VPC ピアリング接続

Amazon Virtual Private Cloud

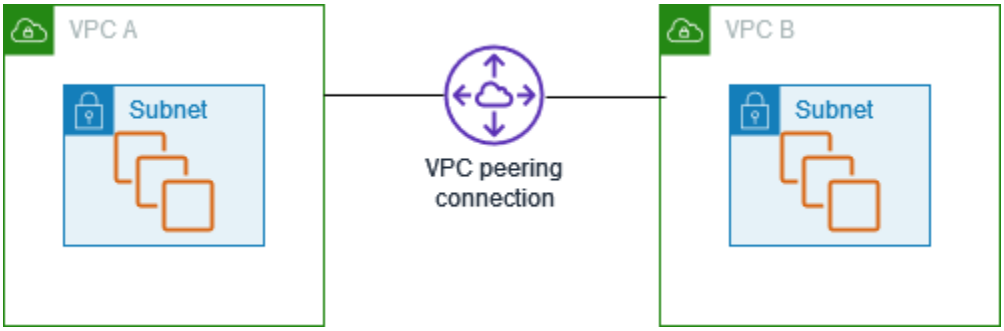
Amazon Virtual Private Cloud: VPC ピアリング接続

Table of Contents

VPC ピア機能とは

仮想プライベートクラウド (VPC) は、AWS アカウント専用の仮想ネットワークです。VPC は、AWS クラウドの他の仮想ネットワークから論理的に切り離されています。VPC 内には、Amazon EC2 インスタンスなどの AWS リソースを起動できます。

VPC ピアリング接続は、プライベート IPv4 アドレスまたは IPv6 アドレスを使用して 2 つの VPC 間でトラフィックをルーティングすることを可能にするネットワーク接続です。どちらの VPC のインスタンスも、同じネットワーク内に存在しているかのように、相互に通信できます。VPC ピアリング接続は、お客様の VPC 間や、他の AWS アカウントの VPC との間に作成できます。VPC は複数の異なるリージョンに存在できます (これはリージョン間 VPC ピアリング接続とも呼ばれます)。



AWS では VPC の既存のインフラストラクチャを使用して VPC ピアリング接続を作成しています。これはゲートウェイでも VPN 接続でもなく、個別の物理ハードウェアに依存するものではありません。通信の単一障害点や帯域幅のボトルネックは存在しません。

VPC ピアリング接続を使用すると、データの転送が容易になります。例えば、複数の AWS アカウントがある場合、これらのアカウント間で VPC をピアリングし、ファイル共有ネットワークを作成できます。また、VPC ピア接続を使用して、他の VPC からお客様のいずれかの VPC に存在するリソースへのアクセスを許可することもできます。

異なる AWS リージョンにある VPC 間でのピアリング関係を確立する場合、異なる AWS リージョンにある VPC 内のリソース (EC2 インスタンスや Lambda 関数など) は、ゲートウェイ、VPN 接続、またはネットワークアプライアンスを使用せずに、プライベート IP アドレスを使用して相互に通信できます。トラフィックはプライベート IP アドレス空間に残ります。すべてのリージョン間トラフィックは AWS の施設から移動する前に暗号化され、単一障害点または帯域幅のボトルネックは存在しません。トラフィックは、常にグローバル AWS バックボーンにとどまり、パブリックインターネットを通過することがないため、一般的なエクスプロイトや DDoS 攻撃などの脅威を減らすことができます。また、リージョン間 VPC ピアリング接続を利用すると、シンプルで費用対効果の

高い方法により、リージョン間でリソースを共有したり、地理的な冗長性のためにデータをレプリケートしたりできます。

VPC ピアリング接続の料金

VPC ピアリング接続の確立に料金はかかりません。アベイラビリティーゾーン内で維持される VPC ピアリング接続経由でのデータ転送は、異なるアカウント間であってもすべて無料です。複数のアベイラビリティーゾーンとリージョンにまたがる VPC ピアリング接続経由でのデータ転送には料金が発生します。詳細については、Amazon EC2 の料金 を参照してください。

VPC ピアリング接続の仕組み

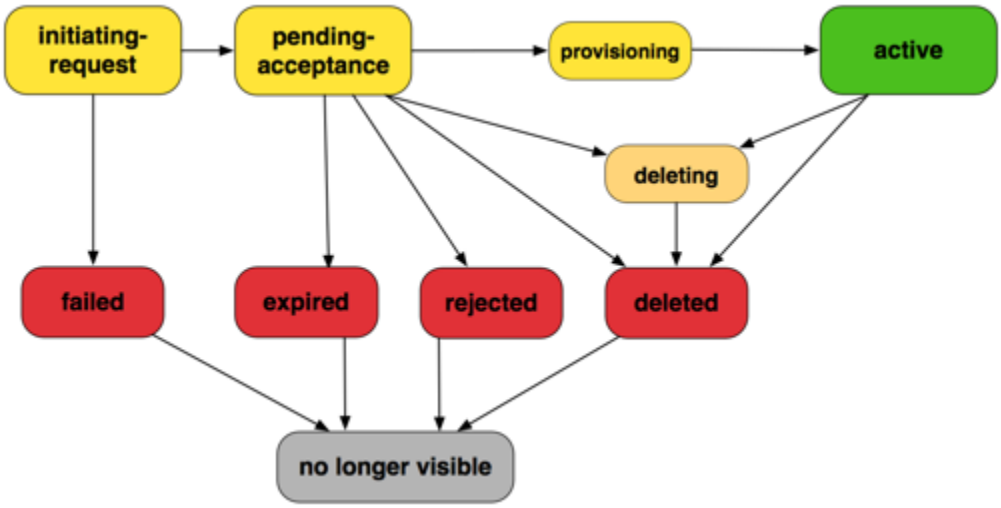
VPC ピアリングプロセスの手順を以下に示します。

1. リクエスタ VPC の所有者がアクセプタ VPC の所有者にリクエストを送信して、VPC ピアリング接続を作成します。アクセプタ VPC を所有できるのは、ユーザーまたは他の AWS アカウントです。また、アクセプタ VPC はリクエスタ VPC の CIDR ブロックと重複する CIDR ブロックを保持することはできません。
2. アクセプタ VPC の所有者は、VPC ピアリング接続リクエストを承認して VPC ピアリング接続を有効にします。
3. プライベート IP アドレスを使用して VPC 間でのトラフィックの流れを有効にするには、VPC ピアリング接続の各 VPC の所有者が、他の VPC (ピア VPC) の IP アドレス範囲を指すルートを 1 つ以上の VPC ルートテーブルに手動で追加する必要があります。
4. 必要に応じて、ピア VPC との間で送受信されるトラフィックが制限されることのないように、EC2 インスタンスに関連付けられているセキュリティグループのルールを更新します。両方の VPC が同じリージョンにある場合は、セキュリティグループのインバウンドルールまたはアウトバウンドルールの送信元または送信先として、ピア VPC のセキュリティグループを参照できます。
5. デフォルトの VPC ピアリング接続オプションでは、VPC ピアリング接続のいずれかの側の EC2 インスタンスがパブリック DNS ホスト名を使用して相互にアドレス指定する場合、ホスト名は EC2 インスタンスのパブリック IP アドレスに解決されます。この動作を変更するには、VPC 接続に対して DNS ホスト名解決を有効にします。DNS ホスト名解決を有効にした後、VPC ピアリング接続の両側の EC2 インスタンスがパブリック DNS ホスト名を使用して相互にアドレス指定する場合、ホスト名は EC2 インスタンスのプライベート IP アドレスに解決されます。

詳細については、「VPC ピアリング接続」を参照してください。

VPC ピアリング接続のライフサイクル

VPC ピアリング接続は、リクエストが開始されたときから始まるさまざなステージで使用されます。それぞれのステージで実行可能なアクションがあり、そのライフサイクルの最後で、VPC ピアリング接続は Amazon VPC コンソールと API またはコマンドライン出力に一定期間表示されます。



- [Initiating-request]: VPC ピアリング接続のリクエストが開始されました。このステージでは、ピアリング接続は失敗する可能性があり、`pending-acceptance` に移行する場合があります。
- [Failed]: VPC ピアリング接続のリクエストが失敗しました。この状態の間は、ピアリング接続を承認、拒否、または削除できません。失敗した VPC ピアリング接続は、リクエスタに 2 時間表示されます。
- [Pending-acceptance]: VPC ピアリング接続リクエストがアクセプタ VPC の所有者からの承認を待っています。このステージの間、リクエスタ VPC の所有者はリクエストを削除できます。アクセプタ VPC の所有者はリクエストを受諾するか拒否できます。リクエストに対するアクションがない場合、リクエストは 7 日後に有効期限が切れます。
- [Expired]: VPC ピアリング接続のリクエストが有効期限切れとなりました。どちらの VPC 所有者もアクションを実行することはできません。期限切れとなった VPC ピアリング接続は、両方の VPC 所有者に対して 2 日間表示されます。
- [Rejected]: アクセプタ VPC の所有者は、VPC ピアリング接続リクエストの `pending-acceptance` を拒否しました。この状態の間は、リクエストを承認することはできません。拒否された VPC ピアリング接続は、リクエスタ VPC の所有者に対して 2 日間表示され、アクセプタ VPC の所有者に対しては 2 時間表示されます。リクエストが同じ AWS アカウント内で作成されている場合、拒否されたリクエストは 2 時間表示されます。
- [Provisioning]: VPC ピアリング接続リクエストが承認され、間もなく `active` 状態に移行します。
- [Active]: VPC ピアリング接続がアクティブであり、トラフィックは VPC 間を流れることができます (セキュリティグループとルートテーブルがトラフィックの流れを許可する場合)。この状態の間、どちらの VPC の所有者も VPC ピア接続を削除できますが、拒否できません。

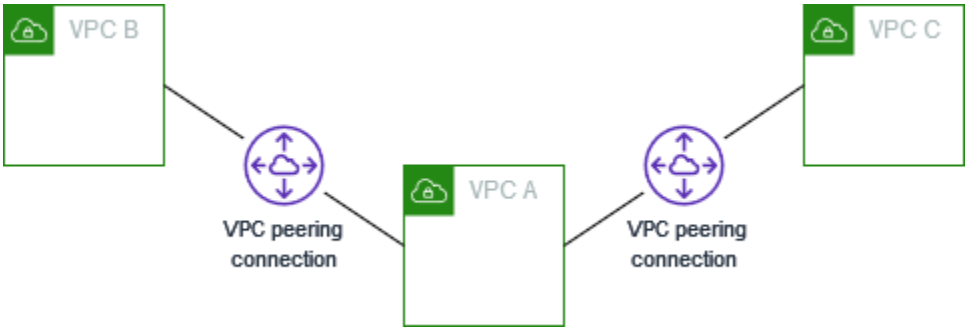
> **Note**
>
> VPC が存在するリージョンでイベントが発生し、トラフィックの流れが中断した場合、VPC ピアリング接続のステータスは `Active` のままとなります。

- [Deleting] (削除中): 削除中であるリージョン間 VPC ピアリング接続に適用されます。いずれかの VPC の所有者が `active` な VPC ピアリング接続を削除するリクエストを送信したか、リクエスタ VPC の所有者が `pending-acceptance` の VPC ピアリング接続リクエストを削除するリクエストを送信しました。
- [Deleted]: `active` な状態の VPC ピアリング接続がいずれかの VPC 所有者によって削除されたか、`pending-acceptance` な状態の VPC ピアリング接続リクエストがリクエスタ VPC の所有者によって削除されました。この状態の間、VPC ピアリング接続を承認または拒否できません。VPC ピアリング接続は、それを削除した当事者に対して 2 時間表示され、その他の当事者に対しては 2 日間表示されます。VPC ピアリング接続が同じ AWS アカウント内で作成されている場合、削除されたリクエストは 2 時間表示されます。

複数の VPC ピアリング接続

VPC ピアリング接続は、2 つの VPC 間の 1 対 1 関係です。自分の各 VPC に対して複数の VPC ピア接続を作成できますが、推移的なピア接続関係はサポートされません。自分の VPC と直接ピア関係にない VPC とのピア関係を作成することはできません。

次の図は、2 つの異なる VPC とピア関係にある VPC の例です。2 つの VPC ピアリング接続があり、VPC A は VPC B と VPC C の両方とピア関係にあります。VPC B と VPC C はピア関係にはありません。VPC B と VPC C との間のピアリング接続の中継データポイントとして VPC A を使用することはできません。VPC B と VPC C との間でトラフィックのルーティングを有効にしたい場合は、その VPC 間で一意の VPC ピアリング接続を作成する必要があります。



VPC ピアリングの制限事項

VPC ピアリング接続には次の制限事項を考慮してください。場合によっては、VPC ピアリング接続の代わりに Transit Gateway アタッチメントを使用できます。詳細については、Amazon VPC Transit Gateway の「Example transit gateway scenarios」を参照してください。

Connections

- VPC ごとに実行中および保留中の VPC ピアリング接続の数にはクォータがあります。詳細については、「クォータ」を参照してください。
- 2 つの VPC 間で同時に複数の VPC ピアリング接続を持つことはできません。
- VPC ピアリング接続用に作成したタグは、作成元のアカウントまたはリージョンでのみ適用されます。
- ピア VPC で Amazon DNS サーバーに接続したり、クエリを実行したりすることはできません。
- VPC ピア接続の VPC の IPv4 CIDR ブロックが、RFC 1918 で指定されたプライベート IPv4 アドレス範囲外である場合、その VPC のプライベート DNS ホスト名をプライベート IP アドレスに解決することはできません。プライベート DNS ホスト名をプライベート IP アドレスに解決するには、VPC ピア接続の DNS 解決サポートを有効にできます。詳細については、「VPC ピアリング接続の DNS 解決を有効にする」を参照してください。
- VPC ピアリング接続の一方のリソースが IPv6 経由で通信できるようにします。IPv6 CIDR ブロックを各 VPC と関連付け、IPv6 通信を行うために VPC のインスタンスを有効にし、ピアリング接続用の IPv6 トラフィックを VPC ピアリング接続にルーティングする必要があります。
- VPC ピアリング接続のユニキャストリバースパス転送 (uRPF) はサポートされていません。詳細については、「レスポンストラフィックのルーティング」を参照してください。

重複する CIDR ブロック

- IPv4 または IPv6 CIDR ブロックが一致または重複する VPC 間で VPC ピアリング接続を作成することはできません。
- 複数の IPv4 CIDR ブロックがある場合、いずれかの CIDR ブロックが重複しているときは、重複していない CIDR ブロックのみ、または IPv6 CIDR ブロックのみを使用する場合でも、VPC ピアリング接続を作成できません。

推移的なピアリング接続

- VPC ピアリングでは、推移的なピアリング関係がサポートされません。例えば、VPC A と VPC B の間、および VPC A と VPC C との間に VPC ピアリング接続がある場合、VPC A 経由で VPC B から VPC C へトラフィックをルーティングすることはできません。VPC B と VPC C との間でトラフィックをルーティングするには、その VPC B と VPC C との間で VPC ピアリング接続を作成する必要があります。詳細については、「3 つの VPC が相互にピアリング接続」を参照してください。

ゲートウェイまたはプライベート接続経由のエッジツーエッジルーティング

- VPC A にインターネットゲートウェイがある場合、VPC B のリソースは VPC A のインターネットゲートウェイを使用してインターネットにアクセスすることはできません。
- VPC A にある NAT デバイスが VPC A のサブネットへのインターネットアクセスを提供している場合、VPC B のリソースは VPC A の NAT デバイスを使用してインターネットにアクセスすることはできません。
- VPC A に企業ネットワークへの VPN 接続がある場合、VPC B のリソースは VPN 接続を使用して企業ネットワークと通信することはできません。
- VPC A に企業ネットワークへの Direct Connect 接続がある場合、VPC B のリソースは Direct Connect 接続を使用して企業ネットワークと通信することはできません。
- VPC A にゲートウェイエンドポイントがあり、このゲートウェイエンドポイントが VPC A のプライベートサブネットに対して Amazon S3 への接続を提供する場合、VPC B のリソースはゲートウェイエンドポイントを使用して Amazon S3 にアクセスすることはできません。

リージョン間 VPC ピアリング接続

- ジャンボフレームの場合、同じリージョン内の VPC ピアリング接続間の最大送信単位 (MTU) は 9001 バイトです。リージョン間の VPC ピアリング接続の MTU は 8500 バイトです。ジャンボフレームの詳細については、「Amazon EC2 ユーザーガイド」の「ジャンボフレーム (9001 MTU)」を参照してください。
- ピアリング接続された VPC のプライベート DNS ホスト名をプライベート IP アドレスに解決するには、VPC ピアリング接に対して DNS 解決のサポートを有効にする必要があります。これは、VPC の IPv4 CIDR ブロックが、RFC 1918 で指定されたプライベート IPv4 アドレス範囲内にあっても同じです。

共有 VPC およびサブネット

- ピアリング接続を操作 (説明、作成、承認、拒否、変更、削除) できるのは VPC の 所有者のみです。参加者はピアリング接続を使用して操作することはできません。詳細については、「Amazon VPC ユーザーガイド」の「他のアカウントでの共有 VPC の使用」を参照してください。

VPC ピアリング接続

VPC ピアリングを使用すると、同じまたは異なる AWS リージョンの 2 つの VPC を接続できます。この結果、ある VPC のインスタンスは、すべてが同じネットワークの一部であるかのように、他の VPC のインスタンスと通信できます。

VPC ピアリングは、プライベート IPv4 アドレスまたは IPv6 アドレスを使用して、2 つの VPC 間に直接ネットワークルートを作成します。接続された VPC 間で送信されるトラフィックは、インターネット、VPN 接続、または AWS Direct Connect 接続を経由しません。そのため、VPC ピアリングは、データベースやウェブサーバーなどのリソースを VPC の境界を越えて共有するための安全な方法になります。

VPC ピアリング接続を確立するには、1 つの VPC からピアリング接続リクエストを作成し、他の VPC の所有者がそのリクエストを承認します。接続が確立されたら、ルートテーブルを更新して VPC 間でトラフィックをルーティングできます。この結果、一方の VPC のインスタンスが他方の VPC のリソースにアクセスできるようになります。

VPC ピアリングは、マルチ VPC アーキテクチャを構築し、AWS において組織の境界を越えてリソースを共有するための重要なツールです。VPN やその他のネットワークサービスの設定における複雑さのない、VPC を接続するためのシンプルで低レイテンシーの方法が提供されます。

VPC ピアリング接続を作成および操作するには、次の手順を使用します。

タスク

- VPC ピアリング接続を作成する
- VPC ピアリング接続の受諾または拒否
- VPC ピアリング接続のルートテーブルを更新する
- セキュリティグループの更新とピアセキュリティグループの参照
- VPC ピアリング接続の DNS 解決を有効にする
- VPC ピアリング接続を削除する
- VPC ピアリング接続のトラブルシューティング

VPC ピアリング接続を作成する

VPC ピアリング接続を作成するには、最初に別の VPC とのピアリング接続リクエストを作成します。リクエストをアクティブ化するには、アクセプタ VPC の所有者がリクエストを承認する必要があります。次のピアリング接続がサポートされています。

- 同じアカウントの同じリージョンにある VPC 間
- 同じアカウント内の異なるリージョンにある VPC 間
- 異なるアカウントの同じリージョンにある VPC 間
- 異なるアカウントの異なるリージョンにある VPC 間

リージョン間 VPC ピアリング接続の場合、リクエストはリクエスタ VPC のリージョンから行う必要があり、リクエストはアクセプタ VPC のリージョンから受け入れられる必要があります。詳細については、「the section called “受諾または拒否”」を参照してください。

タスク

- 前提条件
- コンソールを使用してピアを作成する
- コマンドラインを使用してピアリング接続を作成する

前提条件

- VPC ピアリング接続の制限を確認します。
- VPC に重複している IPv4 CIDR ブロックがないことを確認する 重複している場合、VPC ピアリング接続のステータスが直ちに `failed` に移行します。この制限は、VPC に固有の IPv6 CIDR ブロックがあっても適用されます。

コンソールを使用してピアを作成する

VPC ピアリング接続を作成するには、次の手順を実行します。

コンソールを使用してピアリング接続を作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[Peering Connections] (ピアリング接続) をクリックします。

3. [Create Peering Connection] (ピアリング接続の作成) をクリックします。
4. (オプション) [名前] に、VPC ピアリング接続の名前を指定します。これにより、Name のキーと指定した値を持つタグが作成されます。
5. [VPC ID (リクエスタ)] で、現在のアカウントから VPC を選択します。
6. [ピアリング接続するもうひとつの VPC を選択] で、次の操作を行います。
   a. [アカウント] で、別のアカウントの VPC とピア接続するには、[別のアカウント] を選択し、アカウント ID を入力します。それ以外の場合は、[マイアカウント] を保持します。
   b. [リージョン] で、別のリージョンの VPC とピア接続するには、[別のリージョン] をクリックしてリージョンを選択します。それ以外の場合は、[このリージョン] を保持します。
   c. VPC ID (アクセプタ) で、指定されたアカウントとリージョンから VPC を選択します。
7. (オプション) タグを追加するには、[Add new tag] (新しいタグを追加) を選択し、タグキーとタグ値を入力します。
8. [Create Peering Connection] (ピアリング接続の作成) をクリックします。
9. アクセプタアカウントの所有者はピアリング接続を承認する必要があります。詳細については、「the section called “受諾または拒否”」を参照してください。
10. 両方の VPC のルートテーブルを更新して、それらの間の通信を有効にします。詳細については、「the section called “ルートテーブルの更新”」を参照してください。

コマンドラインを使用してピアリング接続を作成する

次のコマンドを使用して VPC ピアリング接続を作成できます。

- create-vpc-peering-connection ( ) AWS CLI」
- New-EC2VpcPeeringConnection ( ) AWS Tools for Windows PowerShell」

VPC ピアリング接続の受諾または拒否

pending-acceptance 状態にある VPC ピアリング接続は、有効化されるアクセプタ VPC の所有者が承認する必要があります。Deleted ピア接続状態の詳細については、「VPC ピアリング接続のライフサイクル」を参照してください。別の AWS アカウントに送信した VPC ピアリング接続リクエストを承認することはできません。同じ AWS アカウント内の VPC 間で VPC ピアリング接続を作成するには、リクエストを自分で作成し承認することができます。

受信した VPC ピアリング接続リクエストで pending-acceptance 状態にあるものを拒否できます。信頼できる既知の AWS アカウント からの VPC ピアリング接続のみを承認するようにしてください。不要なリクエストは拒否できます。Rejected ピア接続状態の詳細については、「VPC ピアリング接続のライフサイクル」を参照してください。

> **Important**
>
> 不明な AWS アカウントからの VPC ピアリング接続は承認しないでください。悪意のあるユーザーが VPC ピアリング接続リクエストを送信して、VPC に対して不正なネットワークアクセスを行なう場合があります。これは、ピアフィッシングと呼ばれます。AWS アカウントまたは VPC についての情報にリクエスタがアクセスするリスクなしで、不要な VPC ピアリング接続リクエストを安全に拒否できます。詳細については、「VPC ピアリング接続の受諾または拒否」を参照してください。リクエストを無視して有効期限が切れるのを待つこともできます。デフォルトでは、7 日後にリクエストの期限が切れます。

コンソールを使用してピアリング接続を承認または拒否するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. リージョンセレクタを使用して、アクセプタ VPC のリージョンを選択します。
3. ナビゲーションペインで、[Peering Connections] (ピアリング接続) をクリックします。
4. ピアリング接続を拒否するには、VPC ピアリング接続を選択し、[アクション]、[リクエストを拒否] の順に選択します。確認を求めるメッセージが表示されたら、[リクエストを拒否] を選択します。
5. ピアリング接続を承諾するには、保留中の VPC ピアリング接続 (ステータスは pending-acceptance) を選択し、[アクション]、[リクエストを承諾] を選択します。ピアリング接続のライフサイクル状態の詳細については、「VPC ピアリング接続のライフサイクル」を参照してください。

   保留中の VPC ピアリング接続がない場合は、アクセプタ VPC のリージョンが選択されていることを確認します。
6. 確認を求められたら、[リクエストの承諾] を選択します。
7. [ルートテーブルを今すぐ変更] を選択して VPC ルートテーブルにルートを追加すると、ピアリング接続を介してトラフィックを送受信できるようになります。詳細については、「VPC ピアリング接続のルートテーブルを更新する」を参照してください。

コマンドラインを使用してピアリング接続を受け入れるには

- accept-vpc-peering-connection ( ) AWS CLI」
- Approve-EC2VpcPeeringConnection ( ) AWS Tools for Windows PowerShell」

コマンドラインを使用してピアリング接続を拒否するには

- reject-vpc-peering-connection ( ) AWS CLI」
- Deny-EC2VpcPeeringConnection ( ) AWS Tools for Windows PowerShell」

VPC ピアリング接続のルートテーブルを更新する

ピア接続先 VPC 内のインスタンス間のプライベート IPv4 トラフィックを有効にするには、両方のインスタンスのサブネットに関連付けられたルートテーブルにルートを追加する必要があります。このルートの送信先は、ピア VPC の CIDR ブロック (または CIDR ブロックの一部) であり、ターゲットは VPC ピアリング接続の ID です。詳細については、Amazon VPC ユーザーガイドの「ルートテーブルを設定する」を参照してください。

次に、2 つのピア接続先 VPC (VPC A と VPC B) のインスタンス間の通信を可能にするルートテーブルの例を示します。各テーブルには、ローカルルートと、ピア VPC のトラフィックを VPC ピアリング接続に送信するルートがあります。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A CIDR*	ローカル
	VPC B CIDR	pcx-*11112222*
VPC B	*VPC B CIDR*	ローカル
	VPC A CIDR	pcx-*11112222*

同様に、VPC ピアリング接続の VPC に関連付けられている IPv6 CIDR ブロックがある場合は、Pv6 を介したピア VPC との通信を可能にするルートを追加できます。

VPC ピアリング接続でサポートされているルートテーブルのその他の情報については、「一般的な VPC ピアリング接続設定」を参照してください。

考慮事項

- IPv4 CIDR ブロックが重複または一致する複数の VPC にピアリング接続された VPC がある場合は、自分の VPC から間違った VPC にレスポンストラフィックを送信しないようにルートテーブルが設定されていることを確認します。AWS は現在、パケットの送信元 IP を確認してリプライパケットを送信元にルーティングするユニキャストリバースパス転送 (uRPF) を VPC ピアリング接続でサポートしていません。詳細については、「レスポンストラフィックのルーティング」を参照してください。
- アカウントにはルートテーブルごとに追加できるエントリ数にクォータがあります。VPC の VPC ピアリング接続数が 1 つのルートテーブルのルートテーブルエントリクォータを超える場合は、それぞれがカスタムルートテーブルに関連付けられた複数のサブネットの使用を検討してください。
- `pending-acceptance` 状態にある VPC ピアリング接続のルートを追加できます。ただし、ルートには `blackhole` 状態があり、VPC ピアリング接続が `active` 状態になるまで有効になりません。

VPC ピアリング接続に IPv4 ルートを追加するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Route tables] (ルートテーブル) を選択します。
3. インスタンスが存在するサブネットに関連付けられたルートテーブルの横にあるチェックボックスをオンにします。

   サブネットを明示的に関連付けられたルートテーブルがない場合、VPC メインルートテーブルは暗示的にそのサブネットに関連付けられます。
4. [アクション]、[ポリシーの編集] の順に選択します。
5. [Add Rule (ルートの追加)] を選択します。
6. [Destination] に、VPC ピアリング接続のネットワークトラフィックを誘導する必要のある IPv4 アドレスの範囲を入力します。ピア VPC の IPv4 CIDR ブロック全体、特定の範囲、または通信するインスタンスの IP アドレスのような個別の IPv4 アドレスを指定できます。例えば、ピア VPC の CIDR ブロックが `10.0.0.0/16` の場合、`10.0.0.0/24` の部分、または特定の IP アドレス `10.0.0.7/32` を指定できます。
7. [ターゲット] に VPC ピアリング接続を選択します。
8. [Save changes] (変更の保存) をクリックします。

ピア VPC の所有者は、VPC ピアリング接続を介して VPC にトラフィックを戻すためのルートを追加するために、これらのステップを完了する必要があります。

複数の AWS リージョンに IPv6 アドレスを使用するリソースがある場合は、リージョン間ピアリング接続を作成できます。その後、リソース間の通信用に IPv6 ルートを追加できます。

VPC ピアリング接続に IPv6 ルートを追加するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Route tables] (ルートテーブル) を選択します。
3. インスタンスが存在するサブネットに関連付けられたルートテーブルの横にあるチェックボックスをオンにします。

   > **Note**
   >
   > そのサブネットに関連付けられたルートテーブルがない場合は、VPC のメインルートテーブルを選択します。サブネットがこのルートをデフォルトで使用するためです。

4. [アクション]、[ポリシーの編集] の順に選択します。
5. [Add Rule (ルートの追加)] を選択します。
6. [Destination] に、ピア VPC の IPv6 アドレス範囲を入力します。ピア VPC の IPv6 CIDR ブロック全体、特定の範囲、または個別の IPv6 アドレスを指定できます。例えば、ピア VPC の CIDR ブロックが `2001:db8:1234:1a00::/56` の場合、`2001:db8:1234:1a00::/64` の部分、または特定の IP アドレス `2001:db8:1234:1a00::123/128` を指定できます。
7. [ターゲット] に VPC ピアリング接続を選択します。
8. [Save changes] (変更の保存) をクリックします。

詳細については、Amazon VPC ユーザーガイドの「ルートテーブル」を参照してください。

コマンドラインを使用してルートを追加または置換するには

- create-route および replace-route(AWS CLI)
- New-EC2Route および Set-EC2Route(AWS Tools for Windows PowerShell)

セキュリティグループの更新とピアセキュリティグループの参照

VPC セキュリティグループのインバウンドルールまたはアウトバウンドルールを更新して、ピアリング接続された VPC のセキュリティグループを参照できます。これにより、トラフィックはピアリング接続された VPC の参照セキュリティグループに関連付けられたインスタンスに出入りできます。

> **Note**
>
> ピア VPC のセキュリティグループは、コンソールで選択できるものとして表示されません。

要件

- ピア VPC でセキュリティグループを参照するには、VPC ピアリング接続の状態が `active` である必要があります。
- ピア VPC はアカウントの VPC とするか、別の AWS アカウントの VPC とすることができます。別の AWS アカウントにあるが、同じリージョンに存在するセキュリティグループを参照するには、セキュリティグループの ID を持つアカウント番号を含めます。例えば、`123456789012/sg-1a2b3c4d`。
- 別のリージョンにあるピア VPC のセキュリティグループは参照できません。代わりに、ピア VPC の CIDR ブロックを使用します。
- ミドルボックスアプライアンスを介して異なるサブネット内の 2 つのインスタンス間のトラフィックを転送するようにルートを設定するには、両方のインスタンスのセキュリティグループでインスタンス間のトラフィックがフローできるようにする必要があります。各インスタンスのセキュリティグループは、他のインスタンスのプライベート IP アドレス、または他のインスタンスを含むサブネットの CIDR 範囲を送信元として参照する必要があります。他のインスタンスのセキュリティグループを送信元として参照する場合、インスタンス間のトラフィックは許可されません。

コンソールを使用してセキュリティグループルールを更新するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[セキュリティグループ] を選択します。
3. セキュリティグループを選択し、次のいずれかを実行します。

- インバウンドルールを変更するには、[アクション]、[インバウンドルールを編集] の順に選択します。
- アウトバウンドルールを変更するには、[アクション]、[アウトバウンドルールを編集] の順に選択します。

4. ルールを追加するには、[ルールの追加] を選択し、タイプ、プロトコル、ポート範囲を指定します。[送信元] (インバウンドルール) または [送信先] (アウトバウンドルール) で、次のいずれかの操作を行います。
   - 同じアカウントとリージョンのピア VPC の場合は、セキュリティグループの ID を入力します。
   - 別のアカウントにあるが、同じリージョンに存在するピア VPC の場合は、アカウント ID とセキュリティグループ ID をスラッシュで区切って入力します (例: `123456789012/sg-1a2b3c4d`)。
   - 別のリージョンに存在するピア VPC の場合は、ピア VPC の CIDR ブロックを入力します。
5. 既存のルールを編集するには、値 (ソースや説明など) を変更します。
6. ルールを削除するには、ルールの隣にある [削除] を選択します。
7. [Save Rules] (ルールの保存) を選択してください。

コマンドラインを使用してインバウンドルールを更新するには

- authorize-security-group-ingress および revoke-security-group-ingress (AWS CLI)
- Grant-EC2SecurityGroupIngress および Revoke-EC2SecurityGroupIngress (AWS Tools for Windows PowerShell)

例えば、ピア VPC の `sg-bbbb2222` からの HTTP 経由のインバウンドアクセスを許可するようにセキュリティグループ `sg-aaaa1111` を更新するには、次のコマンドを使用します。ピア VPC が同じリージョンに存在するが、別のアカウントにある場合は、`--group-owner` *`aws-account-id`* を追加します。

```
aws ec2 authorize-security-group-ingress --group-id sg-aaaa1111 --protocol tcp --
port 80 --source-group sg-bbbb2222
```

コマンドラインを使用してアウトバウンドルールを更新するには

- authorize-security-group-egress および revoke-security-group-egress (AWS CLI)

- Grant-EC2SecurityGroupEgress および Revoke-EC2SecurityGroupEgress (AWS Tools for Windows PowerShell)

セキュリティグループルールを更新したら、describe-security-groups コマンドを使って、セキュリティグループルールで参照されるセキュリティグループを確認します。

参照されるセキュリティグループを特定する

セキュリティグループがピア VPC のセキュリティグループのルールで参照されているかどうかを確認するには、アカウントの 1 つ以上のセキュリティグループに対して、次のいずれかのコマンドを使用します。

- describe-security-group-references (AWS CLI)
- Get-EC2SecurityGroupReference (AWS Tools for Windows PowerShell)

次の例では、応答はセキュリティグループ sg-bbbb2222 が VPC のセキュリティグループ vpc-aaaaaaaa で参照されていることを示します。

```
aws ec2 describe-security-group-references --group-id sg-bbbb2222
```

```
{
  "SecurityGroupsReferenceSet": [
    {
      "ReferencingVpcId": "vpc-aaaaaaaa",
      "GroupId": "sg-bbbb2222",
      "VpcPeeringConnectionId": "pcx-b04deed9"
    }
  ]
}
```

VPC ピアリング接続が削除されたか、またはピア VPC の所有者が、参照されたセキュリティグループを削除した場合、セキュリティグループルールは古くなります。

古くなったセキュリティグループルールを表示および削除する

古いセキュリティグループルールとは、同じ VPC またはピア VPC 内の削除されたセキュリティグループを参照するルール、または VPC ピアリング接続が削除されたピア VPC のセキュリティグループを参照するルールのことです。セキュリティグループルールは古くなっても、セキュリティグループから自動的に削除されません。手動で削除する必要があります。VPC ピアリング接続が削除

されたためにセキュリティグループルールが古くなった場合、同じ VPC で新しい VPC ピアリング接続を作成すると、そのルールは古くなったとマークされなくなります。

Amazon VPC コンソールを使用して、VPC の古くなったセキュリティグループルールを表示および削除できます。

古くなったセキュリティグループルールを表示および削除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Security Groups] (セキュリティグループ) を選択します。
3. [Action] (アクション)、[Manage stale rules] (古いルールの管理) の順に選択します。
4. VPC で古いルールを持つ VPC を選択します。
5. [Edit] を選択します。
6. 削除するルールの横にある [Delete] (削除) ボタンを選択します。[変更のプレビュー]、[ルールの保存] を選択します。

コマンドラインを使用して古いセキュリティグループルールを記述するには

- describe-stale-security-groups (AWS CLI)
- Get-EC2StaleSecurityGroup (AWS Tools for Windows PowerShell)

次の例では、VPC A (vpc-aaaaaaaa) および VPC B がピアリング接続され、VPC ピアリング接続は削除されています。VPC A のセキュリティグループ sg-aaaa1111 は VPC B の sg-bbbb2222 を参照します。VPC に対して describe-stale-security-groups コマンドを実行すると、応答では、セキュリティグループ sg-aaaa1111 に、sg-bbbb2222 を参照する古くなった SSH ルールがあることが示されます。

```
aws ec2 describe-stale-security-groups --vpc-id vpc-aaaaaaaa
```

```
{
    "StaleSecurityGroupSet": [
        {
            "VpcId": "vpc-aaaaaaaa",
            "StaleIpPermissionsEgress": [],
            "GroupName": "Access1",
            "StaleIpPermissions": [
                {
```

```
                    "ToPort": 22,
                    "FromPort": 22,
                    "UserIdGroupPairs": [
                        {
                            "VpcId": "vpc-bbbbbbbb",
                            "PeeringStatus": "deleted",
                            "UserId": "123456789101",
                            "GroupName": "Prod1",
                            "VpcPeeringConnectionId": "pcx-b04deed9",
                            "GroupId": "sg-bbbb2222"
                        }
                    ],
                    "IpProtocol": "tcp"
                }
            ],
            "GroupId": "sg-aaaa1111",
            "Description": "Reference remote SG"
        }
    ]
}
```

古くなったセキュリティグループルールを特定した後、revoke-security-group-ingress コマンドまたは revoke-security-group-egress コマンドを使用してそれらのルールを削除できます。

VPC ピアリング接続の DNS 解決を有効にする

VPC ピアリング接続の DNS 設定により、VPC ピアリング接続を通過するリクエストに対してパブリック DNS ホスト名が解決される方法が決まります。VPC ピアリング接続の一方の側の EC2 インスタンスが、インスタンスのパブリック IPv4 DNS ホスト名を使用して、もう一方の EC2 インスタンスにリクエストを送信すると、DNS ホスト名は次のように解決されます。

DNS 解決が無効 (デフォルト)

パブリック IPv4 DNS ホスト名は、インスタンスのパブリック IPv4 アドレスに解決されます。

DNS 解決が有効

パブリック IPv4 DNS ホスト名は、インスタンスのプライベート IPv4 アドレスに解決されます。

要件

- 両方の VPC を、DNS ホスト名および DNS 解決に対して有効にする必要があります。詳細についてはAmazon VPC ユーザーガイドの「DNS attributes for your VPC」(VPC の DNS 属性) を参照してください。
- ピアリング接続は `active` 状態である必要があります。ピアリング接続を作成するときに、DNS 解決のサポートを有効にすることはできません。
- リクエスタ VPC の所有者はリクエスタ VPC ピアリングオプションを変更する必要があり、アクセプタ VPC の所有者はアクセプタ VPC ピアリングオプションを変更する必要があります。VPC が同じアカウントにある場合は、リクエスター VPC とアクセプター VPC の DNS 解決を同時に有効にすることができます。これは同じリージョンとクロスリージョンの両方の VPC ピアリング接続で機能します。

コンソールを使用してピアリング接続の DNS 解決を有効にするには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[Peering Connections] (ピアリング接続) をクリックします。
3. VPC ピアリング接続を選択します。
4. [アクション]、[DNS 設定を編集] の順に選択します。
5. リクエスタ VPC からのリクエストの DNS 解決を有効にするには、[リクエスタ DNS 解決]、[アクセプタVPCがリクエスタ VPC の DNS 解決を許可] を順に選択します。
6. アクセプタ VPC からのリクエストの DNS 解決を確認するには、[アクセプタ DNS 解決]、[リクエスタ VPC がアクセプタ VPC の DNS 解決を許可] の順に選択します。
7. [Save changes] (変更の保存) をクリックします。

コマンドラインを使用して DNS 解決を有効にするには

- modify-vpc-peering-connection-options (AWS CLI)
- Edit-EC2VpcPeeringConnectionOption (AWS Tools for Windows PowerShell)

コマンドラインを使用して VPC ピアリング接続を記述する

- describe-vpc-peering-connections ( ) AWS CLI」
- Get-EC2VpcPeeringConnection (AWS Tools for Windows PowerShell)

VPC ピアリング接続を削除する

ピアリング接続されている VPC の所有者は、どちらも VPC ピアリング接続をいつでも削除できます。リクエストした後でまだ pending-acceptance 状態にある VPC ピアリング接続も削除できます。

VPC ピアリング接続が rejected 状態のときは、VPC ピアリング接続を削除できません。自動的に接続が削除されます。

アクティブな VPC ピアリング接続の一部となっている Amazon VPC コンソールの VPC を削除すると、その VPC ピアリング接続も削除されます。別のアカウントにある VPC との VPC ピアリング接続をリクエストして、他の当事者がそのリクエストを承認する前に自分の VPC を削除した場合は、その VPC ピアリング接続も削除されます。別のアカウントの VPC からの pending-acceptance リクエストがある場合は、VPC を削除できません。最初に VPC ピアリング接続リクエストを拒否する必要があります。

ピアリング接続を削除すると、ステータスが Deleting に、その後 Deleted に設定されます。接続を削除すると、ピアリング接続の承認、拒否、編集ができなくなります。ピアリング接続がどの程度の期間表示されるかについて、詳しくは「VPC ピアリング接続のライフサイクル」を参照してください。

VPC ピアリング接続を削除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Peering Connections] (ピアリング接続) をクリックします。
3. VPC ピアリング接続を選択します。
4. [Actions] (アクション)、[Delete peering connection] (ピアリング接続を削除) を選択します。
5. 確認を求められたら、**delete**と入力し、[削除] を選択します。

コマンドラインを使用して VPC ピアリング接続を削除するには

- delete-vpc-peering-connection ( ) AWS CLI」
- Remove-EC2VpcPeeringConnection」(AWS Tools for Windows PowerShell)

VPC ピアリング接続のトラブルシューティング

ピア VPC のリソースから VPC 内のリソースに接続できない場合は、次の手順を実行してください。

- 各 VPC 内の各リソースについて、サブネットのルートテーブルに、ピア VPC 宛てのトラフィックを VPC ピアリング接続に送信するルートが含まれていることを確認します。この結果、ネットワークトラフィックが 2 つの VPC 間を正常に流れることができるようになります。詳細については、「ルートテーブルの更新」を参照してください。
- 関連する EC2 インスタンスについては、そのインスタンスのセキュリティグループで、ピア VPC からのインバウンドおよびアウトバウンドトラフィックが許可されていることを確認します。セキュリティグループのルールは、EC2 インスタンスへのアクセスを許可するトラフィックを制御します。詳細については、「ピアセキュリティグループの参照」を参照してください。
- リソースを含むサブネット ACL のネットワーク ACL で、ピア VPC からの必要なトラフィックが許可されていることを確認します。ネットワーク ACL は、サブネットレベルでトラフィックをフィルタリングする追加のセキュリティレイヤーです。

それでも問題が解決しない場合は、Reachability Analyzer を利用できます。Reachability Analyzer は、2 つの VPC 間の接続問題の原因となっている特定のコンポーネント (ルートテーブル、セキュリティグループ、またはネットワーク ACL) を特定するのに役立ちます。詳細については、「Reachability Analyzer Guide」(到達可能性アナライザーガイド) を参照してください。

VPC ネットワーク設定の徹底的な検証は、発生する可能性のある VPC ピアリング接続の問題のトラブルシューティングと解決の鍵となります。

一般的な VPC ピアリング接続設定

このセクションでは、実装できる一般的なタイプの VPC ピアリング設定 2 つについて説明します。

- VPC 全体へのルートを持つ VPC ピアリング設定: この設定では、各 VPC のルートテーブルに、ピア VPC 宛てのすべてのトラフィックを VPC ピアリング接続に送信するルートを作成します。この結果、1 つの VPC 内の任意のリソースがピア VPC 内の任意のリソースと通信できるようになり、管理が簡素化されます。ただし、VPC 間のすべてのトラフィックがピアリング接続を通過するので、トラフィック量が多いとボトルネックになる可能性があるということにもなります。
- 特定のルートを持つ VPC ピアリング設定: または、ピア VPC 内の特定のサブネットまたはリソースにのみトラフィックを送信する、より詳細なルートを各 VPC のルートテーブルに作成することもできます。そうすると、ピアリング接続を通過するトラフィックを必要なもののみに制限できるため、より効率的になる可能性があります。ただし、通信が必要なピア VPC に新しいリソースを追加するたびにルートテーブルを更新する必要があるため、より多くのメンテナンスが必要になります。

最善のアプローチは、VPC アーキテクチャのサイズと複雑さ、VPC 間での想定されるトラフィックの量、セキュリティとリソースアクセスに関する組織のニーズなどの要因によって異なります。多くの企業はハイブリッドアプローチを用いていて、一般的なトラフィックパターンには幅広いルートを、より機密性の高いユースケースや帯域幅を大量に消費するユースケースには特定のルートを使用しています。

設定

- VPC 全体にルーティングする VPC ピアリング設定
- 特定のルートを使用する VPC ピアリング設定

VPC 全体にルーティングする VPC ピアリング設定

ルートテーブルにピア VPC の CIDR ブロック全体へのアクセスが含まれるように、VPC ピア接続を設定できます。特定の VPC ピアリング接続設定が必要になる可能性があるシナリオの詳細については「VPC ピアリング接続のネットワーキングのシナリオ」を参照してください。VPC ピアリング接続を作成して作業する方法の詳細については、「VPC ピアリング接続」を参照してください。

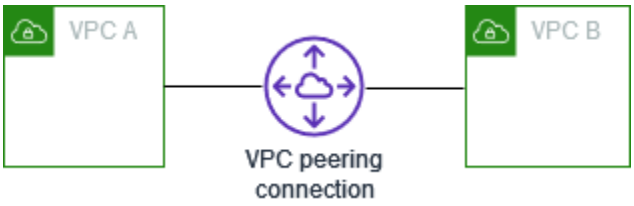
ルートテーブルの更新の詳細については、「VPC ピアリング接続のルートテーブルを更新する」を参照してください。

Configurations

- 2 つの VPC が相互にピアリング接続
- 1 つの VPC が 2 つの VPC とピアリング接続
- 3 つの VPC が相互にピアリング接続
- 多数の VPC が相互にピアリング接続

2 つの VPC が相互にピアリング接続

この設定では、VPC A と VPC B (pcx-11112222) との間にピアリング接続があります。VPC は、同じ AWS アカウント に存在し、CIDR ブロックは重複していません。



互いのリソースへのアクセスを必要とする VPC が 2 つある場合に、この設定を使用することがあります。例えば、会計記録用の VPC A、財務記録用の VPC B をセットアップし、これらの各 VPC が制限なしで他方の VPC のリソースにアクセスできるようにする必要がある場合です。

単一の VPC CIDR

ピア VPC の CIDR ブロックのトラフィックを VPC ピアリング接続に送信するルートで、各 VPC のルートテーブルを更新します。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A CIDR*	ローカル
	VPC B CIDR	pcx-11112222
VPC B	*VPC B CIDR*	ローカル
	VPC A CIDR	pcx-11112222

複数の IPv4 VPC CIDR

VPC A と VPC B に複数の IPv4 CIDR ブロックが関連付けられている場合は、ピア VPC の IPv4 CIDR ブロックの一部またはすべてのルートを使用して、各 VPC のルートテーブルを更新できます。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A CIDR 1*	ローカル
	VPC A CIDR 2	ローカル
	VPC B CIDR 1	pcx-11112222
	VPC B CIDR 2	pcx-11112222
VPC B	*VPC B CIDR 1*	ローカル
	VPC B CIDR 2	ローカル
	VPC A CIDR 1	pcx-11112222
	VPC A CIDR 2	pcx-11112222

IPv4 および IPv6 VPC CIDR

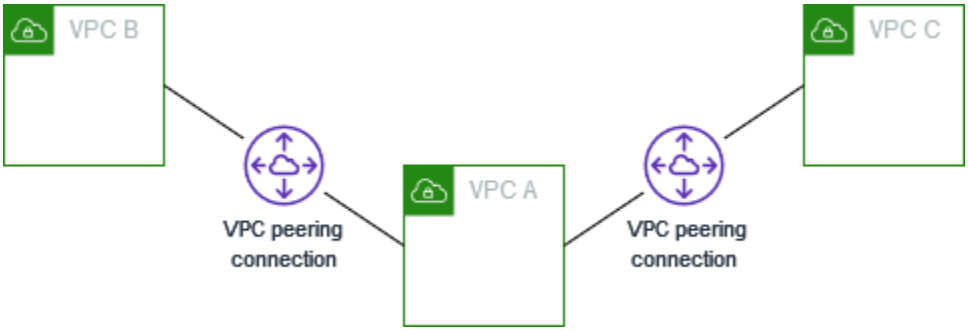
VPC A と VPC B に IPv6 CIDR ブロックが関連付けられている場合は、ピア VPC の IPv4 CIDR ブロックと IPv6 CIDR ブロックの両方のルートを使用して、各 VPC のルートテーブルを更新できます。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A IPv4 CIDR*	ローカル
	VPC A IPv6 CIDR	ローカル
	VPC B IPv4 CIDR	pcx-11112222
	VPC B IPv6 CIDR	pcx-11112222
VPC B	*VPC B IPv4 CIDR*	ローカル

ルートテーブル	送信先	ターゲット
	VPC B IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-11112222
	VPC A IPv6 CIDR	pcx-11112222

1 つの VPC が 2 つの VPC とピアリング接続

この設定では、中央 VPC (VPC A)、VPC A と VPC B との間のピアリング接続 (pcx-12121212)、VPC A と VPC C との間のピアリング接続 (pcx-23232323) があります。これら 3 つの VPC は同じ AWS アカウント に存在し、CIDR ブロックは重複していません。



VPC ピアリングは推移的なピアリングをサポートしていないため、VPC B と VPC C は VPC A を経由して相互に直接トラフィックを送信することはできません。「3 つの VPC が相互にピアリング接続」に示すように、VPC B と VPC C の間に VPC ピアリング接続を作成できます。サポートされないピアリング接続のシナリオの詳細については、「the section called “VPC ピアリングの制限事項”」を参照してください。

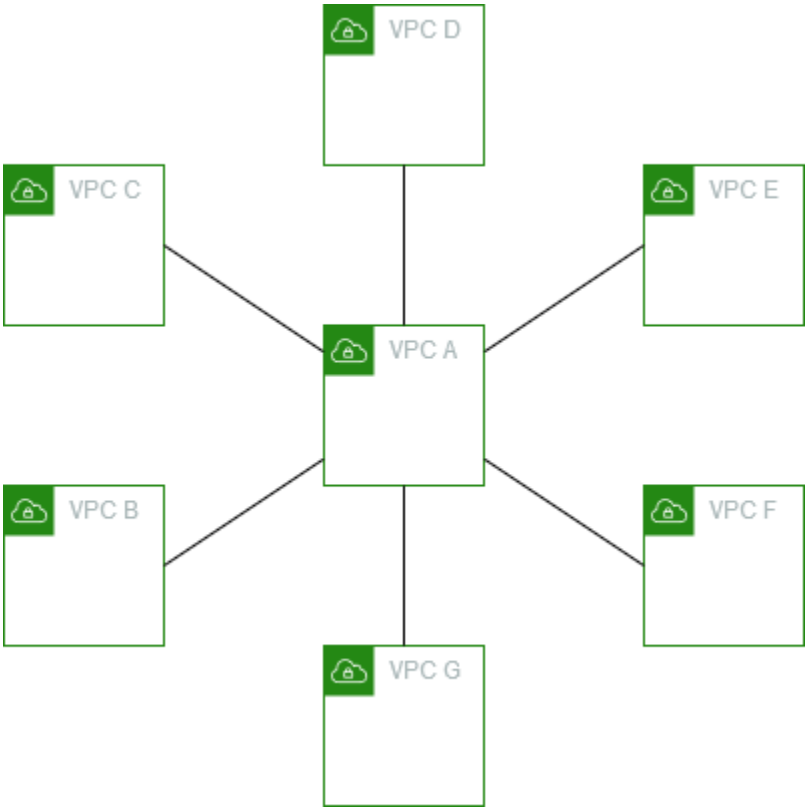
サービスのリポジトリのように、他の VPC がアクセスする必要のあるリソースが中央 VPC に存在する場合に、この設定を使用することがあります。他の VPC は、互いのリソースにアクセスする必要がありません。中央 VPC のリソースに対するアクセスのみが必要です。

VPC ごとに 1 つの CIDR ブロックを使用してこの設定を実装するには、各 VPC のルートテーブルを次のように更新します。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A CIDR*	ローカル

ルートテーブル	送信先	ターゲット
	VPC B CIDR	pcx-12121212
	VPC C CIDR	pcx-23232323
VPC B	*VPC B CIDR*	ローカル
	VPC A CIDR	pcx-12121212
VPC C	*VPC C CIDR*	ローカル
	VPC A CIDR	pcx-23232323

この設定は追加の VPC に拡張できます。例えば、VPC A は、IPv4 CIDR と IPv6 CIDR の両方を使用して VPC B ~ VPC G とピアリング接続していますが、他の VPC は相互にピアリング接続していません。この図では、線は VPC ピアリング接続を表しています。



次のようにルートテーブルを更新します。

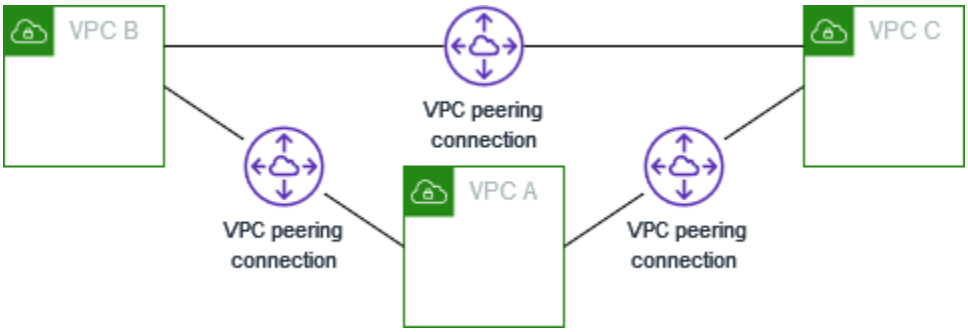
ルートテーブル	送信先	ターゲット
VPC A	*VPC A IPv4 CIDR*	ローカル
	VPC A IPv6 CIDR	ローカル
	VPC B IPv4 CIDR	pcx-aaaabbbb
	VPC B IPv6 CIDR	pcx-aaaabbbb
	VPC C IPv4 CIDR	pcx-aaaacccc
	VPC C IPv6 CIDR	pcx-aaaacccc
	VPC D IPv4 CIDR	pcx-aaaadddd
	VPC D IPv6 CIDR	pcx-aaaadddd
	VPC E IPv4 CIDR	pcx-aaaaeeee
	VPC E IPv6 CIDR	pcx-aaaaeeee
	VPC F IPv4 CIDR	pcx-aaaaffff
	VPC F IPv6 CIDR	pcx-aaaaffff
	VPC G IPv4 CIDR	pcx-aaaagggg
	VPC G IPv6 CIDR	pcx-aaaagggg
VPC B	*VPC B IPv4 CIDR*	ローカル
	VPC B IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-aaaabbbb
	VPC A IPv6 CIDR	pcx-aaaabbbb
VPC C	*VPC C IPv4 CIDR*	ローカル
	VPC C IPv6 CIDR	ローカル

ルートテーブル	送信先	ターゲット
	VPC A IPv4 CIDR	pcx-aaaacccc
	VPC A IPv6 CIDR	pcx-aaaacccc
VPC D	*VPC D IPv4 CIDR*	ローカル
	VPC D IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-aaaadddd
	VPC A IPv6 CIDR	pcx-aaaadddd
VPC E	*VPC E IPv4 CIDR*	ローカル
	VPC E IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-aaaaeeee
	VPC A IPv6 CIDR	pcx-aaaaeeee
VPC F	*VPC F IPv4 CIDR*	ローカル
	VPC F IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-aaaaffff
	VPC A IPv6 CIDR	pcx-aaaaffff
VPC G	*VPC G IPv4 CIDR*	ローカル
	VPC G IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-aaaagggg
	VPC A IPv6 CIDR	pcx-aaaagggg

3 つの VPC が相互にピアリング接続

この設定では、同じ AWS アカウント 内に 3 つの VPC があり、CIDR ブロックは重複していません。VPC は次のようにフルメッシュでピアリングされます。

- VPC A は VPC ピアリング接続 pcx-aaaabbbb により VPC B にピアリング接続しています。
- VPC A は VPC ピアリング接続 pcx-aaaacccc により VPC C にピアリング接続しています。
- VPC B は VPC ピアリング接続 pcx-bbbbcccc により VPC C にピアリング接続しています。



この設定は、VPC 間で制限なしで相互にリソースを共有する必要がある場合に使用できます。例えば、ファイル共有システムの場合です。

この設定を実装するには、各 VPC のルートテーブルを次のように更新します。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A CIDR*	ローカル
	VPC B CIDR	pcx-aaaabbbb
	VPC C CIDR	pcx-aaaacccc
VPC B	*VPC B CIDR*	ローカル
	VPC A CIDR	pcx-aaaabbbb
	VPC C CIDR	pcx-bbbbcccc
VPC C	*VPC C CIDR*	ローカル
	VPC A CIDR	pcx-aaaacccc

ルートテーブル	送信先	ターゲット
	VPC B CIDR	pcx-bbbbcccc

VPC A と VPC B に IPv4 CIDR ブロックと IPv6 CIDR ブロックの両方があっても、VPC C には IPv6 CIDR ブロックがない場合は、次のようにルートテーブルを更新します。VPC A と VPC B のリソースは、VPC ピアリング接続を介して IPv6 経由で通信できます。ただし VPC C は、IPv6 を経由しても、VPC A または VPC B のいずれとも通信できません。

ルートテーブル	デスティネーション	ターゲット
VPC A	*VPC A IPv4 CIDR*	ローカル
	VPC A IPv6 CIDR	ローカル
	VPC B IPv4 CIDR	pcx-aaaabbbb
	VPC B IPv6 CIDR	pcx-aaaabbbb
	VPC C IPv4 CIDR	pcx-aaaacccc
VPC B	*VPC B IPv4 CIDR*	ローカル
	VPC B IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-aaaabbbb
	VPC A IPv6 CIDR	pcx-aaaabbbb
	VPC C IPv4 CIDR	pcx-bbbbcccc
VPC C	*VPC C IPv4 CIDR*	ローカル
	VPC A IPv4 CIDR	pcx-aaaacccc
	VPC B IPv4 CIDR	pcx-bbbbcccc

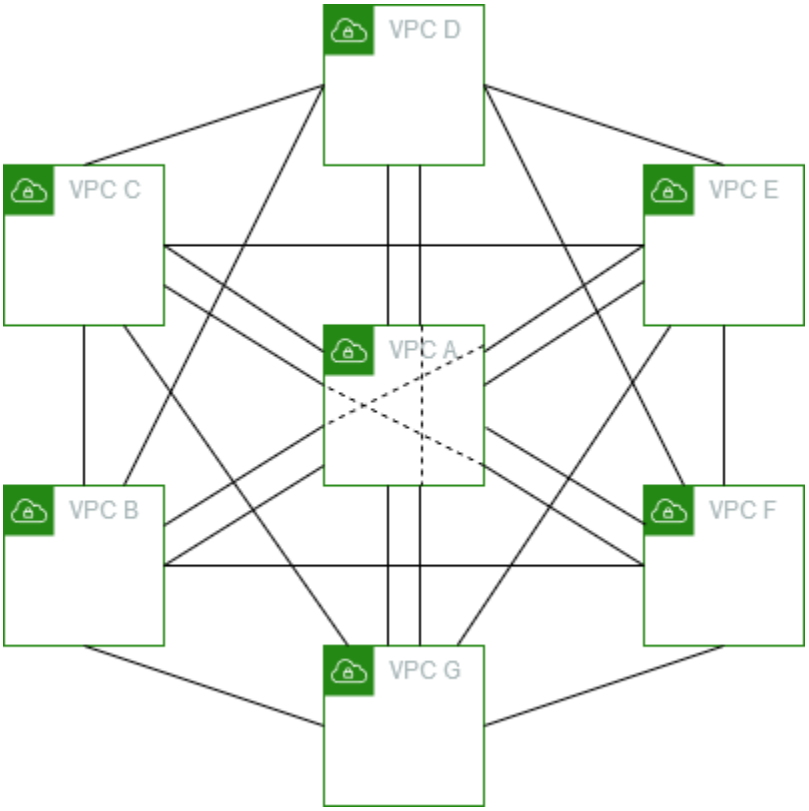
多数の VPC が相互にピアリング接続

この設定では、フルメッシュ設定で 7 つの VPC がピアリング接続されています。VPC は、同じ AWS アカウント に存在し、CIDR ブロックは重複していません。

VPC	VPC	VPC ピア接続
A	B	pcx-aaaabbbb
A	C	pcx-aaaacccc
A	D	pcx-aaaadddd
A	E	pcx-aaaaeeee
A	F	pcx-aaaaffff
A	G	pcx-aaaagggg
B	C	pcx-bbbbcccc
B	D	pcx-bbbbdddd
B	E	pcx-bbbbeeee
B	F	pcx-bbbbffff
B	G	pcx-bbbbgggg
C	D	pcx-ccccdddd
C	E	pcx-cccceeee
C	F	pcx-ccccffff
C	G	pcx-ccccgggg
D	E	pcx-ddddeeee
D	F	pcx-ddddffff
D	G	pcx-ddddgggg

VPC	VPC	VPC ピア接続
E	F	pcx-eeeeffff
E	G	pcx-eeeegggg
F	G	pcx-ffffgggg

この設定は、制限なしで相互のリソースにアクセスする必要のある VPC が複数ある場合に使用できます。例えば、ファイル共有ネットワークの場合です。この図では、線は VPC ピアリング接続を表しています。



この設定を実装するには、各 VPC のルートテーブルを次のように更新します。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A CIDR*	ローカル
	VPC B CIDR	pcx-aaaabbbb

ルートテーブル	送信先	ターゲット
	VPC C CIDR	pcx-aaaacccc
	VPC D CIDR	pcx-aaaadddd
	VPC E CIDR	pcx-aaaaeeee
	VPC F CIDR	pcx-aaaaffff
	VPC G CIDR	pcx-aaaagggg
VPC B	*VPC B CIDR*	ローカル
	VPC A CIDR	pcx-aaaabbbb
	VPC C CIDR	pcx-bbbbcccc
	VPC D CIDR	pcx-bbbbdddd
	VPC E CIDR	pcx-bbbbeeee
	VPC F CIDR	pcx-bbbbffff
	VPC G CIDR	pcx-bbbbgggg
VPC C	*VPC C CIDR*	ローカル
	VPC A CIDR	pcx-aaaacccc
	VPC B CIDR	pcx-bbbbcccc
	VPC D CIDR	pcx-ccccdddd
	VPC E CIDR	pcx-ccceeeee
	VPC F CIDR	pcx-ccccffff
	VPC G CIDR	pcx-ccccgggg
VPC D	*VPC D CIDR*	ローカル

ルートテーブル	送信先	ターゲット
	VPC A CIDR	pcx-aaaadddd
	VPC B CIDR	pcx-bbbbdddd
	VPC C CIDR	pcx-ccccdddd
	VPC E CIDR	pcx-ddddeeee
	VPC F CIDR	pcx-ddddffff
	VPC G CIDR	pcx-ddddgggg
VPC E	*VPC E CIDR*	ローカル
	VPC A CIDR	pcx-aaaaeeee
	VPC B CIDR	pcx-bbbbeeee
	VPC C CIDR	pcx-cccceeee
	VPC D CIDR	pcx-ddddeeee
	VPC F CIDR	pcx-eeeeffff
	VPC G CIDR	pcx-eeeegggg
VPC F	*VPC F CIDR*	ローカル
	VPC A CIDR	pcx-aaaaffff
	VPC B CIDR	pcx-bbbbffff
	VPC C CIDR	pcx-ccccffff
	VPC D CIDR	pcx-ddddffff
	VPC E CIDR	pcx-eeeeffff
	VPC G CIDR	pcx-ffffgggg

ルートテーブル	送信先	ターゲット
VPC G	*VPC G CIDR*	ローカル
	VPC A CIDR	pcx-aaaagggg
	VPC B CIDR	pcx-bbbbgggg
	VPC C CIDR	pcx-ccccgggg
	VPC D CIDR	pcx-ddddgggg
	VPC E CIDR	pcx-eeeegggg
	VPC F CIDR	pcx-ffffgggg

すべての VPC が IPv6 CIDR ブロックと関連付けられている場合は、次のようにルートテーブルを更新します。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A IPv4 CIDR*	ローカル
	VPC A IPv6 CIDR	ローカル
	VPC B IPv4 CIDR	pcx-aaaabbbb
	VPC B IPv6 CIDR	pcx-aaaabbbb
	VPC C IPv4 CIDR	pcx-aaaacccc
	VPC C IPv6 CIDR	pcx-aaaacccc
	VPC D IPv4 CIDR	pcx-aaaadddd
	VPC D IPv6 CIDR	pcx-aaaadddd
	VPC E IPv4 CIDR	pcx-aaaaeeee
	VPC E IPv6 CIDR	pcx-aaaaeeee

ルートテーブル	送信先	ターゲット
	VPC F IPv4 CIDR	pcx-aaaaffff
	VPC F IPv6 CIDR	pcx-aaaaffff
	VPC G IPv4 CIDR	pcx-aaaagggg
	VPC G IPv6 CIDR	pcx-aaaagggg
VPC B	*VPC B IPv4 CIDR*	ローカル
	VPC B IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-aaaabbbb
	VPC A IPv6 CIDR	pcx-aaaabbbb
	VPC C IPv4 CIDR	pcx-bbbbcccc
	VPC C IPv6 CIDR	pcx-bbbbcccc
	VPC D IPv4 CIDR	pcx-bbbbdddd
	VPC D IPv6 CIDR	pcx-bbbbdddd
	VPC E IPv4 CIDR	pcx-bbbbeeee
	VPC E IPv6 CIDR	pcx-bbbbeeee
	VPC F IPv4 CIDR	pcx-bbbbffff
	VPC F IPv6 CIDR	pcx-bbbbffff
	VPC G IPv4 CIDR	pcx-bbbbgggg
	VPC G IPv6 CIDR	pcx-bbbbgggg
VPC C	*VPC C IPv4 CIDR*	ローカル
	VPC C IPv6 CIDR	ローカル

ルートテーブル	送信先	ターゲット
	VPC A IPv4 CIDR	pcx-aaaacccc
	VPC A IPv6 CIDR	pcx-aaaacccc
	VPC B IPv4 CIDR	pcx-bbbbcccc
	VPC B IPv6 CIDR	pcx-bbbbcccc
	VPC D IPv4 CIDR	pcx-ccccdddd
	VPC D IPv6 CIDR	pcx-ccccdddd
	VPC E IPv4 CIDR	pcx-cccceeee
	VPC E IPv6 CIDR	pcx-cccceeee
	VPC F IPv4 CIDR	pcx-ccccffff
	VPC F IPv6 CIDR	pcx-ccccffff
	VPC G IPv4 CIDR	pcx-ccccgggg
	VPC G IPv6 CIDR	pcx-ccccgggg
VPC D	*VPC D IPv4 CIDR*	ローカル
	VPC D IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-aaaadddd
	VPC A IPv6 CIDR	pcx-aaaadddd
	VPC B IPv4 CIDR	pcx-bbbbdddd
	VPC B IPv6 CIDR	pcx-bbbbdddd
	VPC C IPv4 CIDR	pcx-ccccdddd
	VPC C IPv6 CIDR	pcx-ccccdddd

ルートテーブル	送信先	ターゲット
	VPC E IPv4 CIDR	pcx-ddddeeee
	VPC E IPv6 CIDR	pcx-ddddeeee
	VPC F IPv4 CIDR	pcx-ddddffff
	VPC F IPv6 CIDR	pcx-ddddffff
	VPC G IPv4 CIDR	pcx-ddddgggg
	VPC G IPv6 CIDR	pcx-ddddgggg
VPC E	*VPC E IPv4 CIDR*	ローカル
	VPC E IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-aaaaeeee
	VPC A IPv6 CIDR	pcx-aaaaeeee
	VPC B IPv4 CIDR	pcx-bbbbeeee
	VPC B IPv6 CIDR	pcx-bbbbeeee
	VPC C IPv4 CIDR	pcx-cccceeee
	VPC C IPv6 CIDR	pcx-cccceeee
	VPC D IPv4 CIDR	pcx-ddddeeee
	VPC D IPv6 CIDR	pcx-ddddeeee
	VPC F IPv4 CIDR	pcx-eeeeffff
	VPC F IPv6 CIDR	pcx-eeeeffff
	VPC G IPv4 CIDR	pcx-eeeegggg
	VPC G IPv6 CIDR	pcx-eeeegggg

ルートテーブル	送信先	ターゲット
VPC F	*VPC F IPv4 CIDR*	ローカル
	VPC F IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-aaaaffff
	VPC A IPv6 CIDR	pcx-aaaaffff
	VPC B IPv4 CIDR	pcx-bbbbffff
	VPC B IPv6 CIDR	pcx-bbbbffff
	VPC C IPv4 CIDR	pcx-ccccffff
	VPC C IPv6 CIDR	pcx-ccccffff
	VPC D IPv4 CIDR	pcx-ddddffff
	VPC D IPv6 CIDR	pcx-ddddffff
	VPC E IPv4 CIDR	pcx-eeeeffff
	VPC E IPv6 CIDR	pcx-eeeeffff
	VPC G IPv4 CIDR	pcx-ffffgggg
	VPC G IPv6 CIDR	pcx-ffffgggg
VPC G	*VPC G IPv4 CIDR*	ローカル
	VPC G IPv6 CIDR	ローカル
	VPC A IPv4 CIDR	pcx-aaaagggg
	VPC A IPv6 CIDR	pcx-aaaagggg
	VPC B IPv4 CIDR	pcx-bbbbgggg
	VPC B IPv6 CIDR	pcx-bbbbgggg

ルートテーブル	送信先	ターゲット
	VPC C IPv4 CIDR	pcx-ccccgggg
	VPC C IPv6 CIDR	pcx-ccccgggg
	VPC D IPv4 CIDR	pcx-ddddgggg
	VPC D IPv6 CIDR	pcx-ddddgggg
	VPC E IPv4 CIDR	pcx-eeeegggg
	VPC E IPv6 CIDR	pcx-eeeegggg
	VPC F IPv4 CIDR	pcx-ffffgggg
	VPC F IPv6 CIDR	pcx-ffffgggg

特定のルートを使用する VPC ピアリング設定

VPC ピアリング接続のルートテーブルは、サブネット CIDR ブロック、特定の CIDR ブロック (VPC に複数の CIDR ブロックがある場合)、またはピア VPC 内に存在する特定のリソースへのアクセスを制限するように設定できます。この例では、中央 VPC は重複した CIDR ブロックがある少なくとも 2 つの VPC にピアリング接続されます。

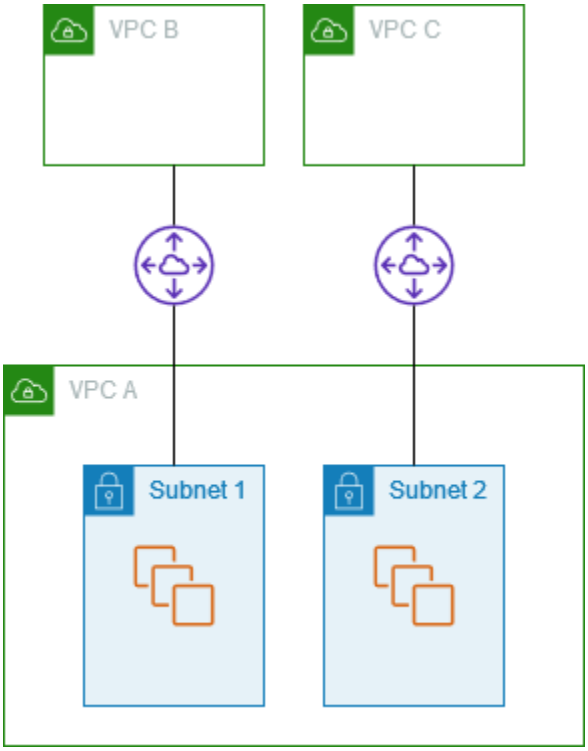
特定の VPC ピアリング接続設定が必要になる可能性があるシナリオの例については、「VPC ピアリング接続のネットワーキングのシナリオ」を参照してください。VPC ピアリング接続の操作方法については、「VPC ピアリング接続」を参照してください。ルートテーブルの更新の詳細については、「VPC ピアリング接続のルートテーブルを更新する」を参照してください。

設定

- 1 つの VPC にある特定のサブネットにアクセスする 2 つの VPC
- 1 つの VPC にある特定の CIDR ブロックにアクセスする 2 つの VPC
- 2 つの VPC にある特定のサブネットにアクセスする 1 つの VPC
- 2 つの VPC にある特定のインスタンスにアクセスする、1 つの VPC にあるインスタンス
- 最長のプレフィックスの一致を使用して 2 つの VPC にアクセスする 1 つの VPC
- 多重 VPC 設定

1 つの VPC にある特定のサブネットにアクセスする 2 つの VPC

この設定では、2 つのサブネットを持つ中央 VPC (VPC A)、VPC A と VPC B との間の VPC ピアリング接続 (pcx-aaaabbbb)、VPC A と VPC C との間のピアリング接続 (pcx-aaaacccc) があります。各 VPC は VPC A 内の 1 つのサブネット内のリソースにのみアクセスする必要があります。



サブネット 1 のルートテーブルは、VPC ピアリング接続 pcx-aaaabbbb を使用して VPC B の CIDR ブロック全体にアクセスします。VPC B のルートテーブルは、pcx-aaaabbbb を使用して VPC A にあるサブネット 1 の CIDR ブロックにアクセスします。サブネット 2 のルートテーブルは、VPC ピアリング接続 pcx-aaaacccc を使用して VPC C の CIDR ブロック全体にアクセスします。VPC C のルートテーブルは、pcx-aaaacccc を使用して VPC A にあるサブネット 2 の CIDR ブロックにアクセスします。

ルートテーブル	送信先	ターゲット
サブネット 1 (VPC A)	*VPC A CIDR*	ローカル
	VPC B CIDR	pcx-aaaabbbb
サブネット 2 (VPC A)	*VPC A CIDR*	ローカル
	VPC C CIDR	pcx-aaaacccc

ルートテーブル	送信先	ターゲット
VPC B	*VPC B CIDR*	ローカル
	##### 1 CIDR	pcx-aaaabbbb
VPC C	*VPC C CIDR*	ローカル
	##### 2 CIDR	pcx-aaaacccc

この設定は複数の CIDR ブロックに拡張できます。VPC A と VPC BB に IPv4 CIDR ブロックと IPv6 CIDR ブロックの両方があり、サブネット 1 には IPv6 CIDR ブロックが関連付けられているとします。VPC ピアリング接続を使用して、IPv6 を介して VPC A のサブネット 1 と VPC B との通信を有効にできます。これを行うには、VPC A のルートテーブルに VPC B の IPv6 CIDR ブロックの宛先ルートを追加し、VPC B のルートテーブルに VPC A のサブネット 1 の IPv6 CIDR の宛先ルートを追加します。

ルートテーブル	送信先	ターゲット	注意事項
VPC A のサブネット 1	*VPC A IPv4 CIDR*	ローカル	
	VPC A IPv6 CIDR	ローカル	VPC 内の IPv6 通信に対して自動的に追加されるローカルルートです。
	VPC B IPv4 CIDR	pcx-aaaabbbb	
	VPC B IPv6 CIDR	pcx-aaaabbbb	VPC B の IPv6 CIDR ブロックへのルートです。
VPC A のサブネット 2	*VPC A IPv4 CIDR*	ローカル	
	VPC A IPv6 CIDR	ローカル	VPC 内の IPv6 通信に対して自動的に追加されるローカルルートです。

ルートテーブル	送信先	ターゲット	注意事項
	VPC C IPv4 CIDR	pcx-aaaacccc	
VPC B	*VPC B IPv4 CIDR*	ローカル	
	VPC B IPv6 CIDR	ローカル	VPC 内の IPv6 通信に対して自動的に追加されるローカルルートです。
	##### 1 IPv4 CIDR	pcx-aaaabbbb	
	##### 1 IPv6 CIDR	pcx-aaaabbbb	VPC A の IPv6 CIDR ブロックへのルートです。
VPC C	*VPC C IPv4 CIDR*	ローカル	
	##### 2 IPv4 CIDR	pcx-aaaacccc	

1 つの VPC にある特定の CIDR ブロックにアクセスする 2 つの VPC

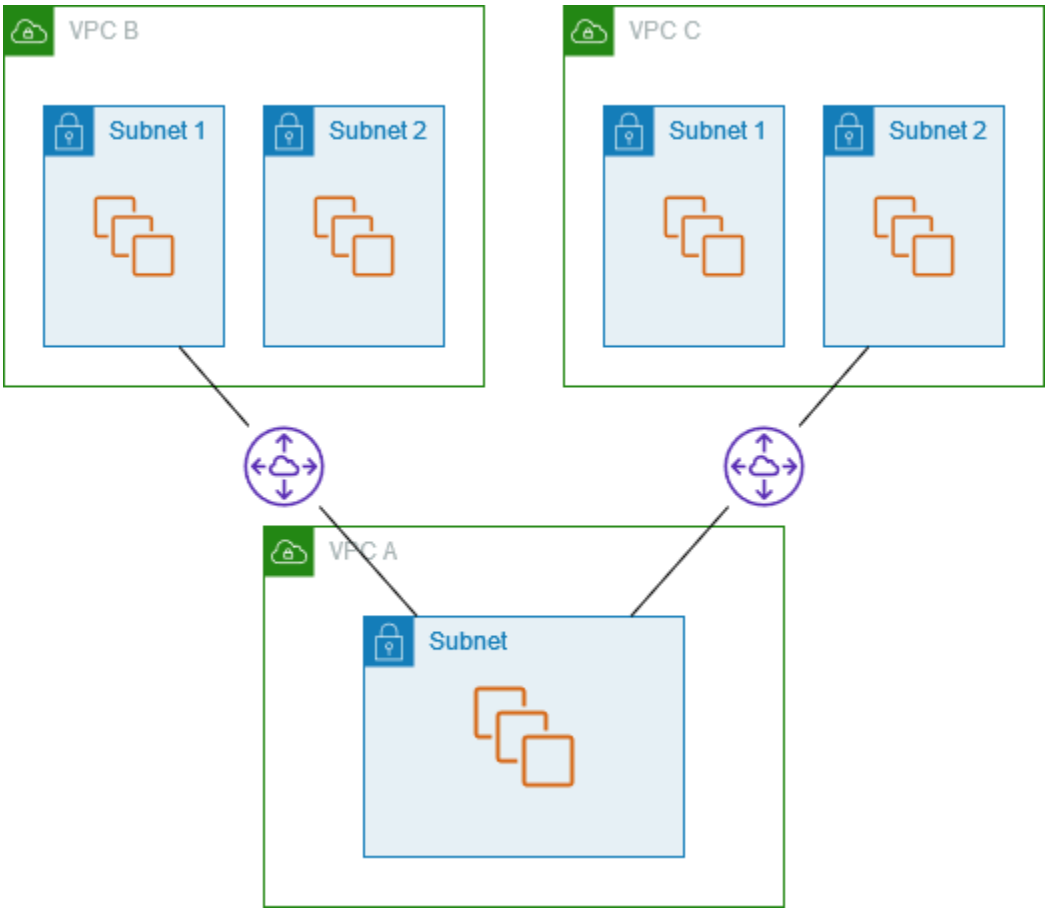
この設定では、中央 VPC (VPC A)、VPC A と VPC B との間のピアリング接続 (pcx-aaaabbbb)、VPC A と VPC C との間のピアリング接続 (pcx-aaaacccc) があります。VPC A は、ピアリング接続ごとに CIDR ブロックを 1 つ持ちます。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A CIDR 1*	ローカル
	VPC A CIDR 2	ローカル
	VPC B CIDR	pcx-aaaabbbb
	VPC C CIDR	pcx-aaaacccc

ルートテーブル	送信先	ターゲット
VPC B	*VPC B CIDR*	ローカル
	VPC A CIDR 1	pcx-aaaabbbb
VPC C	*VPC C CIDR*	ローカル
	VPC A CIDR 2	pcx-aaaacccc

2 つの VPC にある特定のサブネットにアクセスする 1 つの VPC

この設定では、1 つのサブネットを持つ中央 VPC (VPC A)、VPC A と VPC B との間のピアリング接続 (pcx-aaaabbbb)、VPC A と VPC C との間のピアリング接続 (pcx-aaaacccc) があります。VPC B と VPC C には、それぞれ 2 つのサブネットがあります。VPC A と VPC B との間のピアリング接続では、VPC B 内にあるサブネットのうち 1 つだけを使用します。VPC A と VPC C との間のピアリング接続では、VPC C 内にあるサブネットのうち 1 つだけを使用します。



Active Directory サービスなど、他の VPC がアクセスする必要のある単一のリソースセットを持つ中央 VPC がある場合に、この設定を使用します。中央 VPC は、ピアリング接続された VPC にフルアクセスする必要はありません。

VPC A のルートテーブルは、ピアリング接続を使用して、ピアリング接続された VPC 内の特定のサブネットにのみアクセスします。サブネット 1 のルートテーブルは VPC A とのピアリング接続を使用して VPC A 内のサブネットにアクセスします。サブネット 2 のルートテーブルは VPC A とのピアリング接続を使用して VPC A 内サブネットにアクセスします。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A CIDR*	ローカル
	##### 1 CIDR	pcx-aaaabbbb
	##### 2 CIDR	pcx-aaaacccc
サブネット 1 (VPC B)	*VPC B CIDR*	ローカル
	VPC A CIDR ######	pcx-aaaabbbb
サブネット 2 (VPC C)	*VPC C CIDR*	ローカル
	VPC A CIDR ######	pcx-aaaacccc

レスポンストラフィックのルーティング

CIDR ブロックが重複または一致している複数の VPC にピアリング接続された VPC がある場合は、ルートテーブルが VPC からのレスポンストラフィックを間違った VPC に送信しないように設定されていることを確認します。AWS の VPC ピアリング接続では、パケットのソース IP を確認してリプライパケットをソースにルーティングするユニキャストリバースパス転送をサポートしていません。

たとえば、VPC A は VPC B と VPC C にピアリング接続されます。VPC B と VPC C は CIDR ブロックが一致し、そのサブネットは CIDR ブロックが一致しています。VPC B のサブネット 2 のルートテーブルは、VPC A のサブネットにアクセスする VPC ピアリング接続 `pcx-aaaabbbb` を指します。VPC A のルートテーブルは、VPC CIDR 宛のトラフィックをピアリング接続 `pcx-aaaaccccc` に送信するように設定されています。

ルートテーブル	送信先	ターゲット
サブネット 2 (VPC B)	*VPC B CIDR*	ローカル
	VPC A CIDR ######	pcx-aaaabbbb
VPC A	*VPC A CIDR*	ローカル
	VPC C CIDR	pcx-aaaacccc

VPC B のサブネット 2 のインスタンスが、VPC ピアリング接続 pcx-aaaabbbb を使用して VPC A の Active Directory サーバーにトラフィックを送信するとします。VPC A はレスポンストラフィックを Active Directory サーバーに送信します。ただし、VPC A ルートテーブルは、VPC CIDR 範囲内のすべてのトラフィックを、VPC ピアリング接続 pcx-aaaacccc に送信するよう設定されています。VPC C のサブネット 2 に VPC B のサブネット 2 内のインスタンスと同じ IP アドレスを持つインスタンスがある場合、VPC A からレスポンストラフィックを受信します。VPC B のサブネット 2 は、VPC A へのリクエストに対するレスポンスを受信しません。

これを防ぐには、送信先を VPC B のサブネット 2 の CIDR、ターゲットを pcx-aaaabbbb として特定のルートを VPC A のルートテーブルに追加できます。新しいルートはより具体的であるため、サブネット 2 CIDR 宛のトラフィックは VPC ピアリング接続 pcx-aaaabbbb にルーティングされます。

または、次の例で、VPC A ルートテーブルには、各 VPC ピアリング接続の各サブネット用のルートがあります。VPC A は、VPC B のサブネット 2 および VPC C のサブネット 1 と通信できます。このシナリオは、VPC B および VPC C と同じアドレス範囲内にある、別のサブネットとの別の VPC ピアリング接続を追加する必要があるときに便利です。その特定のサブネット用に別のルートを追加するだけです。

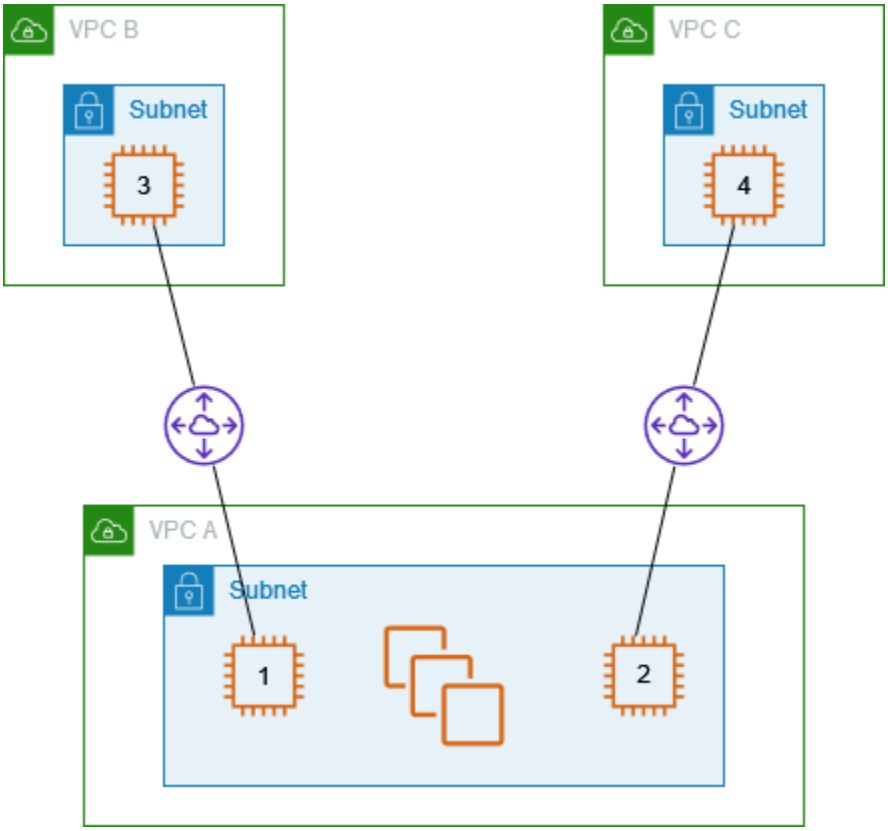
ルーティング先	ターゲット
VPC A CIDR	ローカル
##### 2 CIDR	pcx-aaaabbbb
##### 1 CIDR	pcx-aaaacccc

または、ユースケースに応じて、VPC B の特定の IP アドレスへのルートを作成して、トラフィックが正しいサーバーに戻されるようにします (ルートテーブルでは、プレフィックス最長一致を使用して、ルートの優先順位が決定されます)。

ルーティング先	ターゲット
VPC A CIDR	ローカル
##### 2 #### IP ####	pcx-aaaabbbb
VPC B CIDR	pcx-aaaacccc

2 つの VPC にある特定のインスタンスにアクセスする、1 つの VPC にあるインスタンス

この設定では、1 つのサブネットを持つ中央 VPC (VPC A)、VPC A と VPC B との間のピアリング接続 (pcx-aaaabbbb)、VPC A と VPC C との間のピアリング接続 (pcx-aaaacccc) があります。VPC A には、ピアリング接続ごとに 1 つのインスタンスを持つサブネットがあります。特定のインスタンスに対するピアトラフィックを制限する場合に、この設定を使用できます。

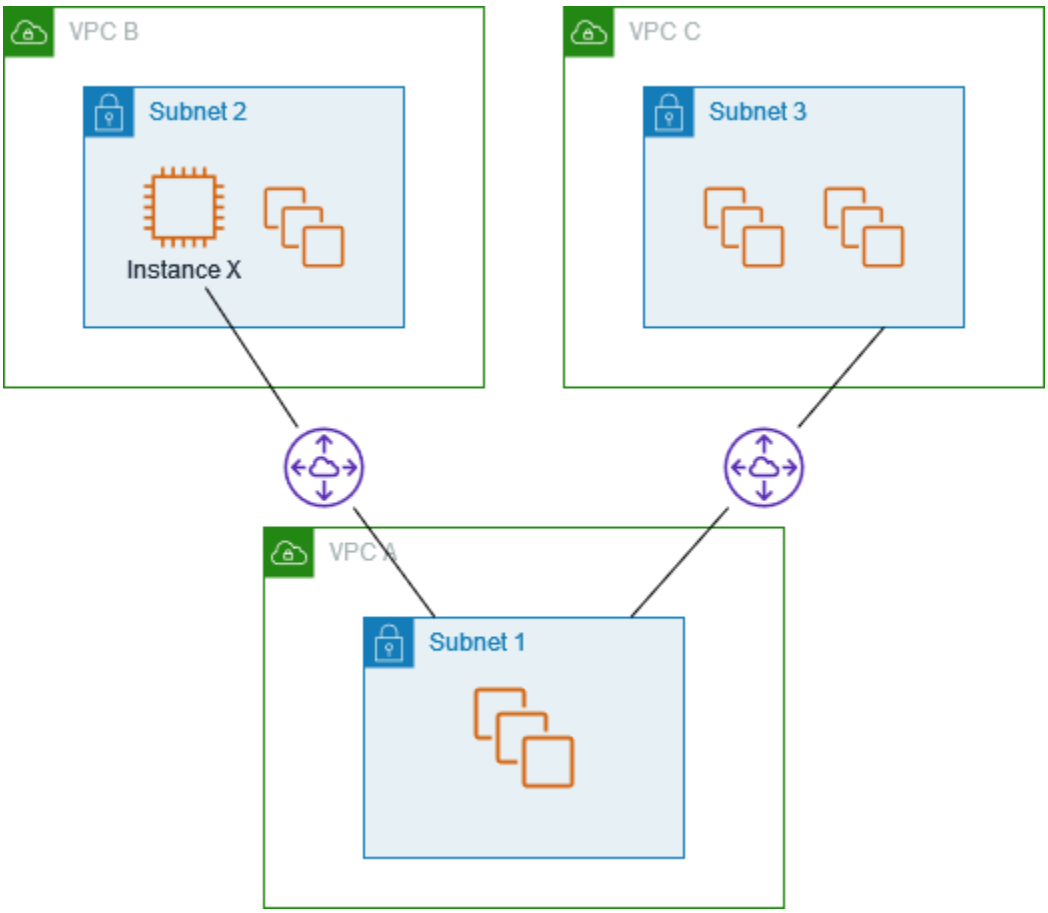


各 VPC のルートテーブルは、ピア VPC の単一の IP アドレス (つまり特定のインスタンス) にアクセスする該当する VPC ピアリング接続を指します。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A CIDR*	ローカル
	###### 3 # IP ####	pcx-aaaabbbb
	###### 4 # IP ####	pcx-aaaacccc
VPC B	*VPC B CIDR*	ローカル
	###### 1 # IP ####	pcx-aaaabbbb
VPC C	*VPC C CIDR*	ローカル
	###### 2 # IP ####	pcx-aaaacccc

最長のプレフィックスの一致を使用して 2 つの VPC にアクセスする 1 つの VPC

この設定では、1 つのサブネットを持つ中央 VPC (VPC A)、VPC A と VPC B との間のピアリング接続 (pcx-aaaabbbb)、VPC A と VPC C との間のピアリング接続 (pcx-aaaacccc) があります。VPC B および VPC C の CIDR ブロックが一致しています。VPC ピアリング接続 pcx-aaaabbbb を使用して VPC A と VPC B 内に存在する特定のインスタンスとの間におけるトラフィックをルーティングします。VPC B と VPC C によって共有される CIDR アドレス範囲に送信されるその他のトラフィックはすべて、pcx-aaaacccc を経由して VPC C にルーティングされます。



VPC のルートテーブルは、プレフィックス最長一致を使用して、目的の VPC ピアリング接続で最も具体的なルートを選択します。その他のすべてのトラフィックは、次に一致するルート、今回の場合は VPC ピアリング接続 pcx-aaaacccc を経由してルーティングします。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A CIDR ####*	ローカル
	###### X # IP ####	pcx-aaaabbbb
	VPC C CIDR ####	pcx-aaaacccc
VPC B	*VPC B CIDR ####*	ローカル
	VPC A CIDR ####	pcx-aaaabbbb
VPC C	*VPC C CIDR ####*	ローカル

ルートテーブル	送信先	ターゲット
	VPC A CIDR ####	pcx-aaaacccc

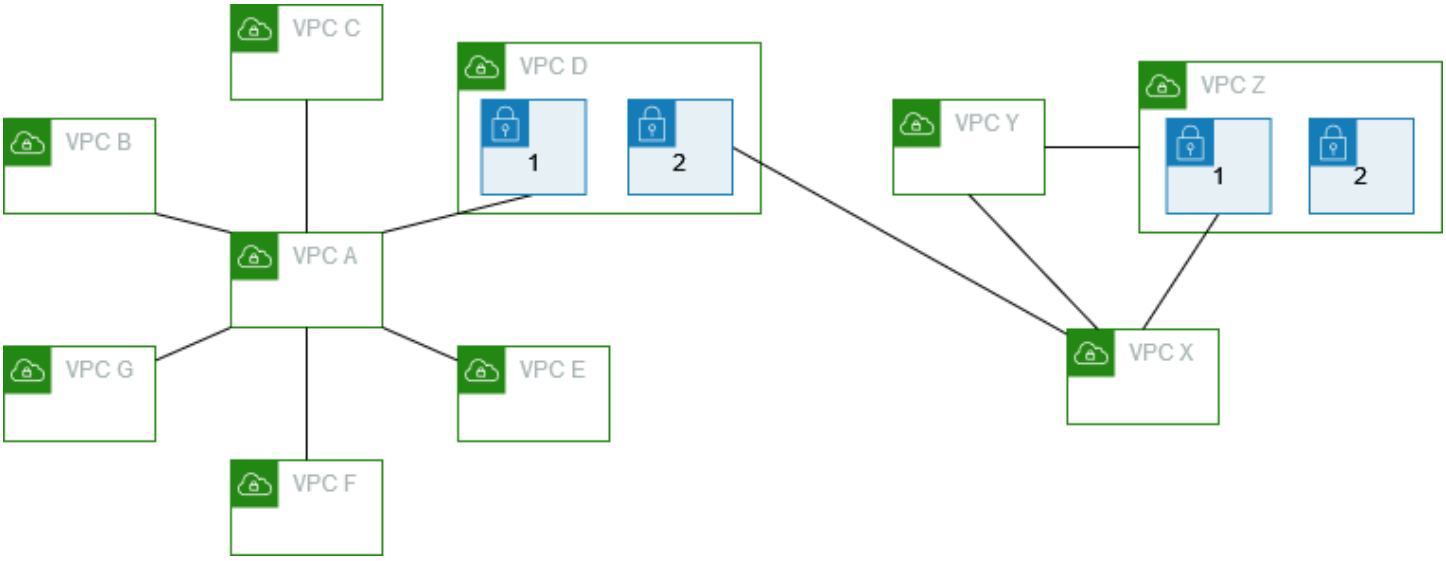
> **Important**
>
> VPC B 内のインスタンス X 以外のインスタンスが VPC A にトラフィックを送信すると、レスポンストラフィックが VPC B の代わりに VPC C にルーティングされる可能性があります。詳細については、「レスポンストラフィックのルーティング」を参照してください。

多重 VPC 設定

この設定では、中央 VPC (VPC A) がスポーク設定で複数の VPC とピアリング接続されています。また、3 つの VPC (VPC X、Y、Z) がフルメッシュ設定でピアリング接続されています。

VPC D には、VPC X との VPC ピアリング接続も設定されています (pcx-ddddxxxx)。VPC A と VPC X は CIDR ブロックが重複しています。これは、VPC A と VPC D との間のピアリングトラフィックが VPC D 内の特定のサブネット (サブネット 1) に制限されていることを意味します。これにより、VPC D が VPC A または VPC X からリクエストを受け取った場合、正しい VPC にレスポンストラフィックが返されます。AWS の VPC ピアリング接続では、パケットのソース IP を確認してリプライパケットをソースにルーティングするユニキャストリバースパス転送をサポートしていません。詳細については、「レスポンストラフィックのルーティング」を参照してください。

同様に、VPC D と VPC Z は CIDR ブロックが重複しています。VPC D と VPC X の間のピアトラフィックは VPC D のサブネット 2 に制限され、VPC X と VPC Z の間のピアトラフィックは VPC Z のサブネット 1 に制限されます。これは、VPC X が VPC D または VPC Z からピアトラフィックを受け取った場合に、正しい VPC にレスポンストラフィックを送り返すようにするためです。



VPC B、C、E、F、G のルートテーブルは、VPC A の CIDR ブロック全体にアクセスする該当するピアリング接続を指します。VPC A のルートテーブルは、VPC B、C、E、F、G それぞれの CIDR ブロック全体にアクセスする該当するピアリング接続を指します。ピアリング接続 pcx-aaaadddd の場合、VPC A のルートテーブルは VPC D のサブネット 1 に対するトラフィックのみをルーティングし、VPC D のサブネット 1 のルートテーブルは VPC A の CIDR ブロック全体を指します。

VPC Y のルートテーブルは VPC X と VPC Z の CIDR ブロック全体にアクセスする該当するピアリング接続を指し、VPC Z のルートテーブルは VPC Y の CIDR ブロック全体にアクセスする該当するピアリング接続を指します。VPC Z のサブネット 1 のルートテーブルは VPC Y の CIDR ブロック全体にアクセスする該当するピアリング接続を指します。VPC X のルートテーブルは VPC D のサブネット 2 と VPC Z のサブネット 1 にアクセスする該当するピアリング接続を指します。

ルートテーブル	送信先	ターゲット
VPC A	*VPC A CIDR*	ローカル
	VPC B CIDR	pcx-aaaabbbb
	VPC C CIDR	pcx-aaaacccc
	VPC D ###### 1 CIDR	pcx-aaaadddd
	VPC E CIDR	pcx-aaaaeeee
	VPC F CIDR	pcx-aaaaffff

ルートテーブル	送信先	ターゲット
	VPC G CIDR	pcx-aaaagggg
VPC B	*VPC B CIDR*	ローカル
	VPC A CIDR	pcx-aaaabbbb
VPC C	*VPC C CIDR*	ローカル
	VPC A CIDR	pcx-aaaacccc
VPC D のサブネット 1	*VPC D CIDR*	ローカル
	VPC A CIDR	pcx-aaaadddd
VPC D のサブネット 2	*VPC D CIDR*	ローカル
	VPC X CIDR	pcx-ddddxxxx
VPC E	*VPC E CIDR*	ローカル
	VPC A CIDR	pcx-aaaaeeee
VPC F	*VPC F CIDR*	ローカル
	VPC A CIDR	pcx-aaaaffff
VPC G	*VPC G CIDR*	ローカル
	VPC A CIDR	pcx-aaaagggg
VPC X	*VPC X CIDR*	ローカル
	VPC D ###### 2 CIDR	pcx-ddddxxxx
	VPC Y CIDR	pcx-xxxxyyyy
	VPC Z ###### 1 CIDR	pcx-xxxxzzzz
VPC Y	*VPC Y CIDR*	ローカル

ルートテーブル	送信先	ターゲット
	VPC X CIDR	pcx-xxxxyyyy
	VPC Z CIDR	pcx-yyyyzzzz
VPC Z	*VPC Z CIDR*	ローカル
	VPC Y CIDR	pcx-yyyyzzzz
	VPC X CIDR	pcx-xxxxzzzz

VPC ピアリング接続のネットワーキングのシナリオ

いくつかの理由で、お客様の VPC の間に、またはお客様の所有する VPC と別の AWS アカウントの VPC との間に VPC ピアリング接続を設定することが必要になる場合があります。以下のシナリオは、どの設定がネットワーク要件に最適か判断するのに役立ちます。

シナリオ

- 複数の VPC をピアリング接続してリソースにフルアクセスする
- 1 つの VPC にピアリング接続して一元管理されているリソースにアクセスする

複数の VPC をピアリング接続してリソースにフルアクセスする

このシナリオでは、ピアリング接続してすべての VPC 間でリソースを完全に共有できるようにしたい VPC が複数あります。次に例をいくつか示します。

- 会社で財務部門用の VPC と、会計部門用の別の VPC を利用しています。財務部門は会計部門にあるすべてのリソースにアクセスする必要があり、会計部門は財務部門にあるすべてのリソースにアクセスする必要があります。
- 会社に IT 部門が複数あり、それぞれが独自の VPC を所有しています。同じ AWS アカウントに配置されている VPC もあれば、異なる AWS アカウントに配置されている VPC もあります。すべての VPC をピアリング接続して、各 IT 部門が互いのリソースにフルアクセスできるようにする必要があります。

このシナリオ用に VPC ピア接続とルートテーブルを設定する方法の詳細については、以下のドキュメントを参照してください。

- 2 つの VPC が相互にピアリング接続
- 3 つの VPC が相互にピアリング接続
- 多数の VPC が相互にピアリング接続

Amazon VPC コンソールで VPC ピアリング接続を作成して作業する方法の詳細については、「VPC ピアリング接続」を参照してください。

1 つの VPC にピアリング接続して一元管理されているリソースにアクセスする

このシナリオでは、他の VPC と共有するリソースを持つ中央 VPC が存在しています。中央 VPC はピア VPC に対するフルアクセスまたは部分アクセスを必要とし、同様に、ピア VPC は中央 VPC に対するフルアクセスまたは部分アクセスを必要とします。次に例をいくつか示します。

- 会社の IT 部門がファイル共有用の VPC を保持しています。その中央 VPC に他の VPC をピアリング接続する必要がありますが、他の VPC が相互にトラフィックを送信する必要はありません。
- 会社の VPC を顧客と共有しています。顧客はそれぞれ会社の VPC との VPC ピアリング接続を作成できますが、ピアリング接続されている他の VPC にトラフィックをルーティングすることはできず、他の顧客のルートも認識しません。
- 中央 VPC を Active Directory サービスで使用しています。ピア VPC の特定のインスタンスが Active Directory サーバーにリクエストを送信し、中央 VPC に対するフルアクセスを必要としています。中央 VPC はピア VPC に対するフルアクセスを必要としません。特定のインスタンスにレスポンストラフィックをルーティングする必要があるだけです。

Amazon VPC コンソールで VPC ピアリング接続を作成して作業する方法の詳細については、「VPC ピアリング接続」を参照してください。

VPC ピアリングの Identity and Access Management

デフォルトでは、 ユーザーは VPC ピア接続を作成または変更することはできません。VPC ピアリングリソースへのアクセス許可を付与するには、IAM アイデンティティ (ロールなど) に IAM ポリシーをアタッチします。

例

- 例: VPC ピアリング接続の作成
- 例: VPC ピアリング接続の承認
- 例: VPC ピアリング接続の削除
- 例: 特定のアカウントでの操作
- 例: コンソールを使用した VPC ピアリング接続の管理

Amazon VPC アクションのリストと、各アクションでサポートされているリソースと条件キーについては、「サービス認可リファレンス」の「Amazon EC2 のアクション、リソース、および条件キー」を参照してください。

例: VPC ピアリング接続の作成

次のポリシーでは、Purpose=Peering というタグが付いている VPC のみを使用して VPC ピアリング接続リクエストを作成するアクセス許可をユーザーに付与しています。最初のステートメントでは、条件キー (ec2:ResourceTag) が VPC リソースに適用されます。CreateVpcPeeringConnection アクションの VPC リソースは、常にリクエスタ VPC であることに注意してください。

2 番目のステートメントでは、VPC ピアリング接続リソースを作成するためのアクセス許可をユーザーに与えます。このため、特定のリソース ID の代わりにワイルドカード * が使用されます。

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action": "ec2:CreateVpcPeeringConnection",
```

```
      "Resource": "arn:aws:ec2:us-east-1:123456789012:vpc/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Purpose": "Peering"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "ec2:CreateVpcPeeringConnection",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:vpc-peering-connection/*"
    }
  ]
}
```

次のポリシーでは、特定の AWS アカウントのユーザーに、特定のリージョン内の任意の VPC を使用して VPC ピアリング接続を作成するアクセス許可を付与しています。ただし、ピアリング接続を受け入れる VPC が指定されたアカウントの特定の VPC である場合に限ります。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVpcPeeringConnection",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:vpc/*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateVpcPeeringConnection",
            "Resource": "arn:aws:ec2:us-east-1:123456789012:vpc-peering-
connection/*",
            "Condition": {
                "ArnEquals": {
                    "ec2:AccepterVpc": "arn:aws:ec2:us-
east-1:111122223333:vpc/vpc-1234567890abcdef0"
                }
            }
        }
```

```
    ]
}
```

例: VPC ピアリング接続の承認

次のポリシーでは、特定の AWS アカウントから VPC ピアリング接続リクエストを受け入れるアクセス許可をユーザーに付与しています。これにより、不明なアカウントから VPC ピア接続リクエストを受け入れることを防ぐことができます。ステートメントでは、これを適用するために ec2:RequesterVpc 条件キーが使用されます。

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action": "ec2:AcceptVpcPeeringConnection",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:vpc-peering-connection/*",
      "Condition": {
        "ArnEquals": {
          "ec2:RequesterVpc": "arn:aws:ec2:us-east-1:111122223333:vpc/*"
        }
      }
    },
    {
      "Effect":"Allow",
      "Action": "ec2:AcceptVpcPeeringConnection",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:vpc/*"
    }
  ]
}
```

次のポリシーでは、VPC に Purpose=Peering というタグが付いている場合に VPC ピアリクエストを受け入れるアクセス許可をユーザーに与えます。

JSON

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action": "ec2:AcceptVpcPeeringConnection",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:vpc-peering-connection/*"
    },
    {
      "Effect": "Allow",
      "Action": "ec2:AcceptVpcPeeringConnection",
      "Resource": "arn:aws:ec2:us-east-1:123456789012:vpc/*",
      "Condition": {
        "StringEquals": {
          "ec2:ResourceTag/Purpose": "Peering"
        }
      }
    }
  ]
}
```

例: VPC ピアリング接続の削除

次のポリシーでは、特定のアカウントのユーザーに、同じアカウント内の指定された VPC を使用する VPC ピアリング接続を除くすべての VPC ピアリング接続を削除するアクセス許可を付与しています。このポリシーでは、`ec2:AccepterVpc` 条件キーと `ec2:RequesterVpc` 条件キーの両方を指定しています。これは、VPC がリクエスタ VPC であるか、元の VPC ピアリング接続リクエスト内のピア VPC である可能性があるためです。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DeleteVpcPeeringConnection",
```

```
            "Resource": "arn:aws:ec2:us-east-1:123456789012:vpc-peering-
connection/*",
            "Condition": {
                "ArnNotEquals": {
                    "ec2:AccepterVpc": "arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-1234567890abcdef0",
                    "ec2:RequesterVpc": "arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-0abcdef1234567890"
                }
            }
        }
    ]
}
```

例: 特定のアカウントでの操作

次のポリシーでは、特定のアカウント内で VPC ピアリング接続を操作するアクセス許可をユーザーに付与しています。ユーザーは、VPC ピアリング接続の表示、作成、受け入れ、拒否、削除を実行できます (それらの接続がすべて AWS アカウント内の接続である場合)。

最初のステートメントでは、すべての VPC ピアリング接続を表示するアクセス許可をユーザーに付与しています。この場合、Resource エレメントではワイルドカード * が必要になります。現時点では、この API アクション (DescribeVpcPeeringConnections) が、リソースレベルのアクセス権限をサポートしていないためです。

2 番目のステートメントでは、VPC ピアリング接続を作成し、必要であれば、特定のアカウント内のすべての VPC へアクセスする許可をユーザーに付与しています。

3 番目のステートメントでは、すべての VPC ピアリング接続アクションの許可を付与するために、Action エレメントの一部としてワイルドカード * が使用されています。条件キーによって、アカウントの一部である VPC を使用する VPC ピアリング接続に対してのみ、アクションを実行することができます。例えば、アクセプタ VPC またはリクエスタ VPC のどちらかが別のアカウントに属する場合、ユーザーは VPC ピアリング接続を削除できません。ユーザーは、別のアカウントに属する VPC を使用して VPC ピアリング接続を作成することはできません。

JSON

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeVpcPeeringConnections",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcPeeringConnection",
                "ec2:AcceptVpcPeeringConnection"
            ],
            "Resource": "arn:aws:ec2:*:111122223333:vpc/*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:*VpcPeeringConnection",
            "Resource": "arn:aws:ec2:*:111122223333:vpc-peering-connection/*",
            "Condition": {
                "ArnEquals": {
                    "ec2:AccepterVpc": "arn:aws:ec2:*:111122223333:vpc/*",
                    "ec2:RequesterVpc": "arn:aws:ec2:*:111122223333:vpc/*"
                }
            }
        }
    ]
}
```

例: コンソールを使用した VPC ピアリング接続の管理

Amazon VPC コンソールへの VPC ピアリング接続を表示するには、ユーザーに `ec2:DescribeVpcPeeringConnections` アクションを使用するアクセス権限が必要です。[Create Peering Connection (ピアリング接続の作成)] ページを使用するには、ユーザーは `ec2:DescribeVpcs` アクションを使用する許可が必要です。これにより、VPC を表示および選択するためのアクセス許可が付与されます。`ec2:DescribeVpcPeeringConnections` を除くすべての `ec2:*PeeringConnection` アクションに、リソースレベルのアクセス権限を適用できます。

次のポリシーでは、VPC ピアリング接続を表示し、[Create VPC Peering Connection] (VPC ピアリング接続の作成) ダイアログボックスで、特定のリクエスタの VPC のみを使用して VPC ピアリング接続を作成するアクセス許可がユーザーに付与されます。ユーザーが別のリクエスト元の VPC を使用して VPC ピア接続を作成しようとすると、リクエストは失敗します。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect":"Allow",
      "Action": [
        "ec2:DescribeVpcPeeringConnections", "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Effect":"Allow",
      "Action": "ec2:CreateVpcPeeringConnection",
      "Resource": [
        "arn:aws:ec2:*:*:vpc/vpc-1234567890abcdef0",
        "arn:aws:ec2:*:*:vpc-peering-connection/*"
      ]
    }
  ]
}
```

1 つのアカウントでの VPC ピアリング接続のクォータ

VPC ピアリングを使用すると、2 つの VPC を接続できます。この結果、ある VPC のリソースは、同じネットワーク内であるかのように、他の VPC のリソースと通信できます。VPC ピアリングは、同じ AWS リージョンであっても、別のリージョンであっても、VPC を接続するのに便利な機能です。このセクションでは、VPC ピアリング接続を使用する際に注意すべきクォータについて説明します。

以下のテーブルは、AWS アカウントの VPC ピアリング接続のクォータ (以前は制限と呼ばれていたもの) の一覧を示しています。特に明記されていない限り、これらのクォータの引き上げをリクエストできます。

現在の VPC ピアリング接続要件がデフォルトのクォータを超えている場合は、サービス制限の引き上げリクエストを送信することをお勧めします。当社でユースケースを確認し、連携してクォータを適宜調整して、VPC 環境がビジネスニーズの拡大に対応できるようにします。

名前	デフォルト	引き上げ可能
VPC 当たりのアクティブな VPC ピアリング接続	50	可能 (最大 125)
未処理の VPC ピアリング接続リクエスト	25	可能
許容されない VPC ピアリング接続リクエストの有効期限	1 週間 (168 時間)	不可

VPC ピアリング接続の使用に関するルールの詳細については、「VPC ピアリングの制限事項」を参照してください。Amazon VPC のクォータの詳細については、「Amazon VPC ユーザーガイド」の「Amazon VPC クォータ」を参照してください。

「Amazon VPC ピアリングガイド」のドキュメント履歴

以下の表は、「Amazon VPC ピアリングガイド」のドキュメントリリースを説明するものです。

変更	説明	日付
作成時のタグ付け	VPC ピア接続とルートテーブルを作成するときに、タグを追加できます。	2020 年 7 月 20 日
リージョン間ピアリング	DNS ホスト名解決は、アジアパシフィック (香港) リージョンのリージョン間 VPC ピア接続でサポートされています。	2019 年 8 月 26 日
リージョン間ピアリング	複数の異なる AWS リージョンの VPC 間で VPC ピアリング接続を作成できます。	2017 年 11 月 29 日
VPC ピアリング接続の DNS 解決サポート	ローカルの VPC を有効にして、ピア VPC のインスタンスからクエリが実行されたときに、パブリック DNS ホスト名がプライベート IP アドレスに解決されるように設定できます。	2016 年 7 月 28 日
古くなったセキュリティグループルール	セキュリティグループがピア VPC のセキュリティグループルールで参照されているかどうかを確認し、古くなったセキュリティグループルールを特定できます。	2016 年 12 月 5 日
VPC ピアリング接続での ClassicLink の使用	ローカルのリンクされた EC2-Classic インスタンスとピア VPC のインスタンスが相互に	2016 年 4 月 26 日

	通信できるように、ピアリング接続を変更できます。	
VPC ピアリング接続	2 つの VPC 間で VPC ピアリング接続を作成して、いずれかの VPC のインスタンスが、プライベート IP アドレスを使用して相互に通信できます	2014 年 3 月 24 日