並將 值設為 `"UNSIGNED-PAYLOAD"`。

範例

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

Python

此範例會透過安全連線將已簽署的請求傳送至網路中註冊的服務。如果您偏好使用請求，botocore 套件可簡化身分驗證程序，但並非嚴格要求。如需詳細資訊，請參閱 Boto3 文件中的登入資料。

若要安裝 `botocore`和 `awscrt`套件，請使用下列命令。如需詳細資訊，請參閱 AWS CRT Python。

```
pip install botocore awscrt
```

如果您在 Lambda 上執行用戶端應用程式，請使用 Lambda 層安裝所需的模組，或將其包含在部署套件中。

在下列範例中，將預留位置值取代為您自己的值。

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

此範例說明如何使用自訂攔截器執行請求簽署。它使用來自 的預設登入資料提供者類別AWS SDK for Java 2.x，其會為您取得正確的登入資料。如果您想要使用特定的登入資料提供者，您可以從 中選取一個登入資料提供者AWS SDK for Java 2.x。 適用於 Java 的 AWS SDK 僅允許透過 HTTPS 的未簽署承載。不過，您可以擴展簽署者，以支援透過 HTTP 的未簽署承載。

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

此範例需要對 的額外相依性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

此範例使用 aws-crt NodeJS 繫結來傳送使用 HTTPS 簽署的請求。

若要安裝aws-crt套件，請使用下列命令。

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

如果AWS_REGION環境變數存在，則範例會使用 指定的區域AWS_REGION。預設區域為 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

此範例使用適用於 Go 的 Smithy 程式碼產生器和AWS 適用於 Go 程式設計語言的 SDK 來處理請求簽署請求。此範例需要 1.21 或更新版本的 Go 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

此範例使用適用於 AWS Go 程式設計語言的 開發套件來處理 GRPC 請求的請求簽署。這可與來自 GRPC 範例程式碼儲存庫的 echo 伺服器搭配使用。

```
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 中的資料保護

AWS 共同責任模型適用於 Amazon VPC Lattice 中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。此內容包括您所使用 AWS 服務 的安全組態和管理任務。如需有關資料隱私權的更多相關資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱AWS 安全性部落格上的AWS 共同責任模型和 GDPR 部落格文章。

傳輸中加密

VPC Lattice 是由控制平面和資料平面組成的全受管服務。每個平面在服務中都有不同的用途。控制平面提供用於建立、讀取/描述、更新、刪除和列出 (CRUDL) 資源APIs （ 例如 `CreateService`和 `UpdateService`)。與 VPC Lattice 控制平面的通訊由 TLS 保護傳輸中。資料平面是 VPC Lattice 調用 API，可提供 服務之間的互連。當您使用 HTTPS 或 TLS 時，TLS 會加密與 VPC Lattice 資料平面的通訊。密碼套件和通訊協定版本使用 VPC Lattice 提供的預設值，且無法設定。如需詳細資訊，請參閱VPC Lattice 服務的 HTTPS 接聽程式。

靜態加密

根據預設，靜態資料的加密有助於降低保護敏感資料時涉及的操作開銷和複雜性。同時，其可讓您建置符合嚴格加密合規性和法規要求的安全應用程式。

目錄

使用 Amazon S3 受管金鑰 (SSE-S3) 的伺服器端加密

使用伺服器端加密搭配 Amazon S3 受管金鑰 (SSE-S3) 時，每個物件都會使用唯一金鑰來加密。作為額外的保護，我們使用我們定期輪換的根金鑰來加密金鑰本身。Amazon S3 伺服器端加密使用目前最強大的其中一種區塊加密法 (256 位元進階加密標準 (AES-256) GCM)，加密您的資料。對於在 AES-GCM 之前加密的物件，仍支援以 AES-CBC 解密這些物件。如需詳細資訊，請參閱搭配使用伺服器端加密與 Amazon S3-managed加密金鑰 (SSE-S3)。

如果您為 VPC Lattice 存取日誌的 S3 儲存貯體使用 Amazon S3-managed加密金鑰 (SSE-S3) 啟用伺服器端加密，我們會在每個存取日誌檔案存放在 S3 儲存貯體之前自動加密。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的傳送至 Amazon S3 的日誌。 Amazon CloudWatch

伺服器端加密與存放在 AWS KMS (SSE-KMS) 中的 AWS KMS 金鑰

具有 AWS KMS 金鑰的伺服器端加密 (SSE-KMS) 類似於 SSE-S3，但使用此服務可獲得額外的好處和費用。 AWS KMS 金鑰有個別的許可，可為 Amazon S3 中的物件提供額外的保護，防止未經授權的存取。SSE-KMS 也為您提供稽核線索，顯示您的 AWS KMS 金鑰何時被使用以及由誰使用。如需詳細資訊，請參閱搭配 AWS Key Management Service (SSE-KMS) 使用伺服器端加密。

目錄

加密和解密憑證的私有金鑰

您的 ACM 憑證和私有金鑰會使用別名為 aws/acm 的 AWS 受管 KMS 金鑰進行加密。您可以在 AWS KMS 主控台的AWS 受管金鑰下，檢視具有此別名的金鑰 ID。

VPC Lattice 不會直接存取您的 ACM 資源。它使用 AWS TLS Connection Manager 來保護和存取憑證的私有金鑰。當您使用 ACM 憑證建立 VPC Lattice 服務時，VPC Lattice 會將您的憑證與 AWS TLS Connection Manager 建立關聯。方法是 AWS KMS 針對 AWS 具有字首 aws/acm 的受管金鑰在 中建立授予。授權是一種政策工具，允許 TLS Connection Manager 使用密碼編譯操作中的 KMS 金鑰。授權可讓承授者主體 (TLS Connection Manager) 呼叫 KMS 金鑰上指定的授予操作，以解密憑證的私有金鑰。TLS Connection Manager 接著會使用憑證和解密的 （純文字） 私有金鑰，與 VPC Lattice 服務的用戶端建立安全連線 (SSL/TLS 工作階段）。當憑證與 VPC Lattice 服務取消關聯時，授權即會淘汰。

如果您想要移除 KMS 金鑰的存取權，建議您使用 或 中的 AWS 管理主控台 `update-service`命令，從服務取代或刪除憑證 AWS CLI。

VPC Lattice 的加密內容

加密內容是選用的一組金鑰/值對，其中包含有關私有金鑰可能用於哪些用途的內容資訊。 會將加密內容 AWS KMS 合併為加密資料，並將其用作額外的已驗證資料，以支援已驗證的加密。

當您的 TLS 金鑰與 VPC Lattice 和 TLS Connection Manager 搭配使用時，您的 VPC Lattice 服務名稱會包含在用來加密靜態金鑰的加密內容中。您可以檢視 CloudTrail 日誌中的加密內容，如下一節所示，或查看 ACM 主控台中的關聯資源索引標籤，來驗證憑證和私有金鑰所使用的 VPC Lattice 服務。

若要解密資料，則必須在請求中包含相同的加密內容。VPC Lattice 在所有 AWS KMS 密碼編譯操作中使用相同的加密內容，其中金鑰為 ，`aws:vpc-lattice:arn`而值為 VPC Lattice 服務的 Amazon Resource Name (ARN)。

下列範例展示操作輸出中的加密內容，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"
}
```

監控 VPC Lattice 的加密金鑰

當您搭配 VPC Lattice 服務使用 AWS 受管金鑰時，您可以使用 AWS CloudTrail 來追蹤 VPC Lattice 傳送的請求 AWS KMS。

CreateGrant

當您將 ACM 憑證新增至 VPC Lattice 服務時，系統會代表您傳送CreateGrant請求，讓 TLS Connection Manager 能夠解密與您的 ACM 憑證相關聯的私有金鑰

您可以在 CloudTrail、事件歷史記錄、CreateGrant 中將CreateGrant操作檢視為事件。

以下是 CreateGrant CloudTrail 操作事件歷史記錄中的範例事件記錄。

```
{
    "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"
}
```

在上述CreateGrant範例中，承授者主體是 TLS Connection Manager，而加密內容具有 VPC Lattice 服務 ARN。

ListGrants

您可以使用 KMS 金鑰 ID 和帳戶 ID 來呼叫 ListGrants API。這可讓您取得指定 KMS 金鑰的所有授予清單。如需詳細資訊，請參閱 ListGrants。

在 中使用下列ListGrants命令 AWS CLI ，以查看所有授予的詳細資訊。

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

以下為範例輸出。

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

在上述`ListGrants`範例中，承授者主體是 TLS Connection Manager，加密內容具有 VPC Lattice 服務 ARN。

解密

VPC Lattice 使用 TLS Connection Manager 呼叫 `Decrypt`操作來解密您的私有金鑰，以便在 VPC Lattice 服務中提供 TLS 連線。您可以在 CloudTrail 事件歷史記錄 Decrypt 中將`Decrypt`操作檢視為事件。

以下是 `Decrypt` CloudTrail 操作事件歷史記錄中的範例事件記錄。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "type": "AWSService",
        "invokedBy": "tlsconnectionmanager.amazonaws.com"
    },
    "eventTime": "2023-02-07T00:07:23Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "tlsconnectionmanager.amazonaws.com",
    "userAgent": "tlsconnectionmanager.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:acm:arn": "arn:aws:acm:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "aws:vpc-lattice:arn": "arn:aws:vpc-lattice:us-west-2:111122223333:service/
svc-0b23c1234567890ab"
        },
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "sharedEventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "eventCategory": "Management"
}
```


Amazon VPC Lattice 的身分和存取管理

下列各節說明如何透過控制可執行 VPC Lattice API 動作的人員，使用 AWS Identity and Access Management (IAM) 協助保護 VPC Lattice 資源。

主題

- Amazon VPC Lattice 如何與 IAM 搭配使用
- Amazon VPC Lattice API 許可
- Amazon VPC Lattice 的身分型政策
- 使用 Amazon VPC Lattice 的服務連結角色
- AWS Amazon VPC Lattice 的 受管政策

Amazon VPC Lattice 如何與 IAM 搭配使用

在您使用 IAM 管理對 VPC Lattice 的存取之前，請先了解哪些 IAM 功能可與 VPC Lattice 搭配使用。

IAM 功能	VPC Lattice 支援
身分型政策	是
資源型政策	是
政策動作	是
政策資源	是
政策條件索引鍵	是
ACL	否
ABAC (政策中的標籤)	是
臨時憑證	是
服務角色	否
服務連結角色	是

如需 VPC Lattice 和其他 AWS 服務如何與大多數 IAM 功能搭配使用的高階檢視，請參閱《IAM 使用者指南》中的AWS 與 IAM 搭配使用的 服務。

VPC Lattice 的身分型政策

支援身分型政策：是

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。如需了解如何建立身分型政策，請參閱《IAM 使用者指南》中的透過客戶管理政策定義自訂 IAM 許可。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素參考。

VPC Lattice 內的資源型政策

支援資源型政策：是

以資源為基礎的政策是您連接到 中資源的 JSON 政策文件 AWS。在支援資源型政策的 AWS 服務中，服務管理員可以使用它們來控制對該 AWS 服務特定資源的存取。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。

VPC Lattice 支援身分驗證政策，這是一種以資源為基礎的政策，可讓您控制對服務網路中服務的存取。如需詳細資訊，請參閱使用身分驗證政策控制對 VPC Lattice 服務的存取。

VPC Lattice 也支援與 整合的資源型許可政策 AWS Resource Access Manager。您可以使用這些以資源為基礎的政策，授予許可來管理與其他 AWS 帳戶或組織的連線，以用於 服務、資源組態和服務網路。如需詳細資訊，請參閱分享您的 VPC Lattice 實體。

VPC Lattice 的政策動作

支援政策動作：是

在 IAM 政策陳述式中，您可以從任何支援 IAM 的服務指定任何 API 動作。對於 VPC Lattice，請使用下列字首搭配 API 動作的名稱：`vpc-lattice:`。例如：`vpc-lattice:CreateService`、`vpc-lattice:CreateTargetGroup` 和 `vpc-lattice:PutAuthPolicy`。

若要在單一陳述式中指定多個動作，請用逗號分隔，如下所示：

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

您也可以使用萬用字元指定多個動作。例如，您可以指定名稱開頭為 `Get` 文字的所有動作，如下所示：

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

如需 VPC Lattice API 動作的完整清單，請參閱《服務授權參考》中的 Amazon VPC Lattice 定義的動作。

VPC Lattice 的政策資源

支援政策資源：是

在 IAM 政策陳述式中，Resource 元素指定陳述式所涵蓋的一個或多個物件。對於 VPC Lattice，每個 IAM 政策陳述式都適用於您使用其 ARNs指定的資源。

特定的 Amazon Resource Name (ARN) 格式取決於資源。當您提供 ARN 時，請將*##*文字取代為您的資源特定資訊。

- 存取日誌訂閱：

```
"Resource": "arn:aws:vpc-lattice:region:account-id:accesslogsubscription/access-log-subscription-id"
```

- 接聽程式：

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

- 資源閘道

```
"Resource": "arn:aws:vpc-lattice:region:account-id:resourcegateway/resource-gateway-id"
```

- 資源組態

```
"Resource": "arn:aws:vpc-lattice:region:account-id:resourceconfiguration/resource-configuration-id"
```

- 規則：

```
"Resource": "arn:aws:vpc-lattice:region:account-id:service/service-id/
listener/listener-id/rule/rule-id"
```

- 服務：

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

- 服務網路：

```
"Resource": "arn:aws:vpc-lattice:region:account-id:servicenetwork/service-network-id"
```

- 服務網路服務關聯：

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkserviceassociation/service-network-service-association-id"
```

- 服務網路資源組態關聯

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkresourceassociation/service-network-resource-association-id"
```

- 服務網路 VPC 關聯：

```
"Resource": "arn:aws:vpc-lattice:region:account-
id:servicenetworkvpcassociation/service-network-vpc-association-id"
```

- 目標群組：

```
"Resource": "arn:aws:vpc-lattice:region:account-id:targetgroup/target-group-id"
```

VPC Lattice 的政策條件索引鍵

支援服務特定政策條件金鑰：是

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

`Condition` 元素會根據定義的條件，指定陳述式的執行時機。您可以建立使用條件運算子的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。若要查看所有 AWS 全域條件索引鍵，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

若要查看 VPC Lattice 條件索引鍵的清單，請參閱《服務授權參考》中的 Amazon VPC Lattice 的條件索引鍵。

AWS 支援全域條件金鑰和服務特定的條件金鑰。如需 AWS 全域條件索引鍵的資訊，請參閱《IAM 使用者指南》中的AWS 全域條件內容索引鍵。

VPC Lattice 中的存取控制清單 (ACLs)

支援 ACL：否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

具有 VPC Lattice 的屬性型存取控制 (ABAC)

支援 ABAC (政策中的標籤)：是

屬性型存取控制 (ABAC) 是一種授權策略，依據稱為標籤的屬性來定義許可。您可以將標籤連接至 IAM 實體 AWS 和資源，然後設計 ABAC 政策，以便在委託人的標籤符合資源上的標籤時允許操作。

如需根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的條件元素中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的使用 ABAC 授權定義許可。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的使用屬性型存取控制 (ABAC)。

搭配 VPC Lattice 使用臨時登入資料

支援臨時憑證：是

臨時登入資料提供對 AWS 資源的短期存取，並在您使用聯合或切換角色時自動建立。 AWS 建議您動態產生臨時登入資料，而不是使用長期存取金鑰。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM 中的臨時安全憑證與可與 IAM 搭配運作的AWS 服務。

VPC Lattice 的服務角色

支援服務角色：否

服務角色是服務擔任的 IAM 角色，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的建立角色以委派許可給 AWS 服務。

> **Warning**
>
> 變更服務角色的許可可能會中斷 VPC Lattice 功能。只有在 VPC Lattice 提供指引時，才能編輯服務角色。

VPC Lattice 的服務連結角色

支援服務連結角色：是

服務連結角色是連結至 的一種服務角色 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理 VPC Lattice 服務連結角色的資訊，請參閱 使用 Amazon VPC Lattice 的服務連結角色。

Amazon VPC Lattice API 許可

您必須授予 IAM 身分 （例如使用者或角色） 許可來呼叫他們所需的 VPC Lattice API 動作，如中所述VPC Lattice 的政策動作。此外，對於某些 VPC Lattice 動作，您必須授予 IAM 身分從其他 AWS APIs許可。

API 的必要許可

從 API 呼叫下列動作時，您必須授予 IAM 使用者呼叫指定動作的許可。

`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（只有在提供安全群組時才需要）

UpdateServiceNetworkVpcAssociation

- vpc-lattice:UpdateServiceNetworkVpcAssociation
- ec2:DescribeSecurityGroups（只有在提供安全群組時才需要）

CreateTargetGroup

- vpc-lattice:CreateTargetGroup

- ec2:DescribeVpcs

RegisterTargets

- vpc-lattice:RegisterTargets
- ec2:DescribeInstances（只有在 INSTANCE是目標群組類型時才需要）
- ec2:DescribeVpcs（只有在 INSTANCE或 IP為目標群組類型時才需要）
- ec2:DescribeSubnets（只有在 INSTANCE或 IP為目標群組類型時才需要）
- lambda:GetFunction（只有在 LAMBDA是目標群組類型時才需要）
- lambda:AddPermission（只有在目標群組尚未擁有叫用指定 Lambda 函數的許可時，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

Amazon VPC Lattice 的身分型政策

根據預設，使用者和角色沒有建立或修改 VPC Lattice 資源的許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。

如需了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的建立 IAM 政策 (主控台)。

如需 VPC Lattice 定義的動作和資源類型的詳細資訊，包括每種資源類型的 ARNs 格式，請參閱《服務授權參考》中的 Amazon VPC Lattice 的動作、資源和條件金鑰。

目錄

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 VPC Lattice 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需更多資訊，請參閱《IAM 使用者指南》中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱《IAM 使用者指南》中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 等使用服務動作 AWS 服務，您也可以使用條件來授予其存取權 CloudFormation。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的使用 IAM Access Analyzer 驗證政策。
- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需詳細資訊，請參閱《IAM 使用者指南》中的透過 MFA 的安全 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱《IAM 使用者指南》中的 IAM 安全最佳實務。

完整存取的其他必要許可

若要使用與 VPC Lattice 整合 AWS 的其他服務，以及整個 VPC Lattice 功能套件，您必須擁有特定的額外許可。由於混淆代理人權限提升風險，這些許可不包含在`VPCLatticeFullAccess`受管政策中。

您必須將下列政策連接至您的角色，並將其與 VPCLatticeFullAccess受管政策搭配使用。

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*"
        }
    ]
}
```

此政策提供下列額外許可：

- `iam:AttachRolePolicy`：可讓您將指定的受管政策連接至指定的 IAM 角色。
- `iam:PutRolePolicy`：可讓您新增或更新內嵌在指定 IAM 角色中的內嵌政策文件。
- `s3:PutBucketPolicy`：可讓您將儲存貯體政策套用至 Amazon S3 儲存貯體。
- `firehose:TagDeliveryStream`：可讓您新增或更新 Firehose 交付串流的標籤。

VPC Lattice 的身分型政策範例

主題

- 範例政策：管理與服務網路的 VPC 關聯
- 範例政策：建立與服務網路的服務關聯
- 範例政策：將標籤新增至資源
- 範例政策：建立服務連結角色

範例政策：管理與服務網路的 VPC 關聯

以下範例示範一個政策，該政策為具有此政策的使用者提供許可，以建立、更新和刪除與服務網路的 VPC 關聯，但僅適用於 條件中指定的 VPC 和服務網路。如需指定條件索引鍵的詳細資訊，請參閱 VPC Lattice 的政策條件索引鍵。

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"
               }
            }
         }
      ]
}
```

範例政策：建立與服務網路的服務關聯

如果您不是使用條件金鑰來控制對 VPC Lattice 資源的存取，您可以在 Resource元素中指定資源 ARNs，以改為控制存取。

下列範例示範一個政策，透過指定可與 CreateServiceNetworkServiceAssociation API 動作搭配使用的服務和服務網路的 ARNs，將服務關聯限制為使用者可建立的服務網路。如需指定 ARN 值的詳細資訊，請參閱 VPC Lattice 的政策資源。

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"
         ]
      }
   ]
}
```

範例政策：將標籤新增至資源

以下範例示範一個政策，該政策授予具有此政策的使用者在 VPC Lattice 資源上建立標籤的許可。

JSON

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

範例政策：建立服務連結角色

VPC Lattice 需要許可，才能在 中的任何使用者第一次建立 VPC Lattice 資源時 AWS 帳戶 建立服務連結角色。如果服務連結角色尚未存在，VPC Lattice 會在您的帳戶中建立該角色。服務連結角色會授予 VPC Lattice 的許可，使其可以 AWS 服務 代表您呼叫其他 。如需詳細資訊，請參閱the section called “使用服務連結角色”。

為能成功自動建立該角色，使用者必須已獲許可執行 `iam:CreateServiceLinkedRole` 動作。

```
"Action": "iam:CreateServiceLinkedRole"
```

以下範例示範一個政策，該政策授予具有此政策的使用者為 VPC Lattice 建立服務連結角色的許可。

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"
            }
         }
      }
   ]
}
```

如需詳細資訊，請參閱 IAM 使用者指南中的服務連結角色許可。

使用 Amazon VPC Lattice 的服務連結角色

Amazon VPC Lattice 會將服務連結角色用於 AWS 服務 代表您呼叫其他 所需的許可。如需詳細資訊，請參閱「IAM 使用者指南」中的服務連結角色。

VPC Lattice 使用名為 的服務連結角色AWSServiceRoleForVpcLattice。

VPC Lattice 的服務連結角色許可

AWSServiceRoleForVpcLattice 服務連結角色信任下列服務擔任該角色：

- `vpc-lattice.amazonaws.com`

名為 的角色許可政策AWSVpcLatticeServiceRolePolicy允許 VPC Lattice 在`AWS/VpcLattice`命名空間中發佈 CloudWatch 指標。如需詳細資訊，請參閱《 AWS 受管政策參考AWSVpcLatticeServiceRolePolicy》中的 。

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱the section called “範例政策：建立服務連結角色”。

為 VPC Lattice 建立服務連結角色

您不需要手動建立服務連結角色，當您在 AWS 管理主控台、 AWS CLI或 AWS API 中建立 VPC Lattice 資源時，VPC Lattice 會為您建立服務連結角色。

若您刪除此服務連結角色，之後需要再次建立，您可以在帳戶中使用相同程序重新建立角色。當您建立 VPC Lattice 資源時，VPC Lattice 會再次為您建立服務連結角色。

編輯 VPC Lattice 的服務連結角色

您可以使用 IAM 編輯 AWSServiceRoleForVpcLattice 的描述。如需詳細資訊，請參閱 IAM 使用者指南中的編輯服務連結角色描述。

刪除 VPC Lattice 的服務連結角色

如果您不再需要使用 Amazon VPC Lattice，我們建議您刪除 AWSServiceRoleForVpcLattice。

只有在刪除 中的所有 VPC Lattice 資源後，才能刪除此服務連結角色 AWS 帳戶。

使用 IAM 主控台 AWS CLI、 或 AWS API 來刪除 AWSServiceRoleForVpcLattice 服務連結角色。如需詳細資訊，請參閱《IAM 使用者指南》中的刪除服務連結角色。

刪除服務連結角色之後，當您在 中建立 VPC Lattice 資源時，VPC Lattice 會再次建立角色 AWS 帳戶。

VPC Lattice 服務連結角色支援的 區域

VPC Lattice 支援在提供服務的所有區域中使用服務連結角色。

AWS Amazon VPC Lattice 的 受管政策

AWS 受管政策是由 AWS AWS 受管政策建立和管理的獨立政策旨在為許多常用案例提供許可，以便您可以開始將許可指派給使用者、群組和角色。

請記住， AWS 受管政策可能不會授予特定使用案例的最低權限許可，因為這些許可可供所有 AWS 客戶使用。我們建議您定義特定於使用案例的客戶管理政策，以便進一步減少許可。

您無法變更 AWS 受管政策中定義的許可。如果 AWS 更新受 AWS 管政策中定義的許可，則更新會影響政策連接的所有委託人身分（使用者、群組和角色）。 AWS 服務 當新的 啟動或新的 API 操作可用於現有服務時， AWS 最有可能更新 AWS 受管政策。

如需詳細資訊，請參閱 IAM 使用者指南中的 AWS 受管政策。

AWS 受管政策：VPCLatticeFullAccess

此政策提供 Amazon VPC Lattice 的完整存取權，以及對其他相依服務的有限存取權。它包含執行下列動作的許可：

- ACM – 擷取自訂網域名稱的 SSL/TLS 憑證 ARN。
- CloudWatch – 檢視存取日誌和監控資料。
- CloudWatch Logs – 設定存取日誌並將其傳送至 CloudWatch Logs。
- Amazon EC2 – 設定網路介面並擷取 EC2 執行個體和 VPCs的相關資訊。這可用於建立資源組態、資源閘道和目標群組、設定 VPC Lattice 實體關聯，以及註冊目標。
- Elastic Load Balancing – 擷取 Application Load Balancer 的相關資訊，將其註冊為目標。
- Firehose – 擷取用於存放存取日誌之交付串流的相關資訊。
- Lambda – 擷取 Lambda 函數的相關資訊，將其註冊為目標。
- Amazon RDS – 擷取 RDS 叢集和執行個體的相關資訊。
- Amazon S3 – 擷取用於存放存取日誌之 S3 儲存貯體的相關資訊。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 VPCLatticeFullAccess。

若要使用與 VPC Lattice 整合 AWS 的其他服務，以及整個 VPC Lattice 功能套件，您必須擁有特定的額外許可。由於混淆代理人權限提升風險，這些許可不包含在`VPCLatticeFullAccess`受管政策中。如需詳細資訊，請參閱完整存取的其他必要許可。

AWS 受管政策：VPCLatticeReadOnlyAccess

此政策提供對 Amazon VPC Lattice 的唯讀存取權，以及對其他相依服務的有限存取權。它包含執行下列動作的許可：

- ACM – 擷取自訂網域名稱的 SSL/TLS 憑證 ARN。
- CloudWatch – 檢視存取日誌和監控資料。
- CloudWatch Logs – 檢視存取日誌訂閱的日誌交付資訊。
- Amazon EC2 – 擷取 EC2 執行個體和 VPCs 的相關資訊，以建立目標群組並註冊目標。
- Elastic Load Balancing – 擷取 Application Load Balancer 的相關資訊。
- Firehose – 擷取有關存取日誌交付的交付串流資訊。
- Lambda – 檢視 Lambda 函數的相關資訊。
- Amazon RDS – 擷取 RDS 叢集和執行個體的相關資訊。

- Amazon S3 – 擷取存取日誌交付的 S3 儲存貯體相關資訊。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 VPCLatticeReadOnlyAccess。

AWS 受管政策：VPCLatticeServicesInvokeAccess

此政策提供叫用 Amazon VPC Lattice 服務的存取權。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 VPCLatticeServicesInvokeAccess。

AWS 受管政策：AWSVpcLatticeServiceRolePolicy

此政策會連接到名為 AWSServiceRoleForVpcLattice 的服務連結角色，以允許 VPC Lattice 代表您執行動作。您無法將此政策連接至 IAM 實體。如需詳細資訊，請參閱使用 Amazon VPC Lattice 的服務連結角色。

若要檢視此政策的許可，請參閱《 AWS 受管政策參考》中的 AWSVpcLatticeServiceRolePolicy。

AWS 受管政策的 VPC Lattice 更新

檢視自此服務開始追蹤這些變更以來，VPC Lattice AWS 受管政策更新的詳細資訊。如需此頁面變更的自動提醒，請訂閱 VPC Lattice 使用者指南的 RSS 摘要。

變更	描述	Date
VPCLatticeFullAccess	VPC Lattice 新增唯讀許可來描述 Amazon RDS 叢集和執行個體。	2024 年 12 月 1 日
VPCLatticeReadOnlyAccess	VPC Lattice 新增唯讀許可來描述 Amazon RDS 叢集和執行個體。	2024 年 12 月 1 日
AWSVpcLatticeServiceRolePolicy	VPC Lattice 新增許可，以允許 VPC Lattice 建立申請者管理的網路介面。	2024 年 12 月 1 日
VPCLatticeFullAccess	VPC Lattice 新增了新的政策，以授予對 Amazon VPC Lattice 的完整存取權，以及對其他相依服務的有限存取權。	2023 年 3 月 31 日
VPCLatticeReadOnlyAccess	VPC Lattice 新增了新的政策，以授予對 Amazon VPC Lattice 的唯讀存取	2023 年 3 月 31 日

變更	描述	Date
	權，以及對其他相依服務的有限存取權。	
VPCLatticeServicesInvokeAccess	VPC Lattice 新增了新政策，以授予叫用 Amazon VPC Lattice 服務的存取權。	2023 年 3 月 31 日
AWSVpcLatticeServiceRolePolicy	VPC Lattice 會將許可新增至其服務連結角色，以允許 VPC Lattice 在`AWS/VpcLattice` 命名空間中發佈 CloudWatch 指標。此`AWSVpcLatticeServiceRolePolicy` 政策包含呼叫 CloudWatch PutMetricData API 動作的許可。如需詳細資訊，請參閱使用 Amazon VPC Lattice 的服務連結角色。	2022 年 12 月 5 日
VPC Lattice 已開始追蹤變更	VPC Lattice 開始追蹤其 AWS 受管政策的變更。	2022 年 12 月 5 日

Amazon VPC Lattice 的合規驗證

在多個合規計畫中，第三方稽核人員會評估 Amazon VPC Lattice 的安全性和 AWS 合規性。

若要了解 AWS 服務 是否在特定合規計劃範圍內，請參閱AWS 服務 合規計劃範圍內然後選擇您感興趣的合規計劃。如需一般資訊，請參閱AWS 合規計劃。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱下載報告 in AWS Artifact

您使用 時的合規責任 AWS 服務 取決於資料的機密性、您公司的合規目標，以及適用的法律和法規。如需使用 時合規責任的詳細資訊 AWS 服務，請參閱 AWS 安全文件。

使用介面端點存取 Amazon VPC Lattice (AWS PrivateLink)

您可以建立介面 VPC 端點，在 VPC 和 Amazon VPC Lattice 之間建立私有連線。介面端點採用 AWS PrivateLink技術，可讓您在沒有網際網路閘道、NAT 裝置、VPN 連線或 Direct Connect 連線的情況

下，私下存取 VPC Lattice APIs。VPC 中的執行個體不需要公有 IP 地址，即可與 VPC Lattice APIs通訊。

每個界面端點都由子網路中的一或多個網路界面表示。

介面 VPC 端點的考量事項

設定 VPC Lattice 的介面 VPC 端點之前，請務必檢閱 AWS PrivateLink 指南中的AWS 服務 透過 存取 AWS PrivateLink。

VPC Lattice 支援從您的 VPC 呼叫其所有 API 動作。

為 VPC Lattice 建立介面 VPC 端點

您可以使用 Amazon VPC 主控台或 AWS Command Line Interface () 為 VPC Lattice 服務建立 VPC 端點AWS CLI。如需詳細資訊，請參閱《 AWS PrivateLink 指南》中的建立介面 VPC 端點。

使用下列服務名稱建立 VPC Lattice 的 VPC 端點：

`com.amazonaws.region.vpc-lattice`

如果您為端點啟用私有 DNS，您可以使用區域的預設 DNS 名稱向 VPC Lattice 提出 API 請求，例如 `vpc-lattice.us-east-1.amazonaws.com`。

Amazon VPC Lattice 中的彈性

AWS 全球基礎設施是以 AWS 區域 和可用區域為基礎建置。

AWS 區域 提供多個實體分隔和隔離的可用區域，這些可用區域與低延遲、高輸送量和高備援聯網連接。

透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域 和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

Amazon VPC Lattice 中的基礎設施安全性

Amazon VPC Lattice 是受管服務，受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及如何 AWS 保護基礎設施的資訊，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 發佈的 API 呼叫，透過網路存取 VPC Lattice。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

監控 Amazon VPC Lattice

使用本節中的功能來監控 Amazon VPC Lattice 服務網路、服務、目標群組和 VPC 連線。

目錄

Amazon VPC Lattice 的 CloudWatch 指標

Amazon VPC Lattice 會將與目標群組和服務相關的資料傳送至 Amazon CloudWatch，並將其處理為可讀且幾近即時的指標。這些指標會保留 15 個月，讓您可以存取歷史資訊，並更清楚 Web 應用程式或服務的效能。您也可以設定警報監看特定閾值，在達到閾值發出通知或採取動作。如需更多資訊，請參閱 Amazon CloudWatch 使用者指南。

Amazon VPC Lattice 會在您的帳戶中使用服務連結角色 AWS ，將指標傳送至 Amazon CloudWatch。如需詳細資訊，請參閱使用 Amazon VPC Lattice 的服務連結角色。

目錄

檢視 Amazon CloudWatch 指標

您可以使用 Amazon CloudWatch CloudWatch 指標 AWS CLI。

使用 CloudWatch 主控台檢視指標

1. 在 https://console.aws.amazon.com/cloudwatch/ 開啟 Amazon CloudWatch 主控台。
2. 在導覽窗格中，選擇 指標。
3. 选择 `AWS/VpcLattice` 命名空间。
4. (選用) 若要檢視所有維度的指標，請在搜尋欄位中輸入其名稱。
5. (選用) 若要根據維度來篩選，請選取下列其中一項：

- 若要僅顯示目標群組報告的指標，請選擇目標群組。若要檢視單一目標群組的指標，請在搜尋欄位中輸入其名稱。
- 若要僅顯示針對您的服務報告的指標，請選擇服務。若要檢視單一服務的指標，請在搜尋欄位中輸入其名稱。

使用 檢視指標 AWS CLI

使用下列 CloudWatch list-metrics AWS CLI 命令來列出可用的指標：

```
aws cloudwatch list-metrics --namespace AWS/VpcLattice
```

如需每個指標及其維度的資訊，請參閱 目標群組指標和 服務指標。

目標群組指標

VPC Lattice 會自動將與目標群組相關的指標存放在 Amazon CloudWatch AWS/VpcLattice 命名空間中。 Amazon CloudWatch 如需目標群組的詳細資訊，請參閱 VPC Lattice 中的目標群組。

維度

若要篩選目標群組的指標，請使用下列維度：

- AvailabilityZone
- TargetGroup

指標	Description	TargetGroup 通訊協定
TotalConnectionCount	連線總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。	HTTP, HTTPS, TCP

指標	Description	TargetGroup 通訊協定
	統計資料 • 最有用的統計資料是 Sum。	
ActiveConnectionCount	作用中連線。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS, TCP
ConnectionErrorCount	連線失敗總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS, TCP

指標	Description	TargetGroup 通訊協定
HTTP1_ConnectionCount	HTTP/1.1 連線總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS
HTTP2_ConnectionCount	HTTP/2 連線總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS

指標	Description	TargetGroup 通訊協定
ConnectionTimeoutCount	連線連線逾時總計。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS, TCP
TotalReceivedConnectionBytes	收到的連線位元組總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS, TCP

指標	Description	TargetGroup 通訊協定
TotalSentConnectionBytes	傳送的連線位元組總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS, TCP
TotalRequestCount	請求總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS

指標	Description	TargetGroup 通訊協定
ActiveRequestCount	作用中請求總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS
RequestTime	請求到最後一個位元組的時間，以毫秒為單位。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Average和 pNN.NN（百分位數）。	HTTP, HTTPS

指標	Description	TargetGroup 通訊協定
HTTPCode_2XX_Count, HTTPCode_3XX_Count, HTTPCode_4XX_Count, HTTPCode_5XX_Count	彙總 HTTP 回應代碼。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS
TLSConnectionErrorCount	不包含失敗憑證驗證的 TLS 連線錯誤總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS, TCP

指標	Description	TargetGroup 通訊協定
TotalTLSConnectionHandshakeCount	TLS 連線交握成功總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。	HTTP, HTTPS, TCP

服務指標

VPC Lattice 會自動將與服務相關的指標存放在 Amazon CloudWatch AWS/VpcLattice 命名空間中。 Amazon CloudWatch 如需 服務的詳細資訊，請參閱 VPC Lattice 中的服務。

維度

若要篩選目標群組的指標，請使用下列維度：

- AvailabilityZone
- Service

指標	Description
RequestTimeoutCount	等待回應逾時的請求總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。

指標	Description
	報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。
TotalRequestCount	請求總數。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。
RequestTime	請求時間，以毫秒為單位。 報告準則 • 從資源接收流量開始，一律回報（無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Average和 pNN.NN（百分位數）。

指標	Description
HTTPCode_2XX_Count ，HTTPCode_3XX_Count ，HTTPCode_4XX_Count ，HTTPCode_5XX_Count	彙總 HTTP 回應代碼。 報告準則 • 從資源接收流量開始，一律回報 （無論是零或非零值）。 報告頻率 • 一分鐘一次。 統計資料 • 最有用的統計資料是 Sum。

Amazon VPC Lattice 的存取日誌

存取日誌會擷取 VPC Lattice 服務和資源組態的詳細資訊。您可以使用這些存取日誌來分析流量模式，並稽核網路中的所有服務。對於 VPC Lattice 服務，我們會發佈 VpcLatticeAccessLogs ，對於資源組態，我們會發佈VpcLatticeResourceAccessLogs需要單獨設定的 。

存取日誌是選用的，預設為停用。啟用存取日誌之後，您可以隨時停用它們。

定價

發佈存取日誌時需支付費用。 AWS 代表您原生發佈的日誌稱為付費日誌。如需已終止日誌定價的詳細資訊，請參閱 Amazon CloudWatch 定價、選擇日誌，以及檢視已終止日誌下的定價。

目錄

啟用存取日誌所需的 IAM 許可

若要啟用存取日誌並將日誌傳送到其目的地，您必須在政策中將下列動作連接到您正在使用的 IAM 使用者、群組或角色。

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": [
                "*"
            ]
        }
    ]
}
```

如需詳細資訊，請參閱《AWS Identity and Access Management 使用者指南》中的新增和移除 IAM 身分許可。

更新連接至您正在使用的 IAM 使用者、群組或角色的政策後，請前往 啟用存取日誌。

存取日誌目的地

您可以將存取日誌傳送至下列目的地。

Amazon CloudWatch Logs

- VPC Lattice 通常會在 2 分鐘內將日誌交付至 CloudWatch Logs。不過，請記住，實際日誌交付時間是盡最大努力的，可能會有額外的延遲。
- 如果日誌群組沒有特定許可，會自動建立資源政策並新增至 CloudWatch 日誌群組。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的傳送至 CloudWatch Logs 的日誌。 Amazon CloudWatch
- 您可以在 CloudWatch 主控台的日誌群組下找到傳送至 CloudWatch 的存取日誌。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的檢視傳送至 CloudWatch Logs 的日誌資料。 Amazon CloudWatch

Amazon S3

- VPC Lattice 通常會在 6 分鐘內將日誌交付至 Amazon S3。不過，請記住，實際日誌交付時間是盡最大努力的，可能會有額外的延遲。
- 如果儲存貯體沒有特定許可，系統會自動建立儲存貯體政策並新增至您的 Amazon S3 儲存貯體。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的傳送至 Amazon S3 的日誌。 Amazon CloudWatch
- 傳送至 Amazon S3 的存取日誌使用以下命名慣例：

```
[bucket]/[prefix]/AWSLogs/[accountId]/VpcLattice/AccessLogs/[region]/[YYYY/
MM/DD]/[resource-id]/[accountId]_VpcLatticeAccessLogs_[region]_[resource-
id]_YYYYMMDDTHHmmZ_[hash].json.gz
```

- 傳送至 Amazon S3 的 VpcLatticeResourceAccessLogs 使用以下命名慣例：

```
[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 通常會在 2 分鐘內將日誌交付至 Firehose。不過，請記住，實際日誌交付時間是盡最大努力的，可能會有額外的延遲。
- 服務連結角色會自動建立，以授予 VPC Lattice 傳送存取日誌的許可 Amazon Data Firehose。為了成功自動建立該角色，使用者必須有 `iam:CreateServiceLinkedRole` 動作的許可。如需詳細資訊，請參閱《Amazon CloudWatch 使用者指南》中的傳送至 的日誌 Amazon Data Firehose。

- 如需檢視傳送至 之日誌的詳細資訊 Amazon Data Firehose，請參閱《 Amazon Data Firehose 開發人員指南》中的監控 Amazon Kinesis Data Streams。

啟用存取日誌

完成下列程序來設定存取日誌，以擷取存取日誌並將其交付至您選擇的目的地。

目錄

使用主控台啟用存取日誌

您可以在建立期間啟用服務網路、服務或資源組態的存取日誌。您也可以在建立服務網路、服務或資源組態後啟用存取日誌，如下列程序所述。

使用主控台建立基本服務

1. 在 https://console.aws.amazon.com/vpc/ 開啟 Amazon VPC 主控台。
2. 選取服務網路、服務或資源組態。
3. 選擇動作、編輯日誌設定。
4. 開啟存取日誌切換開關。
5. 為您的存取日誌新增交付目的地，如下所示：
    - 選取 CloudWatch Log 群組，然後選擇日誌群組。若要建立日誌群組，請選擇在 CloudWatch 中建立日誌群組。
    - 選取 S3 儲存貯體並輸入 S3 儲存貯體路徑，包括任何字首。若要搜尋 S3 儲存貯體，請選擇瀏覽 S3。
    - 選取 Kinesis Data Firehose 交付串流，然後選擇交付串流。若要建立交付串流，請選擇在 Kinesis 中建立交付串流。
6. 選擇儲存變更。

使用 啟用存取日誌 AWS CLI

使用 CLI 命令 create-access-log-subscription 來啟用服務網路或服務的存取日誌。

請求追蹤

VPC Lattice 支援用戶端、目標和日誌之間的請求追蹤和關聯，以便使用 x-amzn-requestid 標頭進行可觀測性和偵錯。此標頭可由用戶端設定和傳送，或由 VPC Lattice 產生，並傳送至目標，也可用於存取日誌。

預設行為

- VPC Lattice 會自動為每個請求產生此標頭。
- 此值是隨機產生的識別符 （ 預設為 UUID 樣式 ）。
- 產生的識別符為：
  - 傳播到下游目標。
  - 在回應標頭中傳回給用戶端。
  - 已登入存取日誌

範例 （ 預設回應 ）

以下是傳送至用戶端的回應範例，其預設行為為 VPC Lattice 為 x-amzn-requestid 標頭產生隨機值。

```
{
    "HTTP/1.1 200 OK
    x-amzn-requestid: a9f2c7a1-6b4f-4c79-9e87-ff5a1234a001"
}
```

用戶端設定 值

- 用戶端可以選擇性地在傳入請求上設定此標頭，以覆寫自動產生的值。
- 考量事項
  - 標頭值不需要遵循 UUID 格式。
  - 如果標頭值超過 512 個位元組，VPC Lattice 會將其截斷為 512。

- 成功覆寫時，提供的標頭值將：
  - 出現在回應標頭中
  - 傳播到目標
  - 出現在存取日誌和指標中

範例（覆寫用戶端請求）

以下是用戶端以標頭值傳送的請求範例。

```
{
    "GET /my-service/endpoint HTTP/1.1
    Host: my-api.example.com
    x-amzn-requestid: trace-request-foobar"
}
```

範例（預設覆寫回應）

以下是使用覆寫值傳送至用戶端的回應範例。

```
{
    "HTTP/1.1 200 OK
    x-amzn-requestid: trace-request-foobar"
}
```

存取日誌內容

下表說明存取日誌項目的欄位。

欄位	Description	格式
`callerPrincipalTags`	請求中的 PrincipalTags。	JSON
`hostHeader`	請求的授權標頭。	string
`sslCipher`	用來建立用戶端 TLS 連線之一組密碼的 OpenSSL 名稱。	string
`serviceNetworkArn`	服務網路 ARN。	arn：aws：vpc-lattice：*region*：*account*：servicenetwork/*id*
`resolvedUser`	身分驗證啟用且身分驗證完成時的使用者 ARN。	null | ARN | "匿名" | "未知"
`authDeniedReason`	啟用身分驗證時拒絕存取的原因。	null | "Service" | "Network" | "Identity"

欄位	Description	格式
requestMethod	請求的方法標頭。	string
targetGroupArn	目標主機所屬的目標主機群組。	string
tlsVersion	TLS 版本。	TLSvx
userAgent	使用者代理程式標頭。	string
serverNameIndication	【僅限 HTTPS】 伺服器名稱指示 (SNI) 在 ssl 連線通訊端上設定的值。	string
destinationVpcId	目的地 VPC ID。	vpc-*xxxxxxxx*
sourceIpPort	來源的 IP 地址和 ：port。	*ip* : *port*
targetIpPort	目標的 IP 地址和連接埠。	*ip* : *port*
serviceArn	服務 ARN。	arn：aws：vpc-lattice：*region* : *account* : service/*id*
sourceVpcId	來源 VPC ID。	vpc-*xxxxxxxx*
requestPath	請求的路徑。	LatticePath？ : *path*
startTime	請求開始時間。	*YYYY*-*MM*-*DD* T*HH* : *MM* : *SS* Z
protocol	通訊協定。目前為 HTTP/1.1 或 HTTP/2。	string
responseCode	HTTP 回應代碼。只會記錄最終標頭的回應碼。如需詳細資訊，請參閱對存取日誌進行故障診斷。	integer
bytesReceived	收到的內文和標頭位元組。	integer

欄位	Description	格式
bytesSent	傳送的內文和標頭位元組。	integer
duration	從開始時間到最後一個位元組輸出的請求總持續時間，以毫秒為單位。	integer
requestToTargetDuration	從開始時間到傳送至目標的最後一個位元組的請求總持續時間，以毫秒為單位。	integer
responseFromTargetDuration	從目標主機讀取的第一個位元組到傳送至用戶端的最後一個位元組的請求總持續時間，以毫秒為單位。	integer
grpcResponseCode	gRPC 回應程式碼。如需詳細資訊，請參閱狀態碼及其在 gRPC 中的使用方式。只有在服務支援 gRPC 時，才會記錄此欄位。	integer
requestId	此唯一識別符會自動包含在回應中，做為 x-amzn-requestid 標頭的值。它可讓用戶端、目標和日誌之間的請求相互關聯，以實現可觀測性和偵錯。	string
callerPrincipal	已驗證的主體。	string
callerX509SubjectCN	主體名稱 (CN)。	string
callerX509IssuerOU	發行者 (OU)。	string
callerX509SANNameCN	發行者替代方案 （ 名稱/CN)。	string
callerX509SANDNS	主體替代名稱 (DNS)。	string

欄位	Description	格式
`callerX509SANURI`	主體替代名稱 (URI)。	string
`sourceVpcArn`	發出請求之 VPC 的 ARN。	arn：aws：ec2：*region*：*account*：vpc/*id*

欄位	Description	格式
failureReason	指出請求失敗的原因。可能值如下： • TargetConnectionError - 請求無法連線到目標群組中的目標。 • TargetProtocolError - 目標未回應有效資料。這可能表示目標具有無效的 TLS 記錄，或使用無效的目標群組通訊協定。 • TargetDataTimeout - 已達到閒置逾時。 • TargetConnectionClosed - 目標在完成回應之前關閉連線。 • ClientConnectionClosed - 用戶端在收到完整回應之前關閉連線。 • ClientRateLimited - 用戶端超過連線限制，VPC Lattice 限制速率。 • ClientAccessDenied - VPC Lattice 拒絕存取資源。如需 VPC Lattice 拒絕存取原因的詳細資訊authDeniedReason ，請使用 。 • ClientProtocolError - 用戶端傳送了無法理解的資料。這可能表示用戶端使用了無效的 TLS 記錄或無效的通訊協定。	string

欄位	Description	格式
	• ConnectionDuration Exceeded - 連線達到連線持續時間上限。 • InternalError - 處理請求時發生內部錯誤。	

範例

以下為日誌項目的範例。

```
{
    "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"
}
```

資源存取日誌內容

下表說明資源存取日誌項目的欄位。

欄位	Description	格式
serviceNetworkArn	服務網路 ARN。	arn : *partition* vpc-lattice : *region* : *account* : servicenetwork/*id*
serviceNetworkResourceAssociationId	服務網路資源 ID。	*snra-xxx*
vpcEndpointId	用來存取資源的端點 ID。	string
sourceVpcArn	來源 VPC ARN 或起始連線的 VPC。	string
resourceConfigurationArn	已存取之資源組態的 ARN。	string
protocol	用來與資源組態通訊的通訊協定。目前僅支援 tcp。	string
sourceIpPort	啟動連線之來源的 IP 地址和連接埠。	*ip* : *port*
destinationIpPort	起始連線的 IP 地址和連接埠。這將是 SN-E/SN-A 的 IP。	*ip* : *port*
gatewayIpPort	資源閘道用來存取資源的 IP 地址和連接埠。	*ip* : *port*
resourceIpPort	資源的 IP 地址和連接埠。	*ip* : *port*

範例

以下為日誌項目的範例。

```
{
    "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"
}
```

對存取日誌進行故障診斷

本節包含您在存取日誌中可能看到的 HTTP 錯誤代碼說明。

錯誤碼	可能原因
HTTP 400：錯誤的請求	• 用戶端傳送的請求格式不正確，不符合 HTTP 規格。 • 整個請求標頭或超過 100 個標頭的請求標頭超過 60K。 • 用戶端在傳送完整請求內文之前關閉了連線。
HTTP 403：禁止	已為服務設定身分驗證，但傳入的請求未經過身分驗證或授權。
HTTP 404：不存在的服務	您嘗試連線到不存在或未註冊至正確服務網路的服務。
HTTP 500：內部伺服器錯誤	VPC Lattice 發生錯誤，例如無法連線至目標。
HTTP 502：無效的閘道	VPC Lattice 發生錯誤。

Amazon VPC Lattice 的 CloudTrail 日誌

Amazon VPC Lattice 已與 整合AWS CloudTrail，這項服務可提供使用者、角色或 所採取動作的記錄 AWS 服務。CloudTrail 會將 VPC Lattice 的所有 API 呼叫擷取為事件。擷取的呼叫包括來自 VPC Lattice 主控台的呼叫，以及對 VPC Lattice API 操作的程式碼呼叫。您可以使用 CloudTrail 所收集的資訊，判斷對 VPC Lattice 提出的請求、提出請求的 IP 地址、提出請求的時間，以及其他詳細資訊。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是使用根使用者還是使用者憑證提出。
- 請求是否代表 IAM Identity Center 使用者提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務服務提出。

當您建立帳戶 AWS 帳戶 時CloudTrail 會在您的 中處於作用中狀態，而且您會自動存取 CloudTrail 事件歷史記錄。CloudTrail 事件歷史記錄為 AWS 區域中過去 90 天記錄的管理事件，提供可檢視、可搜尋、可下載且不可變的記錄。如需詳細資訊，請參閱「AWS CloudTrail 使用者指南」中的使用 CloudTrail 事件歷史記錄。檢視事件歷史記錄不會產生 CloudTrail 費用。

如需 AWS 帳戶 過去 90 天內持續記錄的事件，請建立線索或 CloudTrail Lake 事件資料存放區。

CloudTrail 追蹤

線索能讓 CloudTrail 將日誌檔案交付至 Amazon S3 儲存貯體。使用 建立的所有線索 AWS 管理主控台 都是多區域。您可以使用 AWS CLI建立單一或多區域追蹤。建議您建立多區域追蹤，因為您擷取 AWS 區域 帳戶中所有 的活動。如果您建立單一區域追蹤，您只能檢視追蹤 AWS 區域中記錄的事件。如需追蹤的詳細資訊，請參閱《AWS CloudTrail 使用者指南》中的為您的 AWS 帳戶建立追蹤和為組織建立追蹤。

您可以透過建立追蹤，免費將持續管理事件的一個複本從 CloudTrail 傳遞至您的 Amazon S3 儲存貯體，但這樣做會產生 Amazon S3 儲存費用。如需 CloudTrail 定價的詳細資訊，請參閱 AWS CloudTrail 定價。如需 Amazon S3 定價的相關資訊，請參閱 Amazon S3 定價。

CloudTrail Lake 事件資料存放區

CloudTrail Lake 讓您能夠對事件執行 SQL 型查詢。CloudTrail Lake 會將分列式 JSON 格式的現有事件轉換為 Apache ORC 格式。ORC 是一種單欄式儲存格式，針對快速擷取資料進行了最佳化。系統會將事件彙總到事件資料存放區中，事件資料存放區是事件的不可變集合，其依據為您透過套用進階事件選取器選取的條件。套用於事件資料存放區的選取器控制哪些事件持續存在並可供您

查詢。如需 CloudTrail Lake 的詳細資訊，請參閱AWS CloudTrail 《 使用者指南》中的使用 AWS CloudTrail Lake。

CloudTrail Lake 事件資料存放區和查詢會產生費用。建立事件資料存放區時，您可以選擇要用於事件資料存放區的定價選項。此定價選項將決定擷取和儲存事件的成本，以及事件資料存放區的預設和最長保留期。如需 CloudTrail 定價的詳細資訊，請參閱 AWS CloudTrail 定價。

若要監控其他動作，請使用存取日誌。如需詳細資訊，請參閱存取日誌。

CloudTrail 中的 VPC Lattice 管理事件

管理事件提供有關在 資源上執行的管理操作的資訊 AWS 帳戶。這些也稱為控制平面操作。根據預設，CloudTrail 記錄管理事件。

Amazon VPC Lattice 會將 VPC Lattice 控制平面操作記錄為管理事件。如需 VPC Lattice 記錄到 CloudTrail 的 Amazon VPC Lattice 控制平面操作清單，請參閱 Amazon VPC Lattice API 參考。

VPC Lattice 事件範例

一個事件代表任何來源提出的單一請求，並包含請求 API 操作的相關資訊、操作的日期和時間、請求參數等。CloudTrail 日誌檔案不是公有 API 呼叫的已排序堆疊追蹤，因此事件不會以任何特定順序顯示。

下列範例顯示 CreateService 操作的 CloudTrail 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"
}
```


下列範例顯示 DeleteService 操作的 CloudTrail 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"
}
```


如需有關 CloudTrail 記錄內容的資訊，請參閱《AWS CloudTrail 使用者指南》中的 CloudTrail record contents。

Amazon VPC Lattice 配額

您的 AWS 帳戶 具有每個 的預設配額，先前稱為限制 AWS 服務。除非另有說明，否則每個配額都是區域特定的。您可以要求提高某些配額，而其他配額無法提高。

若要檢視 VPC Lattice 的配額，請開啟 Service Quotas 主控台。在導覽窗格中，選擇AWS 服務並選取 VPC Lattice。

若要請求提高配額，請參閱《Service Quotas 使用者指南》中的請求提高配額。

您的 AWS 帳戶 具有與 VPC Lattice 相關的下列配額。

名稱	預設	可調整	說明
驗證政策大小	每個支援的區域：10 KB	否	驗證政策中 JSON 檔案的大小上限。
每個群組資源組態的子資源組態	每個支援的區域：60	是	群組資源組態中的子資源組態數目上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個 AWS 區域的網域驗證	每個受支援的區域：5	是	每個帳戶可建立的網域驗證數目上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個服務的接聽程式	每個受支援的區域：2	是	您可以為服務建立的接聽程式數目上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個服務網路的資源組態	每個受支援的區域：500	是	與服務網路相關聯的資源組態數目上限。如需增加額外的容量和限制，請聯絡 AWS Support。

名稱	預設	可調整	說明
每個 AWS 區域的資源組態	每個受支援的區域：2,000	是	AWS 每個 AWS 區域帳戶可以擁有的資源組態數量上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個 VPC 的資源閘道	每個受支援的區域：500	是	VPC 中的資源閘道數量上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個接聽程式的規則	每個受支援的區域：10	是	您可以為服務接聽程式定義的規則數目上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個關聯的安全群組	每個受支援的區域：5	否	您可以新增至 VPC 和服務網路之間關聯的安全群組數量上限。
每個服務網路的服務關聯	每個受支援的區域：500	是	您可以與單一服務網路建立關聯的服務數量上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個區域的服務網路	每個受支援的區域：50	是	每個區域的服務網路數量上限。如需增加額外的容量和限制，請聯絡 AWS Support。

名稱	預設	可調整	說明
每個區域的服務	每個受支援的區域：2,000	是	每個區域的服務數量上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個區域的目標群組	每個受支援的區域：500	是	每個區域的目標群組數量上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個服務的目標群組	每個受支援的區域：10	是	您可以與服務建立關聯的目標群組數量上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個目標群組的目標	每個受支援的區域：1,000	是	您可以與單一目標群組建立關聯的目標數量上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個服務網路的 VPC 關聯	每個受支援的區域：500	是	您可以與單一服務網路建立關聯的 VPCs 數量上限。如需增加額外的容量和限制，請聯絡 AWS Support。
每個服務網路類型服務網路的 VPC 端點	每個受支援的區域：200	是	與服務網路相關聯的服務網路端點數量上限。如需增加額外的容量和限制，請聯絡 AWS Support。

VPC Lattice 不支援下列可用區域：use1-az3、usw1-az2、apne1-az3、apne2-az2、euw1-az4、cac1-az3、ilc1-az2。

下列限制也適用。

限制	Value	說明
每個可用區域每個服務的頻寬	10 Gbps	每個可用區域每個服務配置的預設頻寬。這可以提高，請聯絡您的解決方案架構師 (SA) 或技術客戶經理 (TAM) 以取得進一步協助。
每個連線的最大傳輸單位 (MTU)	8500 位元組	服務可接受的最大資料封包大小。
每個可用區域每個服務的每秒請求數	10,000	對於 HTTP 服務，這是每個可用區域每個服務每秒的預設請求數。這可以提高，請聯絡您的解決方案架構師 (SA) 或技術客戶經理 (TAM) 以取得進一步協助。
VPC Lattice 服務每個連線的連線閒置時間	1 分鐘	連線可在沒有作用中請求 （適用於 HTTP 和 GRPC) 或 VPC Lattice 服務沒有作用中資料傳輸 （適用於 TLS-PASSTHROUGH) 的情況下閒置的預設時間。您可以使用 HTTP 和應用程式層級保持連線，將此閒置逾時延長至最長連線生命週期。這可以提高，請聯絡您的解決方案架構師 (SA) 或技術客戶經理 (TAM) 以取得進一步協助。
VPC Lattice 服務每個連線的最大連線生命週期	10 分鐘	用戶端與 VPC Lattice 服務伺服器之間可開啟連線的時間上限。
VPC Lattice 資源的每個連線的最大連線生命週期	NA	VPC Lattice 不會對資源施加任何生命週期連線限制。用戶端和伺服器會決定生命週期連線持續時間，同時注意

限制	Value	說明
		VPC Lattice 資源的閒置逾時，也就是 350 秒。
VPC Lattice 資源的每個連線的連線閒置時間	350 秒	您可以使用 TCP 保持連線來延長此閒置逾時。
每個 VPC 的服務網路	1 個服務網路	您只能透過 關聯將 VPC 連線到一個服務網路。若要將 VPC 連線至多個服務網路，您可以使用服務網路類型的 VPC 端點。

Amazon VPC Lattice 使用者指南的文件歷史記錄

下表說明 VPC Lattice 的文件版本。

變更	描述	日期
新增資源閘道的可設定 IP 地址	VPC Lattice 現在支援資源閘道的可設定 IP 地址。	2025 年 10 月 7 日
新增 的 VPC Lattice Oracle Database@AWS	發行的 VPC Lattice Oracle Database@AWS 。	2025 年 6 月 26 日
新增管理端點的雙堆疊支援	VPC Lattice 現在支援所有 VPC Lattice 管理 APIs的雙堆疊 (IPv4 和 IPv6) 端點。	2025 年 4 月 30 日
共用和存取 資源	VPC Lattice 現在支援跨 VPC 和帳戶邊界共用和存取資源。這包括 VPCLatticeReadOnlyAccess 和 VPCLatticeFullAccess 政策的更新。	2024 年 12 月 1 日
TLS 傳遞	VPC Lattice 現在支援 TLS 傳遞，可讓您在應用程式中執行 TLS 終止以進行end-to-end身分驗證。	2024 年 5 月 14 日
Lambda 事件結構版本	VPC Lattice 現在支援新版本的 Lambda 事件結構。	2023 年 9 月 7 日
支援共用 VPCs	參與者可以在共用 VPC 中建立 VPC Lattice 目標群組。	2023 年 7 月 5 日
一般可用性版本	一般可用性 (GA) 的 VPC Lattice 使用者指南版本	2023 年 3 月 31 日

VPC Lattice 現在會報告其 AWS 受管政策的變更	受管政策的變更會在「安全性」章節的「VPC Lattice 的 AWS 受管政策」中報告。	2023 年 3 月 29 日
Application Load Balancer 目標類型的支援	VPC Lattice 現在支援建立 Application Load Balancer 類型目標群組。	2023 年 3 月 29 日
支援所有執行個體類型	VPC Lattice 現在支援所有執行個體類型。	2023 年 3 月 27 日
IPv6 支援	VPC Lattice 現在支援 IPv4 和 IPv6 IP 目標群組。	2023 年 3 月 27 日
運作狀態檢查的 HTTP2 通訊協定版本	當目標群組通訊協定版本為 HTTP2 時，現在支援運作狀態檢查。	2023 年 3 月 27 日
已修正接聽程式規則的回應動作	VPC Lattice 服務的接聽程式現在支援轉送動作以外的固定回應動作。	2023 年 3 月 27 日
支援自訂網域名稱	您現在可以為 VPC Lattice 服務設定自訂網域名稱	2023 年 2 月 14 日
支援 BYOC（自帶憑證）	VPC Lattice 支援在 ACM 中使用您自己的 SSL/TLS 憑證做為自訂網域名稱。	2023 年 2 月 14 日
VPC Lattice 現在會報告不支援執行個體類型的更新清單	三個額外的執行個體已新增至不支援的執行個體清單。	2023 年 1 月 26 日
VPC Lattice 現在會報告其 AWS 受管政策的變更	從 2022 年 12 月 5 日開始，受管政策的變更會在「AWS 安全性」章節中的「VPC Lattice 的受管政策」主題中報告。列出的第一個變更是新增 CloudWatch 監控所需的許可。	2022 年 12 月 5 日

初始版本	VPC Lattice 使用者指南的初始版本	2022 年 12 月 5 日

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。