Amazon Virtual Private Cloud

ユーザーガイド

Amazon Virtual Private Cloud: ユーザーガイド

Table of Contents

Amazon VPC とは?

Amazon Virtual Private Cloud (Amazon VPC) を使用すると、定義した仮想ネットワーク内で AWS リソースを起動できます。仮想ネットワークは、お客様自身のデータセンターで運用されていた従来のネットワークによく似ていますが、 のスケーラブルなインフラストラクチャを使用できるというメリットがありますAWS

Amazon VPC の概念

Amazon VPC は、Amazon EC2 のネットワークレイヤーです。Amazon EC2 を初めて使用する場合は、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Amazon EC2 とは」を参照してください。

VPC の主な概念は次のとおりです。

- Virtual Private Cloud (VPC) — AWS アカウント専用の仮想ネットワーク。
- サブネット — VPC の IP アドレスの範囲。
- ルートテーブル — ネットワークトラフィックの経路を判断する際に使用される、ルートと呼ばれる一連のルール。
- インターネットゲートウェイ — VPC 内のリソースとインターネット間の通信を可能にするために VPC にアタッチするゲートウェイ。
- VPC エンドポイント - PrivateLink を使用してサポートされている AWS サービスや VPC エンドポイントサービスに VPC をプライベートに接続できます。インターネットゲートウェイ、NAT デバイス、VPN 接続、および AWS Direct Connect 接続は必要ありません。VPC のインスタンスは、サービスのリソースと通信するためにパブリック IP アドレスを必要としません。VPC と他のサービス間のトラフィックは、Amazon ネットワークを離れません。詳細については、「」を参照してください AWS PrivateLink および VPC エンドポイント (p. 359)
- CIDR ブロック —クラスレスドメイン間ルーティング。インターネットプロトコルアドレスの割り当てとルート集計方法。詳細については、Wikipedia の「Classless Inter-Domain Routing」をご参照ください。

Amazon VPC にアクセスする

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

- AWS Management Console — VPC へのアクセスに使用するウェブインターフェイスを提供します。
- AWS Command Line Interface (AWS CLI) — Amazon VPC を含むさまざまな AWS サービス用のコマンドを備えており、Windows、Mac、Linux でサポートされています。AWS Command Line Interface: 詳細については、「」を参照してください。
- AWS SDK — 言語固有の API を提供し、署名の計算、リクエストの再試行処理、エラー処理など、接続のさまざまな詳細を処理します。詳細については、AWS SDK をご参照ください。
- クエリ API — HTTPS リクエストを使用して呼び出す低レベル API アクションを提供します。クエリ API の使用は、Amazon VPC の最も直接的なアクセス方法ですが、リクエストに署名するハッシュの生成やエラー処理など、低レベルの詳細な作業をアプリケーションで処理する必要があります。詳細については、Amazon EC2 API リファレンスを参照してください。

Amazon VPC の料金

VPC は追加料金なしで使用できます。NAT ゲートウェイ、Reachability Analyzer、トラフィックミラーリングなど、一部の VPC コンポーネントには料金が発生します。詳細については、「Amazon VPC の料金」を参照してください。

Amazon VPC クォータ

プロビジョニングできる Amazon VPC コンポーネントの数にはクォータがあります。これらのクォータのいくつかは、リクエストによって引き上げることができます。詳細については、「Amazon VPC クォータ (p. 362)」を参照してください。

Amazon VPC の仕組み

Amazon Virtual Private Cloud (Amazon VPC) を使用すると、定義した仮想ネットワーク内で AWS リソースを起動できます。仮想ネットワークは、お客様自身のデータセンターで運用されていた従来のネットワークによく似ていますが、 のスケーラブルなインフラストラクチャを使用できるというメリットがありますAWS

Amazon VPC は、Amazon EC2 のネットワークレイヤーです。Amazon EC2 を初めて使用する場合は、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Amazon EC2 とは」を参照してください。

目次

VPC とサブネット

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 の IP アドレス範囲を指定して、サブネットを追加し、セキュリティグループを関連付けて、ルートテーブルを設定できます。

サブネットは、VPC の IP アドレスの範囲です。AWS リソースは、指定したサブネット内に起動できます。インターネットに接続する必要があるリソースにはパブリックサブネットを、インターネットに接続しないリソースにはプライベートサブネットを使用してください。

各サブネットの AWS リソースを保護するには、セキュリティグループやネットワークアクセスコントロールリスト (ACL) など、複数のセキュリティレイヤーを使用できます。

オプションで、VPC に IPv6 CIDR ブロックを関連付け、VPC のインスタンスに IPv6 アドレスを割り当てることができます。

詳細はこちら

- VPC とサブネットの基本 (p. 106)
- Amazon VPC でのインターネットワークトラフィックのプライバシー (p. 165)
- VPC の IP アドレス指定 (p. 124)

値とデフォルト以外の VPC

アカウントが 2013 年 12 月 4 日以降に作成された場合、各アベイラビリティーゾーンにデフォルトサブネットを持つデフォルト VPC が付属します。デフォルト VPC は EC2-VPC による高度な機能のメリット

があり、即時に利用することができます。デフォルト VPC をお持ちのお客様が、インスタンス起動時にサブネットを指定しなかった場合は、そのインスタンスはお客様のデフォルト VPC で起動されます。インスタンスをデフォルト VPC で起動するときに、Amazon VPC に関する知識は必要ありません。

独自の VPC を作成し、必要に応じて設定することもできます。これはデフォルト以外の VPC と呼ばれます。デフォルト以外の VPC で作成するサブネット、そしてデフォルト VPC で作成する追加サブネットは、デフォルト以外のサブネットと呼ばれます。

詳細はこちら

- デフォルト VPC とデフォルトサブネット (p. 156)
- Amazon VPC の使用を開始する (p. 11)

ルートテーブル

ルートテーブルは、VPC からのネットワークトラフィックの経路を決めるために使用される一連のルール (ルートと呼ばれます) で構成されます。サブネットを特定のルートテーブルに明示的に関連付けることができます。それ以外の場合、サブネットはメインルートテーブルに暗黙的に関連付けられます。

ルートテーブル内の各ルートは、トラフィックを移動させる IP アドレスの範囲 (宛先) と、トラフィックを送信するゲートウェイ、ネットワークインターフェイス、または接続 (ターゲット) を指定します。

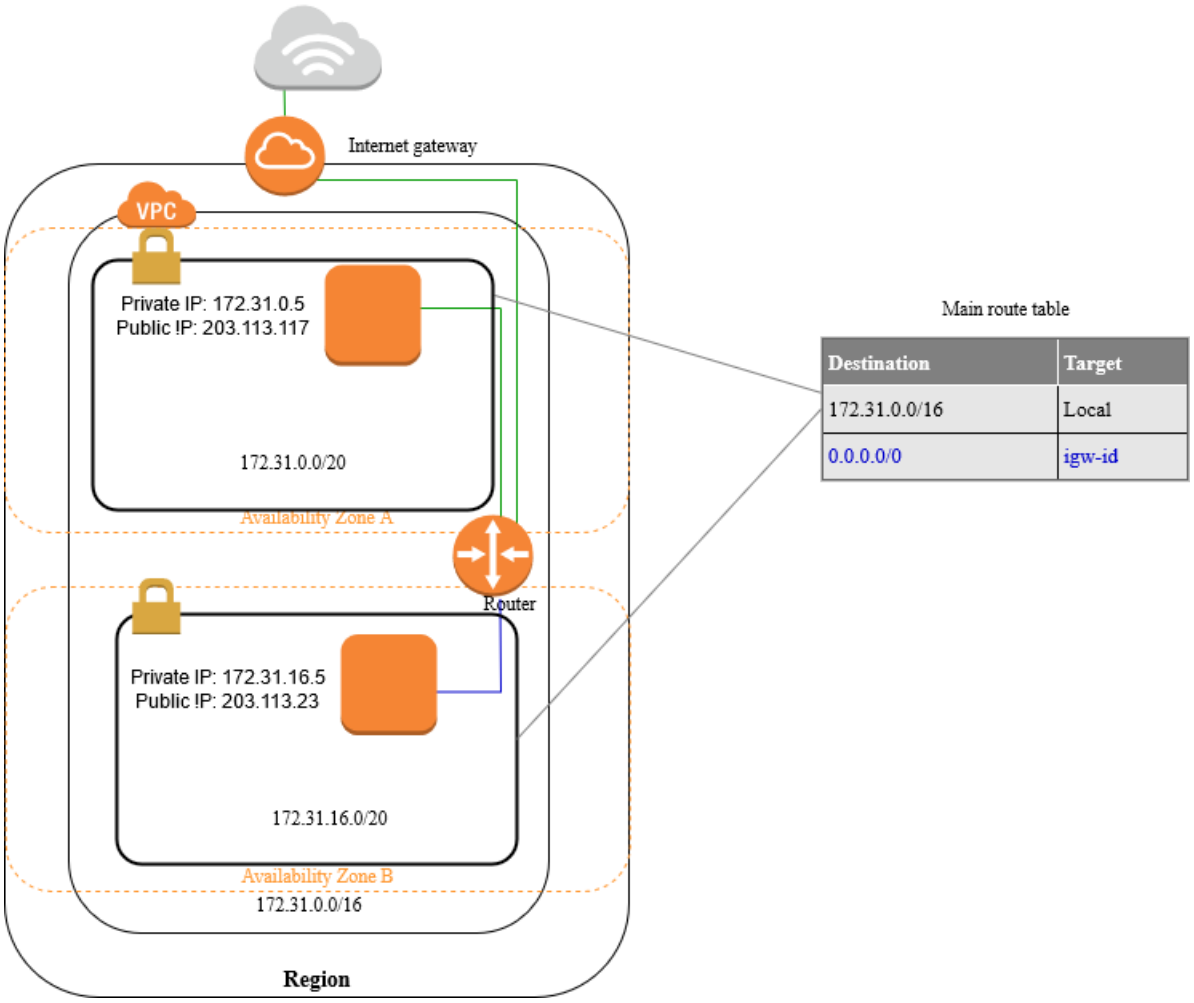
詳細はこちら

- VPC のルートテーブル (p. 294)

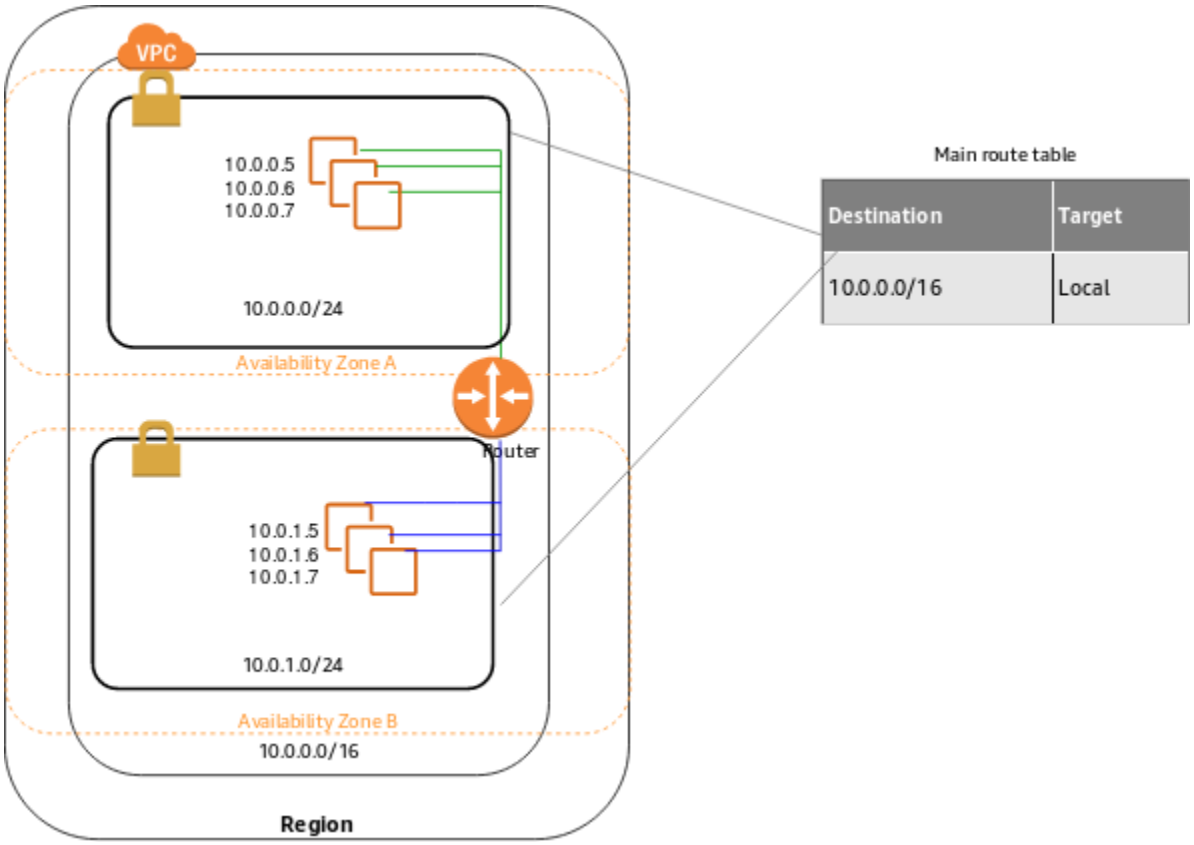
インターネットへのアクセス

VPC 内に起動するインスタンスが VPC 外のリソースにどのようにアクセスするかをコントロールします。

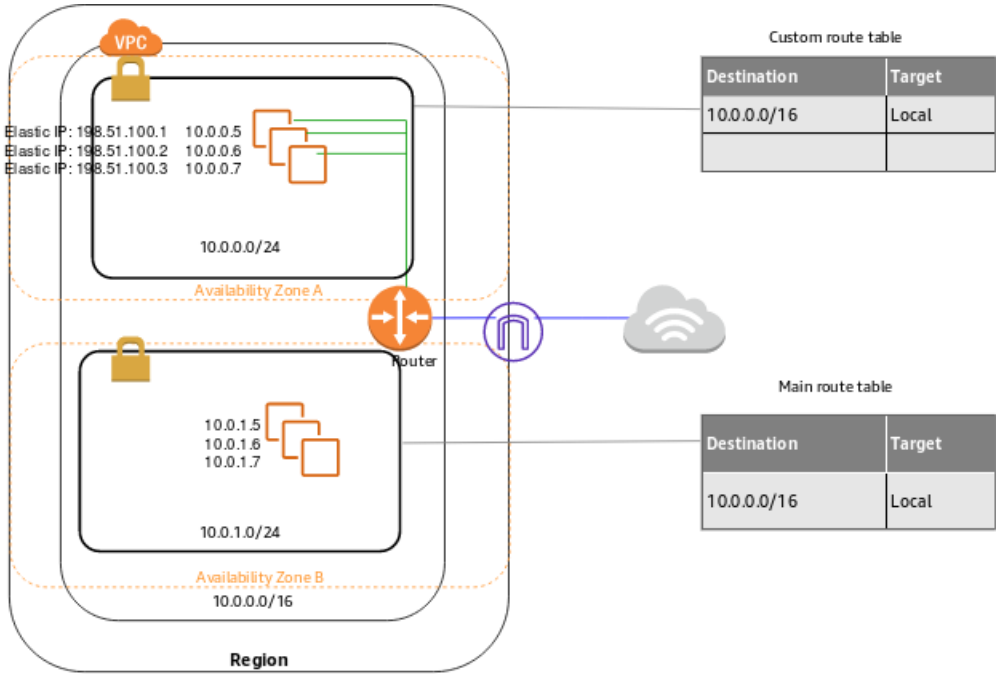
デフォルト VPC にはインターネットゲートウェイが含まれ、各デフォルトサブネットはパブリックサブネットです。デフォルトサブネット内に起動するインスタンスにはそれぞれ、プライベート IPv4 アドレスとパブリック IPv4 アドレスが割り当てられています。これらのインスタンスは、このインターネットゲートウェイを介してインターネットと通信できます。インターネットゲートウェイを使用することで、インスタンスは Amazon EC2 ネットワークエッジを介してインターネットに接続できます。



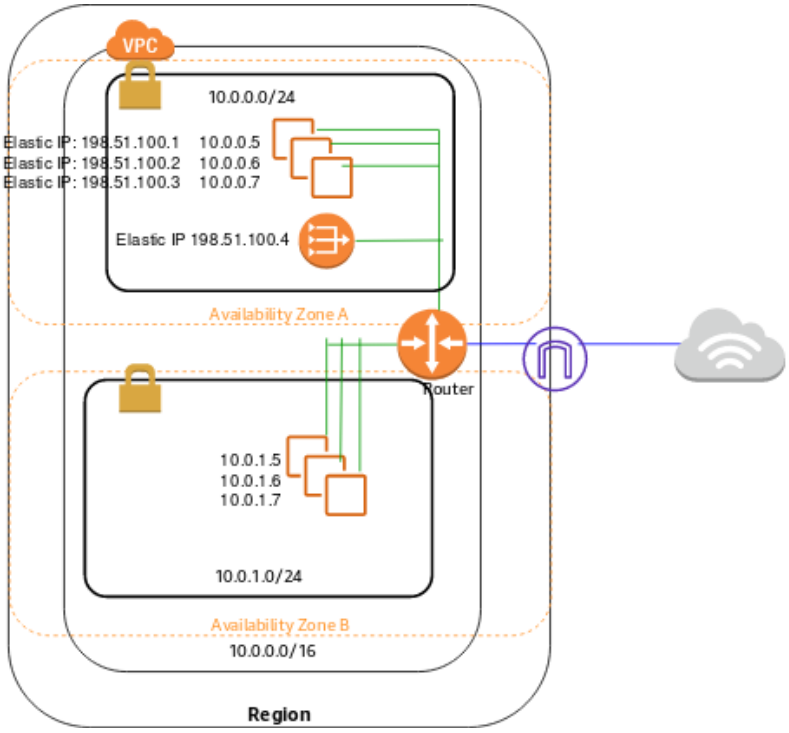
デフォルトでは、デフォルト以外のサブネットで起動した各インスタンスにはプライベート IPv4 アドレスが割り当てられていますが、パブリック IPv4 アドレスは割り当てられていません。ただし、起動時に明示的にパブリック IP アドレスを割り当てた場合や、サブネットのパブリック IP アドレス属性を変更した場合は例外です。これらのインスタンスは相互に通信できますが、インターネットにアクセスできません。



デフォルト以外のサブネットで起動するインスタンスのインターネットアクセスを有効にするには、インターネットゲートウェイをその VPC (デフォルト VPC でない場合) にアタッチし、インスタンスに Elastic IP アドレスを関連付けます。



または、VPC のインスタンスによるインターネットへのアウトバウンド接続の開始を許可し、インターネットからの未承諾のインバウンド接続を拒否するには、ネットワークアドレス変換 (NAT) デバイスを使用できます。NAT では、複数のプライベート IPv4 アドレスが 1 つのパブリック IPv4 アドレスにマッピングされます。NAT デバイスを elastic IP アドレスで構成し、インターネットゲートウェイを介してインターネットに接続できます。これにより、NAT デバイスを介してプライベートサブネットのインスタンスをインターネットに接続できるようになり、トラフィックがインスタンスからインターネットゲートウェイにルーティングされ、すべての応答がインスタンスにルーティングされます。



IPv6 CIDR ブロックを VPC に関連付けて IPv6 アドレスをインスタンスに割り当てると、インスタンスはインターネットゲートウェイを介して IPv6 経由でインターネットに接続できます。また、インスタンスは、Egress-only インターネットゲートウェイを使用して IPv6 経由でインターネットへのアウトバウンド接続を開始できます。IPv6 トラフィックは IPv4 トラフィックと異なるため、IPv6 トラフィックの別のルートをルートテーブルに含める必要があります。

詳細はこちら

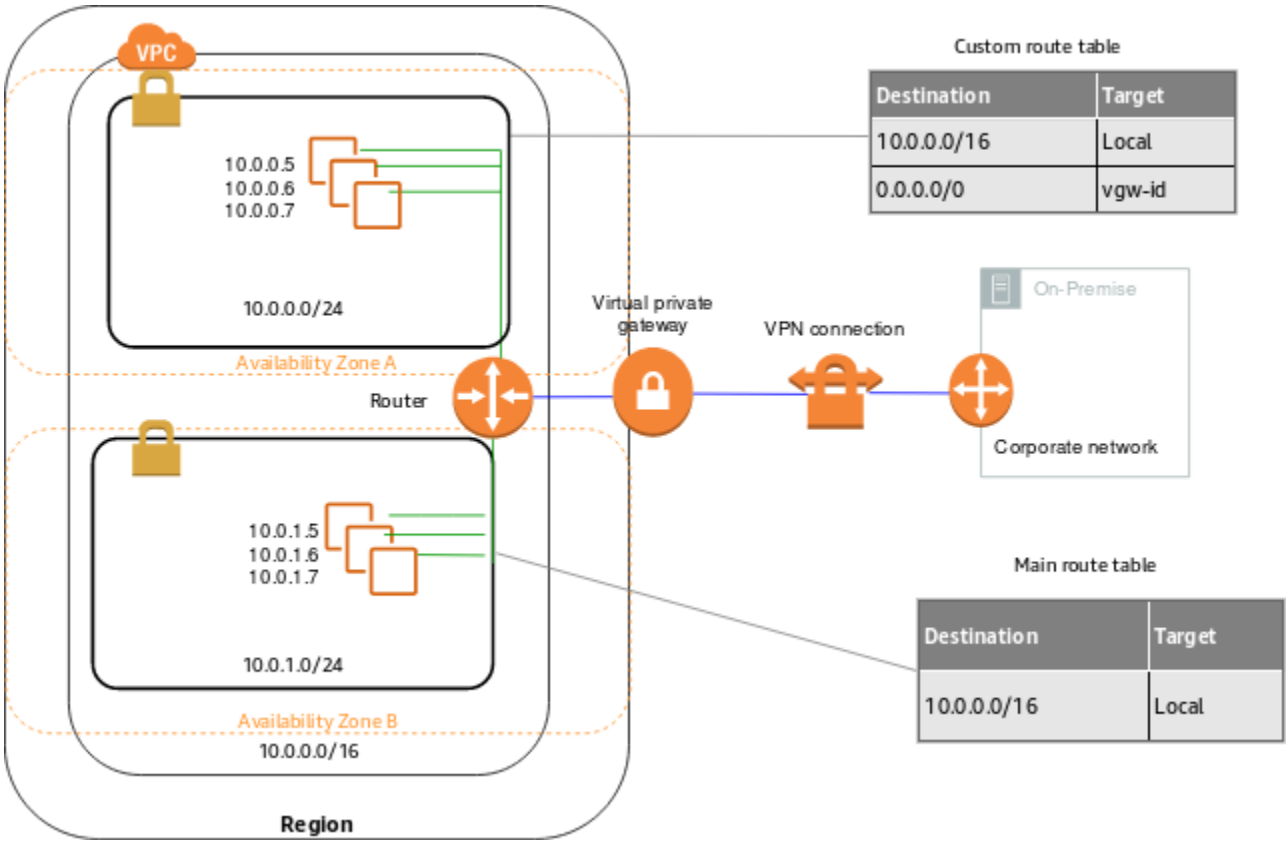
- インターネットゲートウェイ (p. 221)
- Egress-Only インターネットゲートウェイ (p. 228)
- VPC の NAT デバイス (p. 236)

企業ネットワークまたはホームネットワークにアクセスする

オプションで、IPsec AWS Site-to-Site VPN 接続を使用して VPC を自社データセンターに接続すると、AWS クラウドをデータセンターの延長として利用できます。

Site-to-Site VPN 接続は、AWS 側の仮想プライベートゲートウェイまたはトランジットゲートウェイと、データセンターにあるカスタマーゲートウェイデバイスとの間の 2 つの VPN トンネルで構成されます。

カスタマーゲートウェイデバイスは、Site-to-Site VPN 接続のお客様側で設定する物理デバイスまたはソフトウェアアプライアンスです。



詳細はこちら

- AWS Site-to-Site VPN ユーザーガイド
- [Transit Gateways (トランジットゲートウェイ)]

VPC とネットワークの接続

2 つの VPC 間に VPC ピアリング接続を作成して、それらの間のトラフィックをプライベートにルーティングできます。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。

また、トランジットゲートウェイ を作成し、それを使用して VPC とオンプレミスのネットワークを相互接続することもできます。トランジットゲートウェイは、アタッチメント間で流れるトラフィックのリージョン仮想ルーターとして機能します。これには、VPC、VPN 接続、AWS Direct Connect ゲートウェイ、およびトランジットゲートウェイピア接続が含まれます。

詳細はこちら

- VPC ピア機能ガイド
- [Transit Gateways (トランジットゲートウェイ)]

AWS プライベートグローバルネットワークの考慮事項

AWS は、お客様のネットワークニーズに対応するために、セキュアなクラウドコンピューティング環境を提供する、高パフォーマンスで低レイテンシーのプライベートグローバルネットワークを提供します。AWS リージョンは複数のインターネットサービスプロバイダー (ISP) や、プライベートグローバルネットワークバックボーンに接続され、それによりお客様が送信したクロスリージョントラフィックに対して高いネットワークパフォーマンスが提供されます。

以下の考慮事項に注意してください。

- すべてのリージョンのアベイラビリティーゾーン内またはアベイラビリティーゾーン間のトラフィックは、AWS プライベートグローバルネットワーク経由でルーティングされます。
- リージョン間のトラフィックは、中国リージョンを除き、常に AWS プライベートグローバルネットワーク経由でルーティングされます。

ネットワークパケットの損失は、ネットワークフローの衝突、下位レベル (レイヤー2) のエラー、その他のネットワーク障害など、さまざまな要因によって引き起こされる可能性があります。パケット損失を最小限に抑えるために、当社はネットワークを設計および運用しています。AWS リージョンを接続するグローバルバックボーン全体のパケットロス率 (PLR) を測定しています。当社のバックボーンネットワークは、1時間あたりの PLR の p99 が 0.0001% 未満になるように運用されています。

サポートされているプラットフォーム

オリジナルリリースの Amazon EC2 は、ほかのカスタマーと共用していた EC2-Classic プラットフォームという名の単一のフラットネットワークをサポートしていました。以前の AWS アカウントでは、このプラットフォームをまだサポートしており、EC2-Classic または VPC にインスタンスを起動できます。2013 年 12 月 4 日以降に作成されたアカウントは EC2-VPC のみをサポートします。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「EC2-Classic」を参照してください。

Amazon VPC ドキュメント

次の表は、Amazon VPC で作業するときに役立つその他のドキュメントの一覧です。

ガイド	説明
Amazon Virtual Private Cloud Connectivity Options	ネットワーク接続のオプションの概要が記載されています。
VPC ピアリング接続	VPC ピア接続のシナリオと、サポートされるピア設定について説明します。
[Transit Gateways (トランジットゲートウェイ)]	トランジットゲートウェイと、ネットワーク管理者がトランジットゲートウェイを設定する方法を説明します。
Transit Gateway Network Manager	トランジットゲートウェイ Network Manager について説明し、グローバルネットワークを設定およびモニタリングできるようにします。
トラフィックのミラーリング	トラフィックミラーリングのターゲット、フィルタ、およびセッションについて説明し、管理者がこれらを設定できるようにします。

ガイド	説明
AWS Direct Connect	AWS Direct Connect を使用して、リモートのネットワークから VPC への専用のプライベート接続を作成する方法について説明します。
AWS Client VPN	クライアント VPN エンドポイントを作成および設定して、リモートユーザーが VPC のリソースにアクセスできるようにする方法について説明します。
VPC Reachability Analyzer	VPC 内のリソース間のネットワーク到達可能性を分析してデバッグする方法を説明します。

Amazon VPC の使用を開始する

Amazon VPC の使用を開始するには、デフォルト以外の VPC を作成します。以下のステップでは、Amazon VPC ウィザードを使用して、パブリックサブネットを持つデフォルト以外の VPC を作成する方法について説明します。パブリックサブネットは、インターネットゲートウェイを介してインターネットにアクセスできるサブネットです。その後、サブネット内にインスタンスを起動して接続できます。

または、既存のデフォルト VPC でインスタンスを起動するには、「EC2 インスタンスをデフォルト VPC 内に起動する」を参照してください。

Amazon VPC を初めて使用するには、アマゾン ウェブ サービス (AWS) にサインアップする必要があります。サインアップすると、Amazon VPC を含む AWS のすべてのサービスに対して AWS アカウントが自動的にサインアップされます。AWS アカウントをまだ作成していない場合は、https://aws.amazon.com/ にアクセスし、[まずは無料で始める] を選択します。

VPC にローカルゾーンを使用する場合は、VPC を作成してから、ローカルゾーンにサブネットを作成します。詳細については、「the section called “VPC を作成する” (p. 116)」および「the section called “VPC にサブネットを作成する” (p. 117)」を参照してください。

目次

概要

この演習を完了するには、以下の作業を行います。

- 1 つのパブリックサブネットを持つデフォルト以外の VPC を作成する。
- サブネット内に Amazon EC2 インスタンスを起動します。
- Elastic IP アドレスをインスタンスに関連付ける.これでインスタンスがインターネットにアクセスできるようになります。

Amazon VPC を操作するためのアクセス許可を IAM ユーザーに付与する方法については、「Amazon VPC の Identity and Access Management (p. 169)」および「Amazon VPC ポリシーの例 (p. 177)」を参照してください。

ステップ 1: VPC を作成する

このステップでは、Amazon VPC コンソールで Amazon VPC ウィザードを使用して VPC を作成します。ウィザードは次の手順を自動的に行います。

- /16 の IPv4 CIDR ブロック (65,536 個のプライベート IP アドレスのネットワーク) を持つ VPC を作成します。

- インターネットゲートウェイをこの VPC にアタッチします。
- サイズ /24 の IPv4 サブネット (256 個のプライベート IP アドレスを含む範囲) を VPC に作成します。
- カスタムルートテーブルを作成し、サブネットに関連付けると、サブネットとインターネットゲートウェイ間でトラフィックが転送されます。

Amazon VPC ウィザードを使用して VPC を作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションバーの右上から、VPC を作成する AWSリージョン をメモしておきます。別のリージョンから VPC 内にインスタンスを起動できないので、この演習が終了するまでは同じリージョンで作業を続けるようにしてください。
3. ナビゲーションペインで、[VPC ダッシュボード] を選択します。ダッシュボードから、[VPC ウィザードの起動] を選択します。

   Note

   ナビゲーションペインで [お客様の VPC] を選択しないでください。そのページの [VPC の作成] ボタンを使用して VPC ウィザードにアクセスすることはできません。
4. [1 個のパブリックサブネットを持つ VPC] を選択し、[選択] を選択します。
5. 設定ページで [VPC name] フィールドに VPC の名前を入力します (たとえば、`my-vpc`) 。続いて、[Subnet name] フィールドにサブネットの名前を入力します。これは、VPC およびサブネットを作成後に Amazon VPC コンソールにおいてそれらを識別するのに役立ちます。この演習では、その他の設定をページに残したまま、[VPC を作成] を選択します。
6. ステータスウィンドウに、作業の進行状況が表示されます。作業が完了したら、[OK] を選択してステータスウィンドウを閉じます。
7. [Your VPCs] ページは、今作成したデフォルト VPC とその他の VPC が表示されます。デフォルト以外の VPC を作成した場合には、[Default VPC] 列に [No] と表示されます。

VPC に関する情報を表示する

VPC を作成したら、そのサブネット、インターネットゲートウェイ、およびルートテーブルに関する情報を表示できます。作成した VPC には 2 個のルートテーブルがあります ー デフォルトですべての VPC にあるメインルートテーブル、およびウィザードで作成したカスタムルートテーブルです。このカスタムルートテーブルにはサブネットが関連付けられます。したがって、サブネットへのトラフィックの流れ方は、このテーブル内のルートから決定されます。VPC に新規のサブネットを追加する場合、そのサブネットはデフォルトでメインルートテーブルを使用します。

VPC に関する情報を表示するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。作成した VPC の名前と ID を書き留めておきます ([Name] および [VPC ID] 列を参照します) 。この情報で VPC に関連付けられるコンポーネントを識別することができます。
3. ナビゲーションペインで、[Subnets] を選択します。コンソールには、VPC を作成した時に作成されたサブネットが表示されます。[Name] 列からその名前でサブネットを識別したり、前記の手順で VPC 情報を入手して、[VPC] 列から確認できます。
4. ナビゲーションペインで、[Internet Gateways] を選択します。VPC にアタッチしているインターネットゲートウェイは、[VPC] 列から確認でき、ここから VPC の ID と名前 (該当する場合) が表示されます。
5. ナビゲーションペインで、[Route Tables] を選択します。VPC に関連付けられる 2 つのルートテーブルがあります。カスタムルートテーブル ([Main] 列に [No] と表示されているもの) を選択し、[Routes] タブを選択すると、ルート情報が詳細ペインに表示されます。

- テーブルの 1 行目がローカルルートで、これによってインスタンスは VPC 内で通信できるようになります。このルートは、どのルートテーブルにもデフォルトで存在するものであり、削除することはできません。
- 2 行目は Amazon VPC ウィザードが追加したルートです。これにより、インターネット (0.0.0.0/0) に向けられたトラフィックが、サブネットからインターネットゲートウェイに流れるようになります。

6. メインルートテーブルを選択します。メインルートテーブルはローカルルートだけで、それ以外のルートがありません。

ステップ 2: VPC でインスタンスを起動する

EC2 インスタンスを VPC 内で起動するときは、どのサブネットでインスタンスを起動するかを指定する必要があります。この場合、作成した VPC のパブリックサブネットにインスタンスを起動します。Amazon EC2 コンソールで Amazon EC2 起動ウィザードを使用して、インスタンスを起動します。

EC2 インスタンスを VPC 内で起動するには

1. https://console.aws.amazon.com/ec2/ で Amazon EC2 コンソールを開きます。
2. ナビゲーションバーの右上で、VPC を作成したリージョンが選択されていることを確認します。
3. ダッシュボードから、[Launch Instance] を選択します。
4. ウィザードの最初のページで、使用する AMI を選択します。この演習では、Amazon Linux AMI または Windows AMI を選択します。
5. [Choose an Instance Type] ページで、起動するインスタンスのハードウェア設定とサイズを選択できます。デフォルトでは、お客様が選択した AMI に基づいて使用可能な最初のインスタンスタイプがウィザードで選択されます。デフォルトの選択のまま残して [Next: Configure Instance Details を選択できます。
6. [Configure Instance Details] ページで、[Network] リストから作成した VPC を選択し、[Subnet] リストからサブネットを選択します。デフォルト設定の残りを終了して、ウィザードに [Add Tags] ページが表示されるまでページを移動します。
7. [Add Tags] ページで、Name タグを利用してインスタンスにタグ付けができます (例: Name=MyWebServer)。これにより、インスタンスが起動した後、Amazon EC2 コンソールでインスタンスを識別できます。完了したら、[次の手順: セキュリティグループの設定] を選択します。
8. [Configure Security Group] ページで、ウィザードは自動的に launch-wizard-x セキュリティグループを定義して、インスタンスに接続できるようにします。[Review and Launch] を選択します。

   Important

   ウィザードは、すべての IP アドレス (0.0.0.0/0) が SSH または RDP を使用してインスタンスへのアクセスを許可するセキュリティグループルールを作成します。これは、短期間の実習では許容されますが、本稼働環境では安全ではありません。実稼働環境では、特定の IP アドレスまたは特定のアドレス範囲にのみ、インスタンスへのアクセスを限定します。
9. [Review Instance Launch] ページで、[Launch] を選択します。
10. [Select an existing key pair or create a new key pair] ダイアログボックスで、既存のキーペアを選択するか、新しいキーペアを作成できます。新しいキーペアを作成する場合、ファイルをダウンロードして、安全な場所に保存してください。起動後にインスタンスに接続するには、プライベートキーの内容が必要になります。

    インスタンスを起動するには、確認のチェックボックスをオンにし、続いて [Launch Instances] を選択します。
11. 確認ページで、[View Instances] を選択して、[Instances] ページのインスタンスを表示します。インターフェイスを選択し、[Description] タブで詳細を表示します。[Private IPs] フィールドは、サブネットの IP アドレスの範囲からのインスタンスに割り当てられたプライベート IP アドレスが表示されます。

Amazon EC2 起動ウィザードで使用できるオプションの詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「インスタンスの起動」を参照してください。

ステップ 3: Elastic IP アドレスをインスタンスに割り当てる

これまでのステップでは、パブリックサブネット内にインスタンスを起動しました。パブリックサブネットはインターネットゲートウェイへのルートが存在するサブネットです。しかし、サブネットのインスタンスが、インターネットと通信するには、パブリック IPv4 アドレスも必要です。デフォルトでは、デフォルト以外の VPC のインスタンスはパブリック IPv4 アドレスを割り当てられません。このステップでは、Elastic IP アドレスをアカウントに割り当て、それをインスタンスに関連付けます。

Elastic IP アドレスを割り当てるには

1. https://console.aws.amazon.com/vpc/ で Amazon VPC コンソールを開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. [新しいアドレスの割り当て] を選択し、続いて [割り当て] を選択します。
4. リストで Elastic IP アドレスを選び、[Actions] を選択してから [Associate Address] を選択します。
5. [リソースタイプ] で、[インスタンス] が選択されていることを確認します。[インスタンス] リストからインスタンスを選択します。完了したら、[関連付ける] を選択します。

インスタンスが、インターネットからアクセス可能になりました。SSH またはホームネットワークからリモートデスクトップを使って、Elastic IP アドレスからインスタンスに接続できます。Linux インスタンスに接続する方法の詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Linux インスタンスへの接続」を参照してください。Windows インスタンスに接続する方法の詳細については、Windows インスタンス用 Amazon EC2 ユーザーガイドの「Windows インスタンスへの接続」を参照してください。

ステップ 4: クリーンアップする

このまま VPC でインスタンスを使用し続けることができます。もしこのインスタンスが不要な場合には、アドレスへの料金発生を防ぐためにインスタンスを終了して Elastic IP アドレスを解除してください。VPC を削除することもできます — この演習で作成された VPC および VPC のコンポーネントには料金が発生しないことを注記します (サブネットやルートテーブルなど)。

VPC を削除する前に、VPC で実行中のすべてのインスタンスを終了する必要があります。その後、VPC コンソールを使用して VPC とそのコンポーネントを削除できます。

インスタンスを終了し、Elastic IP アドレスを解除して VPC を削除するには

1. https://console.aws.amazon.com/ec2/ で Amazon EC2 コンソールを開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選び、[Actions] を選択し、[Instance State] を選択した後、[Terminate] を選択します。
4. ダイアログボックスで [Release attached Elastic IPs] セクションを展開し、Elastic IP アドレスの横にあるチェックボックスを選択します。[Yes, Terminate] を選択します。
5. https://console.aws.amazon.com/vpc/ で Amazon VPC コンソールを開きます。
6. 画面左枠のナビゲーションペインで、[VPC] を選択します。
7. VPC を選び、[Actions] を選択後、[Delete VPC] を選択します。

8. 確認を求めるメッセージが表示されたら、[VPC の削除] を選択します。

次のステップ

デフォルト以外の VPC を作成したら、次の操作を実行できます。

- VPC にサブネットを追加します。詳細については、「VPC にサブネットを作成する (p. 117)」を参照してください。
- VPC とサブネットの IPv6 サポートを有効にします。詳細については、「IPv6 CIDR ブロックと VPC の関連付け (p. 119)」および「IPv6 CIDR ブロックとサブネットの関連付け (p. 120)」を参照してください。
- プライベートサブネットのインスタンスがインターネットにアクセスできるようにします。詳細については、「VPC の NAT デバイス (p. 236)」を参照してください。

Amazon VPC での IPv6 の使用開始

次のステップでは、IPv6 アドレス指定をサポートするデフォルト以外の VPC を作成する方法について説明します。

この演習を完了するには、以下の作業を行います。

- IPv6 CIDR ブロックと 1 つのパブリックサブネットを持つデフォルトでない VPC を作成します。サブネットを使うと、インスタンスをセキュリティや運用上の必要に応じてグループ化することができます。パブリックサブネットとは、インターネットゲートウェイを通してインターネットにアクセスするサブネットです。
- 特定のポートのみからトラフィックを許可するセキュリティグループをインスタンスに作成します。
- サブネット内に Amazon EC2 インスタンスを起動し、起動時に IPv6 アドレスをインスタンスに関連付けます。IPv6 アドレスはグローバルに一意であり、インスタンスがインターネットと通信できるようにします。
- VPC の IPv6 CIDR ブロックをリクエストできます。このオプションを選択すると、IPV6 CIDR ブロックをアドバタイズする場所であるネットワーク境界グループを設定できます。ネットワーク境界グループを設定すると、CIDR ブロックがこのグループに制限されます。

IPv4 アドレスと IPv6 アドレスの詳細については、「VPC の IP アドレス指定」を参照してください。

VPC にローカルゾーンを使用する場合は、VPC を作成してから、ローカルゾーンにサブネットを作成します。詳細については、「the section called “VPC を作成する” (p. 116)」および「the section called “VPC にサブネットを作成する” (p. 117)」を参照してください。

タスク

- ステップ 1: VPC を作成する (p. 15)
- ステップ 2: セキュリティグループを作成する (p. 18)
- ステップ 3: インスタンスを起動する (p. 19)

ステップ 1: VPC を作成する

このステップでは、Amazon VPC コンソールで Amazon VPC ウィザードを使用して VPC を作成します。デフォルトでは、ウィザードは次のステップを自動的に行います。

- /16 の IPv4 CIDR ブロックを持つ VPC を作成し、/56 の IPv6 CIDR ブロックを VPC と関連付けます。詳細については、「Your VPC」を参照してください。IPv6 CIDR ブロックのサイズ (/56) は固定されて

おり、IPv6 アドレスの範囲は、Amazon の IPv6 アドレスのプールから自動的に割り当てられます (独自の IPv6 アドレス範囲を指定することはできません)。

- インターネットゲートウェイをこの VPC にアタッチします。インターネットゲートウェイの詳細については、「インターネットゲートウェイ」を参照してください。
- /24 の IPv4 CIDR ブロックと、/64 の IPv6 CIDR ブロックを持つサブネットを VPC 内に作成します。IPv6 CIDR ブロックのサイズは固定されています (/64)。
- カスタムルートテーブルを作成し、サブネットに関連付けると、サブネットとインターネットゲートウェイ間でトラフィックが転送されます。ルートテーブルに関する詳細については、「ルートテーブル」を参照してください。
- Amazon が提供する IPv6 の CIDR ブロックをネットワーク境界グループに関連付けます。詳細については、「the section called “Local Zones で VPC リソースを拡張する” (p. 149)」を参照してください。

Note

この演習では、VPC ウィザードの最初のシナリオを扱います。そのほかのシナリオについては、「Amazon VPC のシナリオ」を参照してください。

デフォルトのアベイラビリティーゾーンに VPC を作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションバーの右上から、VPC を作成するリージョンをメモしておきます。別のリージョンから VPC 内にインスタンスを起動できないので、この演習が終了するまでは同じリージョンで作業を続けるようにしてください。詳細については、Linux インスタンス用の Amazon EC2 ユーザーガイドの「リージョンとアベイラビリティーゾーン」を参照してください。
3. ナビゲーションペインで、[VPC ダッシュボード] を選択し、[VPC ウィザードの起動] を選択します。

   Note

   ナビゲーションペインで [お客様の VPC] を選択しないでください。そのページの [VPC の作成] ボタンを使用して VPC ウィザードにアクセスすることはできません。
4. 実装する設定のオプション (例えば、[VPC with a Single Public Subnet (単一のパブリックサブネットを持つ VPC)]) を選択し、[Select (選択)] を選択します。
5. 設定ページで [VPC name] に VPC の名前 (例: `my-vpc`) を入力後、[Subnet name] にサブネットの名前を入力します。これは、VPC およびサブネットを作成後に Amazon VPC コンソールにおいてそれらを識別するのに役立ちます。
6. [IPv4 CIDR block] の場合は、デフォルト設定 (`10.0.0.0/16`) のままにするか、独自の設定を指定できます。詳細については、「VPC のサイズ設定」を参照してください。

   [IPv6 CIDR block] の場合は、[Amazon-provided IPv6 CIDR block] を選択します。
7. [Public subnet's IPv4 CIDR] の場合は、デフォルト設定のままにするか、または独自の設定を指定します。[Public subnet's IPv6 CIDR] の場合は、[Specify a custom IPv6 CIDR] を選択します。IPv6 サブネットは、デフォルトの 16 進法のキーペア値 (`00`) のまま残すことができます。
8. 他の設定についてもデフォルト設定のままにして、[Create a VPC] を選択します。
9. ステータスウィンドウに、作業の進行状況が表示されます。作業が完了したら、[OK] を選択してステータスウィンドウを閉じます。
10. [Your VPCs] ページは、今作成したデフォルト VPC とその他の VPC が表示されます。

ローカルゾーンに VPC を作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションバーの右上から、VPC を作成するリージョンをメモしておきます。別のリージョンから VPC 内にインスタンスを起動できないので、この演習が終了するまでは同じリージョンで作業を続

けるようにしてください。詳細については、Linux インスタンス用の Amazon EC2 ユーザーガイドの「リージョンとゾーン」を参照してください。

3. ナビゲーションペインで、[VPC ダッシュボード] を選択し、[VPC ウィザードの起動] を選択します。

   Note

   ナビゲーションペインで [お客様の VPC] を選択しないでください。そのページの [VPC の作成] ボタンを使用して VPC ウィザードにアクセスすることはできません。

4. 実装する設定のオプション (例えば、[VPC with a Single Public Subnet (単一のパブリックサブネットを持つ VPC)]) を選択し、[Select (選択)] を選択します。
5. 設定ページで [VPC name] に VPC の名前 (例: `my-vpc`) を入力後、[Subnet name] にサブネットの名前を入力します。これは、VPC およびサブネットを作成後に Amazon VPC コンソールにおいてそれらを識別するのに役立ちます。
6. [IPv4 CIDR ブロック] の場合は、CIDR ブロックを指定します。詳細については、「VPC のサイズ設定」を参照してください。
7. [IPv6 CIDR block] の場合は、[Amazon-provided IPv6 CIDR block] を選択します。
8. [Network Border Group (ネットワーク境界グループ)] の場合は、AWS が IP アドレスをアドバタイズするグループを選択します。
9. 他の設定についてもデフォルト設定のままにして、[Create a VPC] を選択します。
10. ステータスウィンドウに、作業の進行状況が表示されます。作業が完了したら、[OK] を選択してステータスウィンドウを閉じます。
11. [Your VPCs] ページは、今作成したデフォルト VPC とその他の VPC が表示されます。

VPC に関する情報を表示する

VPC を作成したら、該当するサブネット、インターネットゲートウェイ、ルートテーブルに関する情報を表示できます。作成した VPC には 2 個のルートテーブル (すべての VPC にあるメインルートテーブル、およびウィザードから作成したカスタムルートテーブル) があります。このカスタムルートテーブルにはサブネットが関連付けられます。したがって、サブネットへのトラフィックの流れ方は、このテーブル内のルートから決定されます。VPC に新規のサブネットを追加する場合、そのサブネットはデフォルトでメインルートテーブルを使用します。

VPC に関する情報を表示するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。作成した VPC の名前と ID を書き留めておきます ([Name] および [VPC ID] 列を参照します)。この情報は、VPC に関連付けられているコンポーネントを識別するために使用します。

   Local Zones を使用する場合、IPv6 (ネットワーク境界グループ) エントリは VPC ネットワーク境界グループ (`us-west-2-lax-1` など) を示します。
3. ナビゲーションペインで、[Subnets (サブネット)] を選択します。コンソールには、VPC を作成した時に作成されたサブネットが表示されます。[Name] 列からその名前でサブネットを識別したり、前記の手順で VPC 情報を入手して、[VPC] 列から確認できます。
4. ナビゲーションペインで、[Internet Gateways] を選択します。VPC にアタッチしているインターネットゲートウェイは、[VPC] 列から確認でき、ここから VPC の ID と名前 (該当する場合) が表示されます。
5. ナビゲーションペインで、[Route Tables] を選択します。VPC に関連付けられる 2 つのルートテーブルがあります。カスタムルートテーブル ([Main] 列に [No] と表示されているもの) を選択し、[Routes] タブを選択すると、ルート情報が詳細ペインに表示されます。

   - テーブル内の最初の 2 行はローカルルートです。VPC 内のインスタンスが IPv4 および IPv6 経由で通信できるようにします。これらのルートは削除できません。

- 次の行は、Amazon VPC ウィザードで追加したルートを表します。これにより、VPC の外の IPv4 アドレス (`0.0.0.0/0`) に向けられたトラフィックが、サブネットからインターネットゲートウェイに流れるようになります。
- 次の行は、VPC の外の IPv6 アドレス (`::/0`) に向けられたトラフィックが、サブネットからインターネットゲートウェイに流れるようにするルートを表します。

6. メインルートテーブルを選択します。メインルートテーブルはローカルルートだけで、それ以外のルートがありません。

ステップ 2: セキュリティグループを作成する

セキュリティグループは、仮想ファイアウォールとして機能し、関連付けられたインスタンスへのトラフィックを管理します。セキュリティグループを使用するには、インスタンスの出力トラフィックを制御するインバウンドルール、およびインスタンスから送信されるトラフィックを制御するアウトバンドルールを追加します。インスタンスにセキュリティグループを関連付けるには、インスタンスの起動時にセキュリティグループを指定します。

VPC には、デフォルトのセキュリティグループが用意されています。起動時に他のセキュリティグループに関連付けられていないインスタンスは、デフォルトのセキュリティグループに関連付けられます。この演習では、新しいセキュリティグループを作成し、`WebServerSG`、続いて VPC でインスタンスを起動するときにこのセキュリティグループを指定します。

WebServerSG セキュリティグループを作成する

Amazon VPC コンソールを使いセキュリティグループを作成することができます。

WebServerSG セキュリティグループを作成し、ルールを追加するには

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 [セキュリティグループ] を選択して、[セキュリティグループの作成] を選択します。
3. [Group name] で、セキュリティグループ名に `WebServerSG` と入力し、説明を入力します。オプションで [Name tag] フィールドを使用して、キー `Name` および指定する値で、セキュリティグループのタグを作成できます。
4. [VPC] メニューで VPC の ID を選択し、[Yes, Create] を選択します。
5. 作成した `WebServerSG` セキュリティグループを選択します (グループ名は [Group Name] 列で確認できます)。
6. [インバウンドルール] タブで [編集] を選択して、次に示すようにインバウンドトラフィックのルールを追加します。

   a. [Type] (タイプ) で、[HTTP] を選択し、[Source] (送信元) フィールドに「`::/0`」と入力します。

   b. [Add another rule] (別のルールを追加する) を選択し、[Type] (タイプ) で [HTTPS] を選択し、[Source] フィールドに「`::/0`」と入力します。

   c. [別のルールの追加] を選択します。Linux インスタンスを起動した場合は、[Type] で [SSH] を選択します。Windows インスタンスを起動した場合は、[RDP] を選択します。[Source] フィールドに、ネットワークのパブリック IPv6 アドレスの範囲を入力します。このアドレス範囲が不明の場合は、この実習では `::/0` を使用してください。

      Important

      `::/0` を使用すると、すべての IPv6 アドレスから SSH や RDP 経由でインスタンスにアクセスできるようになります。これは、短期間の実習では許容されますが、本稼働環境では安全ではありません。実稼働環境では、特定の IP アドレスまたは特定のアドレス範囲にのみ、インスタンスへのアクセスを限定します。

d. [Save] を選択します。

ステップ 3: インスタンスを起動する

EC2 インスタンスを VPC 内で起動するときは、どのサブネットでインスタンスを起動するかを指定する必要があります。この場合、作成した VPC のパブリックサブネットにインスタンスを起動します。Amazon EC2 コンソールで Amazon EC2 起動ウィザードを使用して、インスタンスを起動します。

インターネットからインスタンスにアクセスできることを確認するには、起動時に、サブネットの範囲からインスタンスへ IPv6 アドレスを割り当てます。これにより、インスタンスは、IPv6 経由でインターネットと通信できるようになります。

EC2 インスタンスを VPC 内で起動するには

EC2 インスタンスを VPC 内で起動する前に、IPv6 IP アドレスが自動的に割り当てされるように VPC のサブネットを設定します。詳細については、「the section called “サブネットのパブリック IPv6 アドレス属性を変更する” (p. 128)」を参照してください。

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションバーの右上で、VPC とセキュリティグループを作成したリージョンが選択されていることを確認します。
3. ダッシュボードから、[Launch Instance] を選択します。
4. ウィザードの最初のページで、使用する AMI を選択します。この演習では、Amazon Linux AMI または Windows AMI を選択することをお勧めします。
5. [Choose an Instance Type] ページで、起動するインスタンスのハードウェア設定とサイズを選択できます。デフォルトでは、お客様が選択した AMI に基づいて使用可能な最初のインスタンスタイプがウィザードで選択されます。デフォルト設定のまま、[Next: Configure Instance Details] を選択できます。
6. [Configure Instance Details] ページで、[Network] リストから作成した VPC を選択し、[Subnet] リストからサブネットを選択します。
7. [Auto-assign IPv6 IP] で、[Enable] を選択します。
8. デフォルト設定の残りを終了して、ウィザードに [Add Tags] ページが表示されるまでページを移動します。
9. [Add Tags] ページで、`Name` タグを利用してインスタンスにタグ付けができます (例: `Name=MyWebServer`)。これにより、インスタンスが起動した後、Amazon EC2 コンソールでインスタンスを識別できます。完了したら、[次の手順: セキュリティグループの設定] を選択します。
10. [Configure Security Group] ページで、ウィザードは自動的に launch-wizard-x セキュリティグループを定義して、インスタンスに接続できるようにします。代わりに、[Select an existing security group] オプションを選択し、以前から作成してある [WebServerSG] グループを選び、続いて [Review and Launch] を選択します。
11. [Review Instance Launch] ページでインスタンスの詳細を確認後、[Launch] を選択します。
12. [Select an existing key pair or create a new key pair] ダイアログボックスで、既存のキーペアを選択するか、新しいキーペアを作成できます。新しいキーペアを作成する場合、ファイルをダウンロードして、安全な場所に保存してください。起動後にインスタンスに接続するには、このプライベートキーの内容が必要になります。

    インスタンスを起動するには、確認のチェックボックスをオンにし、[Launch Instances] を選択します。
13. 確認ページで、[View Instances] を選択して、[Instances] ページのインスタンスを表示します。インターフェイスを選択し、[Description] タブで詳細を表示します。[Private IPs] フィールドには、サブネットの IPv4 アドレスの範囲のインスタンスに割り当てられたプライベート IPv4 アドレスが表示されます。[IPv6 IPs] フィールドには、サブネットの IPv6 アドレスの範囲のインスタンスに割り当てられたプライベート IPv6 アドレスが表示されます。

Amazon EC2 起動ウィザードで使用できるオプションの詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「インスタンスの起動」を参照してください。

SSH またはホームネットワークからリモートデスクトップを使って、IPv6 アドレスからインスタンスに接続できます。ローカルコンピュータに IPv6 アドレスがあり、IPv6 を使用するように設定されている必要があります。Linux インスタンスに接続する方法の詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Linux インスタンスへの接続」を参照してください。Windows インスタンスに接続する方法の詳細については、「Windows インスタンス用 Amazon EC2 ユーザーガイド」の「RDP を使用して Windows インスタンスに接続する」を参照してください。

Note

インターネット、SSH、RDP を介して IPv4 アドレス経由でインスタンスにアクセスできるようにするには、Elastic IP アドレス (静的なパブリック IPv4 アドレス) をインスタンスに関連付け、IPv4 経由のアクセスを許可するようにセキュリティグループルールを調整する必要があります。詳細については、「Amazon VPC の使用を開始する (p. 11)」を参照してください。

Amazon VPC コンソールウィザードの設定

Amazon VPC コンソールウィザードを使用して、次のいずれかのデフォルト以外の VPC 設定を作成できます。

設定

1 つのパブリックサブネットを持つ VPC

このシナリオの設定には、1 つのパブリックサブネットを持つ Virtual Private Cloud (VPC) と、インターネットを介した通信を有効にするインターネットゲートウェイが含まれます。この設定は、ブログや簡単なウェブサイトなど、単層でパブリックなウェブアプリケーションを実行する必要がある場合にお勧めします。

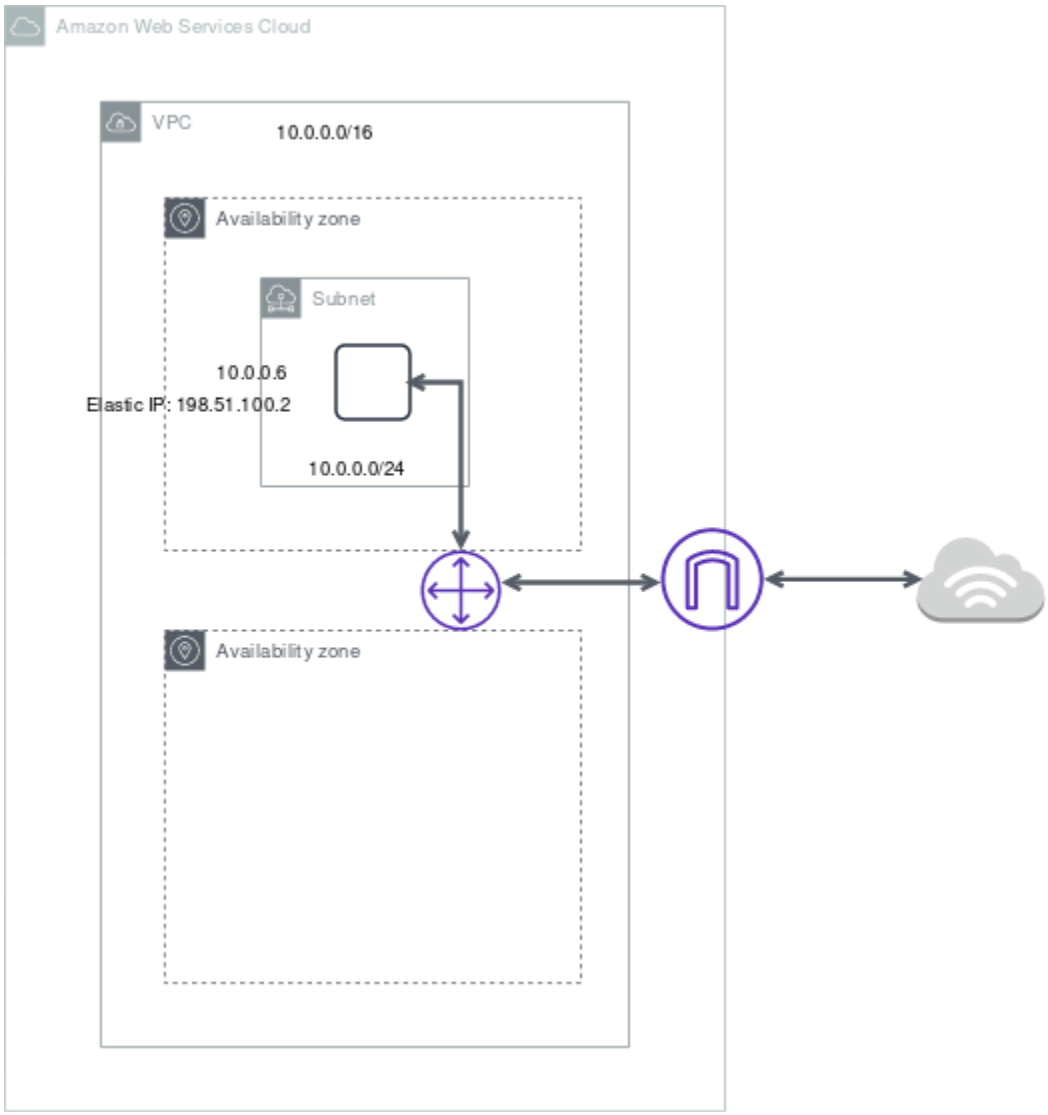
また、このシナリオは、オプションで IPv6 に設定することもできます。VPC ウィザードを使用して、関連する IPv6 CIDR ブロックで VPC およびサブネットを作成できます。パブリックサブネットに起動されたインスタンスは、IPv6 を取得できます。また、IPv6 を使用して通信できます。IPv4 アドレスと IPv6 アドレスの詳細については、「VPC の IP アドレス指定 (p. 124)」を参照してください。

EC2 インスタンスソフトウェアの管理については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Linux インスタンスでソフトウェアを管理する」を参照してください。

目次

Overview

次の図は、このシナリオの設定に重要なコンポーネントを示しています。



Note

「Amazon VPC の使用を開始する (p. 11)」が完了している場合、このシナリオはすでに Amazon VPC コンソールの VPC ウィザードを使用して実装済みであることを意味します。

このシナリオの設定には、以下の項目が含まれています。

- IPv4 CIDR ブロックサイズが /16 (例: 10.0.0.0/16) の Virtual Private Cloud (VPC)。65,536 個のプライベート IPv4 アドレスを提供します。
- サイズ /24 の IPv4 CIDR ブロック (例: 10.0.0.0/24) を持つサブネット。256 個のプライベート IPv4 アドレスを提供します。
- インターネットゲートウェイ。VPC をインターネットおよび他の AWS サービスに接続します。
- サブネット範囲のプライベート IPv4 アドレス (例: 10.0.0.6) と Elastic IPv4 アドレス (例: 198.51.100.2) を持つインスタンス。前者はインスタンスが VPC 内の他のインスタンスと通信できるようにするアドレスであり、後者はインスタンスがインターネットに接続してインターネットからインスタンスにアクセスできるようにするパブリック IPv4 アドレスです。
- サブネットに関連付けられているカスタムルートテーブル。ルートテーブルのエントリにより、サブネットのインスタンスは IPv4 を使用して VPC 内の他のインスタンスと通信したり、インターネット

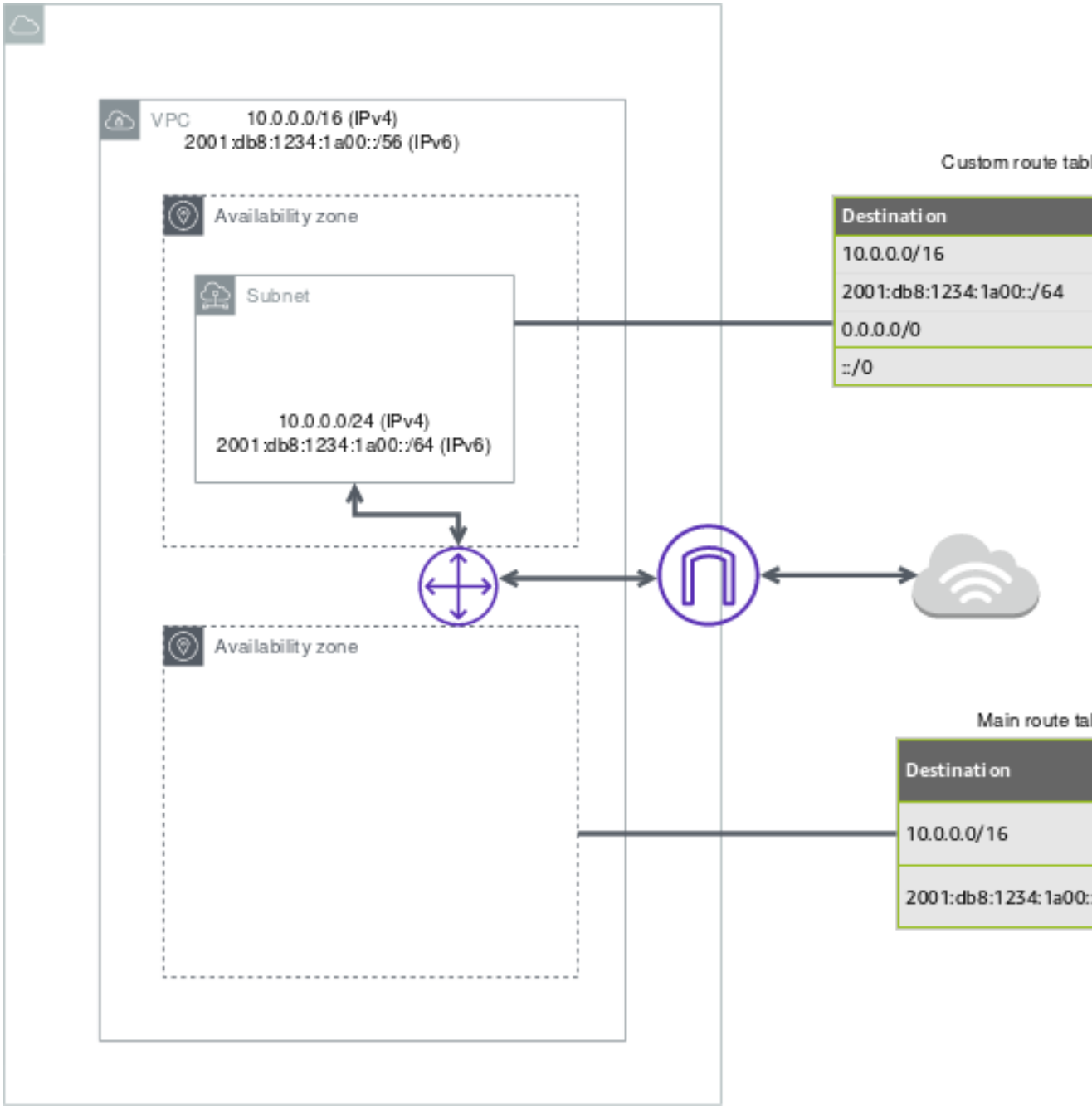
で直接通信したりできます。サブネットに関連付けられているルートテーブルにインターネットゲートウェイへのルートがある場合、そのサブネットはパブリックサブネットと呼ばれます。

サブネットの詳細については、「VPC とサブネット (p. 106)」を参照してください。インターネットゲートウェイの詳細については、「インターネットゲートウェイ (p. 221)」を参照してください。

IPv6 の概要

オプションでこのシナリオの IPv6 を有効にできます。上記のコンポーネントに加えて、この設定には、次に示す情報が含まれます。

- VPC に関連付けられたサイズ /56 の IPv6 CIDR ブロック (例: 2001:db8:1234:1a00::/56)。Amazon の CIDR は自動的に割り当てられます。独自にアドレス範囲を選択することはできません。
- パブリックサブネットに関連付けられたサイズ /64 の IPv6 CIDR ブロック (例: 2001:db8:1234:1a00::/64)。VPC に割り当てられた範囲からサブネットの範囲を選択できます。サブネットの IPv6 CIDR ブロックのサイズを選択することはできません。
- サブネットの範囲 (例: 2001:db8:1234:1a00::123) からインスタンスへ割り当てられた IPv6 アドレス。
- カスタムエントリテーブル内のルートテーブルエントリ。VPC のインスタンスが IPv6 を使用して相互通信したり、インターネット経由で直接通信したりできるようにします。



Routing

VPC には暗示的なルーターがあります (上の設定図を参照)。このシナリオでは、VPC ウィザードによって、送信先が VPC 外のアドレスであるトラフィックすべてをインターネットゲートウェイにルーティングするカスタムルートテーブルを作成し、それをサブネットに関連付けます。

次の表は、上の設定図に含まれている例のルートテーブルを示しています。最初のエントリは、VPC のローカル IPv4 ルーティングのデフォルトエントリです。このエントリによって、この VPC 内のインスタンスが相互に通信できるようになります。2 番目のエントリは、他のすべての IPv4 サブネットトラフィックをインターネットゲートウェイ (例: `igw-1a2b3c4d`) にルーティングします。

送信先	ターゲット
10.0.0.0/16	ローカル
0.0.0.0/0	igw-id

IPv6 のルーティング

IPv6 CIDR ブロックを VPC およびサブネットと関連付ける場合は、IPv6 トラフィックの別のルートをルートテーブルに含める必要があります。VPC 内の IPv6 通信を有効化した場合のシナリオのカスタムルートテーブルを次の表に示します。2 番目のエントリは、IPv6 経由で VPC 内のローカルルーティングに自動的に追加されたデフォルトルートです。4 番目のエントリは、他のすべての IPv6 サブネットトラフィックをインターネットゲートウェイにルーティングします。

送信先	ターゲット
10.0.0.0/16	ローカル
2001:db8:1234:1a00::/56	ローカル
0.0.0.0/0	igw-id
::/0	igw-id

Security

AWS では、セキュリティグループとネットワーク ACL という 2 つの機能を使用して、VPC のセキュリティを強化できます。セキュリティグループは、インスタンスのインバウンドトラフィックとアウトバウンドトラフィックをコントロールします。ネットワーク ACL は、サブネットのインバウンドトラフィックとアウトバウンドトラフィックをコントロールします。通常、セキュリティグループで用が足りますが、VPC に追加のセキュリティレイヤーが必要な場合は、ネットワーク ACL を使用することもできます。詳細については、「」を参照してくださいAmazon VPC でのインターネットワークトラフィックのプライバシー (p. 165)

このシナリオでは、セキュリティグループを使用し、ネットワーク ACL は使用しません。ネットワーク ACL を使用する場合は、「1 つのパブリックサブネットを持つ VPC に対する推奨ネットワーク ACL ルール (p. 27)」を参照してください。

VPC には、デフォルトのセキュリティグループ (p. 190)が用意されています。VPC 内に起動されるインスタンスは、起動時に別のセキュリティグループを指定しないと、デフォルトのセキュリティグループに自動的に関連付けられます。デフォルトのセキュリティグループに特定のルールを追加できますが、そのルールは VPC 内に起動する他のインスタンスには適切でない場合があります。代わりに、ウェブサーバー用にカスタムセキュリティグループを作成することをお勧めします。

このシナリオでは、`WebServerSG` という名前のセキュリティグループを作成します。作成したセキュリティグループには、インスタンスからのすべてのトラフィックを許可する単一のアウトバウンドルールがあるだけです。インバウンドトラフィックを有効にし、必要に応じてアウトバウンドトラフィックを制限するようにルールを変更する必要があります。このセキュリティグループは、VPC 内にインスタンスを起動するときに指定します。

以下は、WebServerSG セキュリティグループの IPv4 トラフィック用インバウンドルールとアウトバウンドルールです。

インバウンド

送信元	プロトコル	ポート範囲	コメント
0.0.0.0/0	TCP	80	任意の IPv4 アドレスからウェブサーバーへのインバウンド HTTP アクセスを許可する。
0.0.0.0/0	TCP	443	任意の IPv4 アドレスからウェブサーバーへのインバウンド HTTPS アクセスを許可する。
ネットワークのパブリック IPv4 アドレスの範囲	TCP	22	(Linux インスタンス) ネットワークからの IPv4 経由のインバウンド SSH アクセスを許可する。ローカルコンピュータのパブリック IPv4 アドレスは、http://checkip.amazonaws.com または https://checkip.amazonaws.com などのサービスを使用して取得できます。ISP 経由で、またはファイアウォールの内側から静的な IP アドレスなしで接続している場合は、クライアントコンピュータで使用されている IP アドレスの範囲を見つける必要があります。
ネットワークのパブリック IPv4 アドレスの範囲	TCP	3389	(Windows インスタンス) ネットワークからの IPv4 経由のインバウンド RDP アクセスを許可する。
The security group ID (sg-xxxxxxxx)	All	All	(Optional) Allow inbound traffic from other instances associated with this security group. This rule is automatically added to the default security group for the VPC; for any custom security group you create, you must manually add the rule to allow this type of communication.

アウトバウンド (オプション)

送信先	プロトコル	ポート範囲	コメント

0.0.0.0/0	All	All	Default rule to allow all outbound access to any IPv4 address. If you want your web server to initiate outbound traffic, for example, to get software updates, you can keep the default outbound rule. Otherwise, you can remove this rule.

IPv6 のセキュリティグループルール

IPv6 CIDR ブロックを VPC およびサブネットと関連付ける場合は、別のルールをセキュリティグループに追加して、ウェブサーバーインスタンスのインバウンドおよびアウトバウンドの IPv6 トラフィックを制御する必要があります。このシナリオでは、ウェブサーバーは、IPv6 経由ですべてのインターネットトラフィックを受信したり、IPv6 経由でローカルネットワークから SSH または RDP トラフィックを受信したりできます。

WebServerSG セキュリティグループの IPv6 固有ルールを次に示します (上記のルールは含まない)。

インバウンド			
送信元	プロトコル	ポート範囲	コメント
::/0	TCP	80	任意の IPv6 アドレスからウェブサーバーへのインバウンド HTTP アクセスを許可する。
::/0	TCP	443	任意の IPv6 アドレスからウェブサーバーへのインバウンド HTTPS アクセスを許可する。
ネットワークの IPv6 アドレス範囲	TCP	22	(Linux インスタンス) ネットワークからの IPv6 経由のインバウンド SSH アクセスを許可する。
ネットワークの IPv6 アドレス範囲	TCP	3389	(Windows インスタンス) ネットワークからの IPv6 経由のインバウンド RDP アクセスを許可する
アウトバウンド (オプション)			
送信先	プロトコル	ポート範囲	コメント
::/0	All	All	Default rule to allow all outbound access to any IPv6 address. If you want your web server to initiate outbound traffic, for example, to

					get software updates, you can keep the default outbound rule. Otherwise, you can remove this rule.

1 つのパブリックサブネットを持つ VPC に対する推奨ネットワーク ACL ルール

次の表に、推奨ルールを示します。これらのルールでは、明示的に必要なトラフィックを除き、すべてのトラフィックをブロックします。

Inbound					
ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント
100	0.0.0.0/0	TCP	80	許可	任意の IPv4 アドレスからのインバウンド HTTP トラフィックを許可します。
110	0.0.0.0/0	TCP	443	許可	任意の IPv4 アドレスからのインバウンド HTTPS トラフィックを許可します。
120	ホームネットワークのパブリック IPv4 アドレスの範囲	TCP	22	許可	(インターネットゲートウェイを介した) ホームネットワークからのインバウンド SSH トラフィックを許可します。
130	ホームネットワークのパブリック IPv4 アドレスの範囲	TCP	3389	許可	(インターネットゲートウェイを介した) ホームネットワークからのインバウンド RDP トラフィックを許可します。
140	0.0.0.0/0	TCP	32768-65535	許可	リクエストに応答しているか、送信元がサブネットである、インターネット上のホストからのインバウンドリターントラフィックを許可します。

					この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv4 トラフィックを拒否します (変更不可)。

Outbound

ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント
100	0.0.0.0/0	TCP	80	許可	サブネットからインターネットへのアウトバウンド HTTP トラフィックを許可します.
110	0.0.0.0/0	TCP	443	許可	サブネットからインターネットへのアウトバウンド HTTPS トラフィックを許可します.

120	0.0.0.0/0	TCP	32768-65535	許可	インターネット上のクライアントに対するアウトバウンド応答を許可します (たとえば、サブネット内のウェブサーバーを訪問するユーザーに対するウェブページの提供など)。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのアウトバウンド IPv4 トラフィックを拒否します (変更不可)。

IPv6 に推奨されるネットワーク ACL ルール

IPv6 サポートを実装し、関連付けられた IPv6 CIDR ブロックを持つ VPC とサブネットを作成した場合は、別のルールをネットワーク ACL に追加して、インバウンドおよびアウトバウンド IPv6 トラフィックを制御する必要があります。

ネットワーク ACL の IPv6 固有のルールを以下に示します (上記のルールは含みません)。

Inbound					
ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント
150	::/0	TCP	80	許可	任意の IPv6 アドレスからのインバウンド HTTP トラフィックを許可します。
160	::/0	TCP	443	許可	任意の IPv6 アドレスからのインバウンド HTTPS トラ

					フィックを許可します。
170	ホームネットワークの IPv6 アドレス範囲	TCP	22	許可	(インターネットゲートウェイを介した) ホームネットワークからのインバウンド SSH トラフィックを許可します。
180	ホームネットワークの IPv6 アドレス範囲	TCP	3389	許可	(インターネットゲートウェイを介した) ホームネットワークからのインバウンド RDP トラフィックを許可します。
190	::/0	TCP	32768-65535	許可	リクエストに応答しているか、送信元がサブネットである、インターネット上のホストからのインバウンドリターントラフィックを許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv6 トラフィックを拒否します (変更不可)。
Outbound					
ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント

130	::/0	TCP	80	許可	サブネットからインターネットへのアウトバウンド HTTP トラフィックを許可します.
140	::/0	TCP	443	許可	サブネットからインターネットへのアウトバウンド HTTPS トラフィックを許可します.
150	::/0	TCP	32768-65535	許可	インターネット上のクライアントに対するアウトバウンド応答を許可します (たとえば、サブネット内のウェブサーバーを訪問するユーザーに対するウェブページの提供など) 。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのアウトバウンド IPv6 トラフィックを拒否します (変更不可)。

パブリックサブネットとプライベートサブネットを持つ VPC (NAT)

このシナリオの設定には、パブリックサブネットとプライベートサブネットを持つ Virtual Private Cloud (VPC) が含まれます。このシナリオは、パブリックにはアクセスできないバックエンドサーバーを維持しながら、パブリックなウェブアプリケーションを実行する場合にお勧めします。一般的な例としては、パブリックサブネットのウェブサーバーとプライベートサブネットのデータベースサーバーを持つ多階層

のウェブサイトが挙げられます。ウェブサーバーがデータベースサーバーと通信できるように、セキュリティとルーティングを設定できます。

パブリックサブネットのインスタンスはアウトバウンドトラフィックを直接インターネットに送信できますが、プライベートサブネットのインスタンスはできません。代わりに、プライベートサブネットのインスタンスは、パブリックサブネットに存在するネットワークアドレス変換 (NAT) ゲートウェイを使用して、インターネットにアクセスできます。データベースサーバーは、NAT ゲートウェイを通じてインターネットに接続してソフトウェアアップデートを行うことができますが、インターネットはデータベースサーバーへの接続を確立できません。

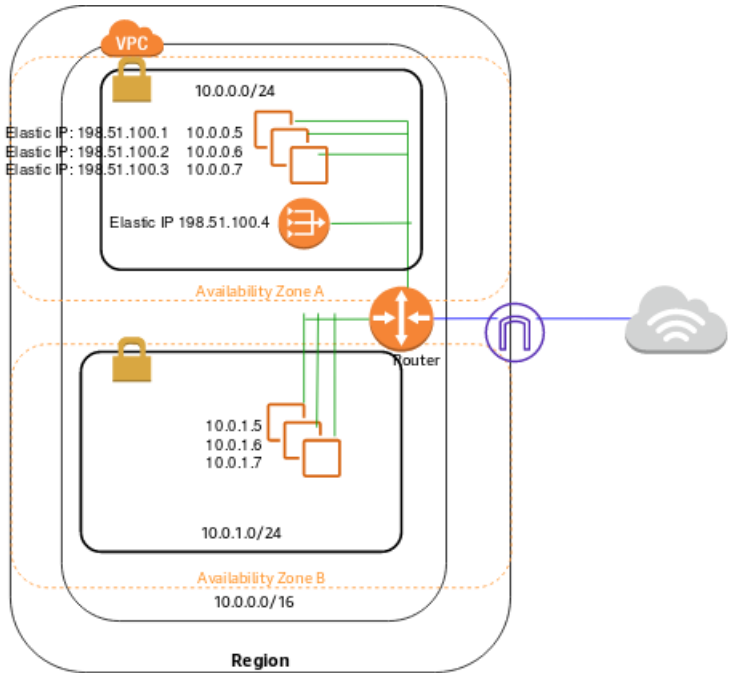
このシナリオは、オプションで IPv6 に設定することもできます。VPC ウィザードを使用して、関連する IPv6 CIDR ブロックで VPC およびサブネットを作成できます。サブネットに起動されたインスタンスは、IPv6 を取得できます。また、IPv6 を使用して通信できます。プライベートサブネットのインスタンスは、Egress-Only インターネットゲートウェイを使用して IPv6 経由でインターネットに接続できますが、IPv6 経由でプライベートインスタンスに接続を確立することはできません。IPv4 アドレスと IPv6 アドレスの詳細については、「VPC の IP アドレス指定 (p. 124)」を参照してください。

EC2 インスタンスソフトウェアの管理については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Linux インスタンスでソフトウェアを管理する」を参照してください。

目次

Overview

次の図は、このシナリオの設定に重要なコンポーネントを示しています。



このシナリオの設定には、以下の項目が含まれています。

- IPv4 CIDR ブロックサイズが /16 (例: 10.0.0.0/16) の VPC。65,536 個のプライベート IPv4 アドレスを提供します。
- IPv4 CIDR ブロックサイズが /24 (例: 10.0.0.0/24) のパブリックサブネット。256 個のプライベート IPv4 アドレスを提供します。パブリックサブネットは、インターネットゲートウェイへのルートが含まれているルートテーブルに関連付けられているサブネットです。
- IPv4 CIDR ブロックサイズが /24 (例: 10.0.1.0/24) のプライベートサブネット。256 個のプライベート IPv4 アドレスを提供します。
- インターネットゲートウェイ。VPC をインターネットおよび他の AWS サービスに接続します。
- プライベート IPv4 アドレスがサブネットの範囲内 (例: 10.0.0.5、10.0.1.5) のインスタンス。これにより、VPC の各インスタンスは相互に通信できるようになります。
- Elastic IPv4 アドレス (例: 198.51.100.1) を使用するパブリックサブネットのインスタンス。Elastic IPv4 アドレスは、インターネットからインスタンスにアクセスできるようにするパブリック IPv4 アドレスです。インスタンスには、起動時に Elastic IP アドレスではなくパブリック IP アドレスを割り当てることができます。プライベートサブネットのインスタンスは、インターネットからの着信トラフィックを受信する必要がないバックエンドサーバーであるため、パブリック IP アドレスを必要としません。ただし、NAT ゲートウェイを使用して、リクエストをインターネットに送信することができます (次の箇条書きを参照)。
- NAT ゲートウェイとその独自の Elastic IPv4 アドレス。プライベートサブネットのインスタンスは、IPv4 で NAT ゲートウェイ経由でインターネットにリクエストを送信できます (例: ソフトウェアをアップデートする場合)。
- パブリックサブネットに関連付けられているカスタムルートテーブル。このルートテーブルには、サブネットのインスタンスが IPv4 経由で VPC 内の他のインスタンスと通信できるようにするエントリと、サブネットのインスタンスが IPv4 経由でインターネットと直接通信できるようにするエントリが含まれます。
- プライベートサブネットに関連付けられているメインルートテーブル。このルートテーブルには、サブネットのインスタンスが IPv4 経由で VPC 内の他のインスタンスと通信できるようにするエントリと、サブネットのインスタンスが IPv4 経由で NAT ゲートウェイを介してインターネットと通信できるようにするエントリが含まれます。

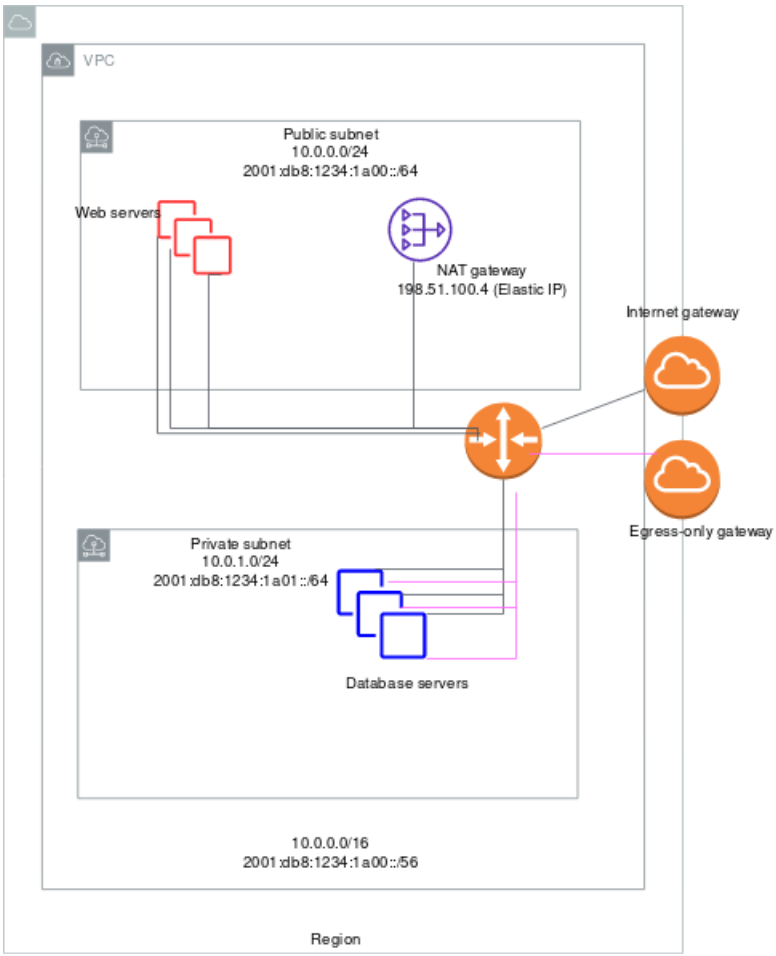
サブネットの詳細については、「VPC とサブネット (p. 106)」を参照してください。インターネットゲートウェイの詳細については、「インターネットゲートウェイ (p. 221)」を参照してください。NAT ゲートウェイの詳細については、「NAT ゲートウェイ (p. 237)」を参照してください

IPv6 の概要

オプションでこのシナリオの IPv6 を有効にできます。上記のコンポーネントに加えて、この設定には、次に示す情報が含まれます。

- VPC に関連付けられたサイズ /56 の IPv6 CIDR ブロック (例: 2001:db8:1234:1a00::/56)。Amazon の CIDR は自動的に割り当てられます。独自にアドレス範囲を選択することはできません。
- パブリックサブネットに関連付けられたサイズ /64 の IPv6 CIDR ブロック (例: 2001:db8:1234:1a00::/64)。VPC に割り当てられた範囲からサブネットの範囲を選択できます。VPC の IPv6 CIDR ブロックのサイズを選択することはできません。
- プライベートサブネットに関連付けられたサイズ /64 の IPv6 CIDR ブロック (例: 2001:db8:1234:1a01::/64)。VPC に割り当てられた範囲からサブネットの範囲を選択できます。サブネットの IPv6 CIDR ブロックのサイズを選択することはできません。
- サブネットの範囲からのインスタンスに割り当てられていた IPv6 アドレス (例: 2001:db8:1234:1a00::1a)。
- Egress-Only インターネットゲートウェイ。ゲートウェイを使用して、IPv6 経由でプライベートサブネット内のインスタンスからインターネットへのリクエストを処理します (ソフトウェアの更新など)。Egress-Only インターネットゲートウェイは、プライベートサブネットのインスタンスが IPv6 経由で通信を始められるようにする場合に必要です。詳細については、「」を参照してくださいEgress-Only インターネットゲートウェイ (p. 228)

- カスタムエントリテーブル内のルートテーブルエントリ。パブリックサブネットのインスタンスが IPv6 を使用して相互通信したり、インターネット経由で直接通信したりできるようにします。
- メインルートテーブルのルートテーブルエントリ。プライベートサブネットのインスタンスが、IPv6 経由で相互に通信したり、Egress-Only インターネットゲートウェイを通じてインターネット通信したりできるようにします。



パブリックサブネット内のウェブサーバーには、次のアドレスがあります。

サーバー	IPv4 アドレス	Elastic IP アドレス	IPv6 アドレス
1	10.0.0.5	198.51.100.1	2001:db8:1234:1a00::1a
2	10.0.0.6	198.51.100.2	2001:db8:1234:1a00::2b
3	10.0.0.7	198.51.100.3	2001:db8:1234:1a00::3c

プライベートサブネット内のデータベースサーバーには、次のアドレスがあります。

サーバー	IPv4 アドレス	IPv6 アドレス
1	10.0.1.5	2001:db8:1234:1a01::1a

サーバー	IPv4 アドレス	IPv6 アドレス
2	10.0.1.6	2001:db8:1234:1a01::2b
3	10.0.1.7	2001:db8:1234:1a01::3c

Routing

このシナリオでは、VPC ウィザードによって、プライベートサブネットで使用されるメインルートテーブルを更新し、カスタムルートテーブルを作成してパブリックサブネットに関連付けます。

このシナリオでは、各サブネットから AWS に向かう (例えば、Amazon EC2 または Amazon S3 エンドポイントに向かう) すべてのトラフィックが、インターネットゲートウェイを介して流れます。プライベートサブネットのデータベースサーバーには Elastic IP アドレスがありません。したがって、インターネットからのトラフィックを直接受け取ることはできません。ただし、パブリックサブネットで NAT デバイスを使用すれば、データベースサーバーでインターネットトラフィックを送受信できます。

追加のサブネットを作成した場合、そのサブネットはデフォルトでメインルートテーブルを使用します。つまり、デフォルトではプライベートサブネットです。サブネットをパブリックにする場合は、関連付けられているルートテーブルをいつでも変更できます。

以下の表は、このシナリオのルートテーブルを示しています。

メインルートテーブル

メインルートテーブルは、プライベートサブネットに関連付けられています。最初のエントリは、VPC のローカルルーティングのデフォルトエントリです。このエントリによって、VPC 内のインスタンスが相互に通信できるようになります。2 番目のエントリは、他のすべてのサブネットトラフィックを NAT ゲートウェイ (例: `nat-12345678901234567`) に送信します。

送信先	ターゲット
`10.0.0.0/16`	ローカル
`0.0.0.0/0`	nat-gateway-id

カスタムルートテーブル

カスタムルートテーブルは、パブリックサブネットに関連付けられています。最初のエントリは、VPC のローカルルーティングのデフォルトエントリです。このエントリによって、この VPC 内のインスタンスが相互に通信できるようになります。2 番目のエントリは、他のすべてのサブネットトラフィックをインターネットゲートウェイ (例: `igw-1a2b3d4d`) 経由でインターネットにルーティングします。

送信先	ターゲット
`10.0.0.0/16`	ローカル
`0.0.0.0/0`	igw-id

IPv6 のルーティング

IPv6 CIDR ブロックを VPC およびサブネットと関連付ける場合は、IPv6 トラフィックの別のルートをルートテーブルに含める必要があります。VPC 内の IPv6 通信を有効化した場合のシナリオのルートテーブルを次の表に示します。

メインルートテーブル

2 番目のエントリは、IPv6 経由で VPC 内のローカルルーティングに自動的に追加されたデフォルトルートです。4 番目のエントリは、他のすべての IPv6 サブネットトラフィックを Egress-Only インターネットゲートウェイにルーティングします。

送信先	ターゲット
10.0.0.0/16	ローカル
2001:db8:1234:1a00::/56	ローカル
0.0.0.0/0	nat-gateway-id
::/0	egress-only-igw-id

カスタムルートテーブル

2 番目のエントリは、IPv6 経由で VPC 内のローカルルーティングに自動的に追加されたデフォルトルートです。4 番目のエントリは、他のすべての IPv6 サブネットトラフィックをインターネットゲートウェイにルーティングします。

送信先	ターゲット
10.0.0.0/16	ローカル
2001:db8:1234:1a00::/56	ローカル
0.0.0.0/0	igw-id
::/0	igw-id

Security

AWS では、セキュリティグループとネットワーク ACL という 2 つの機能を使用して、VPC のセキュリティを強化できます。セキュリティグループは、インスタンスのインバウンドトラフィックとアウトバウンドトラフィックをコントロールします。ネットワーク ACL は、サブネットのインバウンドトラフィックとアウトバウンドトラフィックをコントロールします。通常、セキュリティグループで用が足りますが、VPC に追加のセキュリティレイヤーが必要な場合は、ネットワーク ACL を使用することもできます。詳細については、「」を参照してくださいAmazon VPC でのインターネットワークトラフィックのプライバシー (p. 165)

シナリオ 2 では、セキュリティグループを使用しますが、ネットワーク ACL は使用しません。ネットワーク ACL を使用する場合は、「パブリックサブネットとプライベートサブネット (NAT) を持つ VPC の推奨ネットワーク ACL ルール (p. 41)」を参照してください。

VPC には、デフォルトのセキュリティグループ (p. 190)が用意されています。VPC 内に起動されるインスタンスは、起動時に別のセキュリティグループを指定しないと、デフォルトのセキュリティグループに自動的に関連付けられます。このシナリオでは、デフォルトのセキュリティグループを使用するのではなく、以下のセキュリティグループを作成することをお勧めします。

- WebServerSG: パブリックサブネットでウェブサーバーを起動するときに、このセキュリティグループを指定します。
- DBServerSG: プライベートサブネットでデータベースサーバーを起動するときに、このセキュリティグループを指定します。

セキュリティグループに割り当てられたインスタンスのサブネットは様々です。ただし、このシナリオでは、各セキュリティグループがインスタンスの役割の種類に対応しており、役割ごとにインスタンスが特定のサブネットに属さなければなりません。したがって、このシナリオでは、1 つのセキュリティグループに割り当てられたインスタンスはすべて、同じサブネットに属しています。

次の表では、WebServerSG セキュリティグループの推奨ルールについて説明します。このルールにより、ウェブサーバーがインターネットトラフィックを受信したり、ご利用のネットワークから SSH や RDP トラフィックを受信したりできます。また、ウェブサーバーはプライベートサブネットで、データベースサーバーへのリクエストの読み取り/書き込みを開始し、インターネットにトラフィックを送信できます (例えば、ソフトウェアの更新プログラムを取得するなど)。ウェブサーバーがその他のアウトバウンド通信を開始しないため、デフォルトのアウトバウンドルールは削除されます。

> **Note**
>
> これらの推奨事項には SSH アクセスと RDP アクセスの両方、および Microsoft SQL Server アクセスと MySQL アクセスの両方が含まれます。この場合は、Linux (SSH および MySQL) または Windows (RDP および Microsoft SQL Server) に対するルールのみで十分かもしれません。

WebServerSG: 推奨ルール

インバウンド			
送信元	プロトコル	ポート範囲	コメント
0.0.0.0/0	TCP	80	任意の IPv4 アドレスからウェブサーバーへのインバウンド HTTP アクセスを許可する。
0.0.0.0/0	TCP	443	任意の IPv4 アドレスからウェブサーバーへのインバウンド HTTPS アクセスを許可する。
ホームネットワークのパブリック IPv4 アドレスの範囲	TCP	22	ホームネットワークから Linux インスタンスへのインバウンド SSH アクセス (インターネットゲートウェイ経由) を許可します。ローカルコンピュータのパブリック IPv4 アドレスは、http://checkip.amazonaws.com または https://checkip.amazonaws.com などのサービスを使用して取得できます。ISP 経由で、またはファイアウォールの内側から静的な IP アドレスなしで接続している場合は、クライアントコンピュータで使用されている IP アドレスの範囲を見つける必要があります。
ホームネットワークのパブリック IPv4 アドレスの範囲	TCP	3389	ホームネットワークから Windows インスタンスへのインバウンド RDP アクセス (インターネッ

			トゲートウェイ経由) を許可します。
アウトバウンド			
送信先	プロトコル	ポート範囲	コメント
DBServerSG セキュリティグループの ID	TCP	1433	DBServerSG セキュリティグループに割り当てられたデータベースサーバーへのアウトバウンド Microsoft SQL Server アクセスを許可する。
DBServerSG セキュリティグループの ID	TCP	3306	DBServerSG セキュリティグループに割り当てられたデータベースサーバーへのアウトバウンド MySQL アクセスを許可する。
0.0.0.0/0	TCP	80	任意の IPv4 アドレスへのアウトバウンド HTTP アクセスを許可します。
0.0.0.0/0	TCP	443	任意の IPv4 アドレスへのアウトバウンド HTTPS アクセスを許可します。

次の表では、DBServerSG セキュリティグループの推奨ルールについて説明します。このルールにより、ウェブサーバーからの読み込みおよび書き込みデータベースリクエストが許可されます。また、データベースサーバーはインターネットへのトラフィックを開始することもできます (そのトラフィックは、ルートテーブルから NAT ゲートウェイに送信され、さらに NAT ゲートウェイからインターネットゲートウェイを介してインターネットに転送されます)。

DBServerSG: 推奨ルール

インバウンド			
送信元	プロトコル	ポート範囲	コメント
WebServerSG セキュリティグループの ID	TCP	1433	WebServerSG セキュリティグループに関連付けられたウェブサーバーからのインバウンド Microsoft SQL Server アクセスを許可する。
WebServerSG セキュリティグループの ID	TCP	3306	WebServerSG セキュリティグループに関連付けられたウェブサーバーからのインバウンド MySQL Server アクセスを許可する。
アウトバウンド			
送信先	プロトコル	ポート範囲	コメント

0.0.0.0/0	TCP	80	IPv4 経由でインターネットへのアウトバウンド HTTP アクセス (例: ソフトウェアのアップデート用) を許可します。
0.0.0.0/0	TCP	443	IPv4 経由でインターネットへのアウトバウンド HTTPS アクセス (例: ソフトウェアのアップデート用) を許可します。

(オプション) VPC のデフォルトのセキュリティグループには、割り当てられたインスタンス間で相互に通信することを自動的に許可するルールがあります。そのような通信をカスタムセキュリティグループに許可するには、以下のルールを追加する必要があります。

インバウンド			
送信元	プロトコル	ポート範囲	コメント
セキュリティグループの ID	すべて	すべて	このセキュリティグループに割り当てられた他のインスタンスからのインバウンドトラフィックを許可する。
アウトバウンド			
送信先	プロトコル	ポート範囲	コメント
The ID of the security group	All	All	Allow outbound traffic to other instances assigned to this security group.

(オプション) ホームネットワークからプライベートサブネットへの SSH または RDP トラフィックのプロキシとして使用する踏み台ホストをパブリックサブネットで起動する場合は、踏み台インスタンスまたはその関連セキュリティグループからのインバウンド SSH または RDP トラフィックを許可するルールを DBServerSG セキュリティグループに追加します。

IPv6 のセキュリティグループルール

IPv6 CIDR ブロックを VPC およびサブネットと関連付ける場合は、別のルールを WebServerSG および DBServerSG セキュリティグループに追加して、インバウンドおよびアウトバウンドの IPv6 トラフィックを制御する必要があります。このシナリオでは、ウェブサーバーは、IPv6 経由ですべてのインターネットトラフィックを受信したり、IPv6 経由でローカルネットワークから SSH または RDP トラフィックを受信したりできます。また、インターネットへのアウトバウンド IPv6 トラフィックを開始することもできます。データベースサーバーは、インターネットへのアウトバウンド IPv6 トラフィックを開始できます。

WebServerSG セキュリティグループの IPv6 固有ルールを次に示します (上記のルールは含まない)。

インバウンド

送信元	プロトコル	ポート範囲	コメント
::/0	TCP	80	任意の IPv6 アドレスからウェブサーバーへのインバウンド HTTP アクセスを許可する。
::/0	TCP	443	任意の IPv6 アドレスからウェブサーバーへのインバウンド HTTPS アクセスを許可する。
ネットワークの IPv6 アドレス範囲	TCP	22	(Linux インスタンス) ネットワークからの IPv6 経由のインバウンド SSH アクセスを許可する。
ネットワークの IPv6 アドレス範囲	TCP	3389	(Windows インスタンス) ネットワークからの IPv6 経由のインバウンド RDP アクセスを許可する
アウトバウンド			
送信先	プロトコル	ポート範囲	コメント
::/0	TCP	HTTP	Allow outbound HTTP access to any IPv6 address.
::/0	TCP	HTTPS	Allow outbound HTTPS access to any IPv6 address.

DBServerSG セキュリティグループの IPv6 固有ルールを次に示します (上記のルールは含まない)。

アウトバウンド			
送信先	プロトコル	ポート範囲	コメント
::/0	TCP	80	任意の IPv6 アドレスへのアウトバウンド HTTP アクセスを許可します
::/0	TCP	443	任意の IPv6 アドレスへのアウトバウンド HTTPS アクセスを許可します

シナリオ 2 を実装する

VPC ウィザードを使用して、VPC や、サブネット、NAT ゲートウェイ、さらにはオプションで、Egress-Only インターネットゲートウェイを作成できます。NAT ゲートウェイに Elastic IP アドレスを指定します。アドレスがない場合は、それを最初にアカウントに割り当てる必要があります。既存の Elastic IP アドレスを使用する場合は、そのアドレスが別のインスタンスやネットワークインターフェイスに現在関連

付けられていないことを確認します。NAT ゲートウェイは、VPC のパブリックサブネットで自動的に作成されます。

パブリックサブネットとプライベートサブネット (NAT) を持つ VPC の推奨ネットワーク ACL ルール

このシナリオでは、パブリックサブネット用のネットワーク ACL と、プライベートサブネット用の別のネットワーク ACL があります。次の表に、各 ACL に推奨されるルールを示します。これらのルールでは、明示的に必要なトラフィックを除き、すべてのトラフィックをブロックします。これらのルールは、このシナリオのセキュリティグループルールとほとんど同じです。

パブリックサブネット用の ACL ルール

Inbound					
ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント
100	0.0.0.0/0	TCP	80	許可	任意の IPv4 アドレスからのインバウンド HTTP トラフィックを許可します。
110	0.0.0.0/0	TCP	443	許可	任意の IPv4 アドレスからのインバウンド HTTPS トラフィックを許可します。
120	ホームネットワークのパブリック IP アドレスの範囲	TCP	22	許可	(インターネットゲートウェイを介した) ホームネットワークからのインバウンド SSH トラフィックを許可します。
130	ホームネットワークのパブリック IP アドレスの範囲	TCP	3389	許可	(インターネットゲートウェイを介した) ホームネットワークからのインバウンド RDP トラフィックを許可します。
140	0.0.0.0/0	TCP	1024-65535	許可	リクエストに応答しているか、送信元がサブネットである、インターネット上のホストからのインバウンドリターントラ

					フィックを許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv4 トラフィックを拒否します (変更不可)。
Outbound					
ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント
100	0.0.0.0/0	TCP	80	許可	サブネットからインターネットへのアウトバウンド HTTP トラフィックを許可します.
110	0.0.0.0/0	TCP	443	許可	サブネットからインターネットへのアウトバウンド HTTPS トラフィックを許可します.

120	10.0.1.0/24	TCP	1433	許可	プライベートサブネット内のデータベースサーバーに対するアウトバウンド MS SQL アクセスを許可します. このポート番号は一例に過ぎません。別の例として、MySQL/Aurora アクセス用の 3306、PostgreSQL アクセス用の 5432、Amazon Redshift アクセス用の 5439、Oracle アクセス用の 1521 などがあります。
140	0.0.0.0/0	TCP	32768-65535	許可	インターネット上のクライアントに対するアウトバウンド応答を許可します (たとえば、サブネット内のウェブサーバーを訪問するユーザーに対するウェブページの提供など) 。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。

150	10.0.1.0/24	TCP	22	許可	(SSH 拠点から) プライベートサブネットのインスタンスへのアウトバウンド SSH アクセスを許可します (該当する場合)。
*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのアウトバウンド IPv4 トラフィックを拒否します (変更不可)。

プライベートサブネット用の ACL ルール

Inbound					
ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント
100	10.0.0.0/24	TCP	1433	許可	パブリックサブネット内のウェブサーバーから、プライベートサブネット内の MS SQL サーバーに対する読み取りと書き込みを許可します. このポート番号は一例に過ぎません。別の例として、MySQL/Aurora アクセス用の 3306、PostgreSQL アクセス用の 5432、Amazon Redshift アクセス用の 5439、Oracle アクセス用の 1521 などがあります。
120	10.0.0.0/24	TCP	22	許可	SSH 拠点からのインバウンド SSH トラフィックを許可

					します (該当する場合)。
130	10.0.0.0/24	TCP	3389	許可	パブリックサブネット内の Microsoft Terminal Services ゲートウェイからのインバウンド TDP トラフィックを許可します.
140	0.0.0.0/0	TCP	1024-65535	許可	送信元がプライベートサブネットであるリクエストについて、パブリックサブネットの NAT デバイスからのインバウンドリターントラフィックを許可します。 正しい一時ポートの選択の詳細については、このトピックの冒頭にある注意点を参照してください。
*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv4 トラフィックを拒否します (変更不可)。
Outbound					
ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント
100	0.0.0.0/0	TCP	80	許可	サブネットからインターネットへのアウトバウンド HTTP トラフィックを許可します.

110	0.0.0.0/0	TCP	443	許可	サブネットからインターネットへのアウトバウンド HTTPS トラフィックを許可します.
120	10.0.0.0/24	TCP	32768-65535	許可	パブリックサブネットに対するアウトバウンド応答 (例: プライベートサブネット内の DB サーバーと通信するパブリックサブネット内のウェブサーバーに対する応答) を許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのアウトバウンド IPv4 トラフィックを拒否します (変更不可)。

IPv6 に推奨されるネットワーク ACL ルール

IPv6 サポートを実装し、関連付けられた IPv6 CIDR ブロックを持つ VPC とサブネットを作成した場合は、別のルールをネットワーク ACL に追加して、インバウンドおよびアウトバウンド IPv6 トラフィックを制御する必要があります。

ネットワーク ACL の IPv6 固有のルールを以下に示します (上記のルールは含みません) 。

パブリックサブネット用の ACL ルール

Inbound					
ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント
150	::/0	TCP	80	許可	任意の IPv6 アドレスからのインバウン

					ド HTTP トラフィックを許可します。
160	::/0	TCP	443	許可	任意の IPv6 アドレスからのインバウンド HTTPS トラフィックを許可します。
170	ホームネットワークの IPv6 アドレス範囲	TCP	22	許可	(インターネットゲートウェイを介した) ホームネットワークからの IPv6 経由のインバウンド SSH トラフィックを許可します。
180	ホームネットワークの IPv6 アドレス範囲	TCP	3389	許可	(インターネットゲートウェイを介した) ホームネットワークからの IPv6 経由のインバウンド RDP トラフィックを許可します。
190	::/0	TCP	1024-65535	許可	リクエストに応答しているか、送信元がサブネットである、インターネット上のホストからのインバウンドリターントラフィックを許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。

*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv6 トラフィックを拒否します (変更不可)。

Outbound

ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント
160	::/0	TCP	80	許可	サブネットからインターネットへのアウトバウンド HTTP トラフィックを許可します.
170	::/0	TCP	443	許可	サブネットからインターネットへのアウトバウンド HTTPS トラフィックを許可します
180	2001:db8:1234:1a00::/64	TCP	1433	許可	プライベートサブネット内のデータベースサーバーに対するアウトバウンド MS SQL アクセスを許可します. このポート番号は一例に過ぎません。別の例として、MySQL/Aurora アクセス用の 3306、PostgreSQL アクセス用の 5432、Amazon Redshift アクセス用の 5439、Oracle アクセス用の 1521 などがあります。

200	::/0	TCP	32768-65535	許可	インターネット上のクライアントに対するアウトバウンド応答を許可します (たとえば、サブネット内のウェブサーバーを訪問するユーザーに対するウェブページの提供など) 。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
210	2001:db8:1234:1a00::/64	TCP	22	許可	(SSH 拠点から) プライベートサブネットのインスタンスへのアウトバウンド SSH アクセスを許可します (該当する場合)。
*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのアウトバウンド IPv6 トラフィックを拒否します (変更不可)。

プライベートサブネット用の ACL ルール

Inbound					
ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント
150	2001:db8:1234:1a00::/64	TCP	1433	許可	パブリックサブネット内のウェブサーバーから、プライベートサブネット内の MS SQL サーバーに対する読み取りと書

					き込みを許可します. このポート番号は一例に過ぎません。別の例として、MySQL/Aurora アクセス用の 3306、PostgreSQL アクセス用の 5432、Amazon Redshift アクセス用の 5439、Oracle アクセス用の 1521 などがあります。
170	2001:db8:1234:1a00::/64	TCP	22	許可	パブリックサブネット内の SSH 拠点からのインバウンド SSH トラフィックを許可します (該当する場合)。
180	2001:db8:1234:1a00::/64	TCP	3389	許可	パブリックサブネット内の Microsoft Terminal Services ゲートウェイからのインバウンド RDP トラフィックを許可します (該当する場合)。

190	::/0	TCP	1024-65535	許可	送信元がプライベートサブネットであるリクエストについて、Egress-Only インターネットゲートウェイからのインバウンドリターントラフィックを許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv6 トラフィックを拒否します (変更不可)。

Outbound

ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント
130	::/0	TCP	80	許可	サブネットからインターネットへのアウトバウンド HTTP トラフィックを許可します.
140	::/0	TCP	443	許可	サブネットからインターネットへのアウトバウンド HTTPS トラフィックを許可します.

150	2001:db8:1234:1a00::/64	TCP	32768-65535	許可	パブリックサブネットに対するアウトバウンド応答 (例: プライベートサブネット内の DB サーバーと通信するパブリックサブネット内のウェブサーバーに対する応答) を許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのアウトバウンド IPv6 トラフィックを拒否します (変更不可)。

パブリックサブネットとプライベートサブネット、および AWS Site-to-Site VPN アクセスを持つ VPC

このシナリオの設定には、パブリックサブネットとプライベートサブネットを持つ Virtual Private Cloud (VPC)、および IPsec VPN トンネルを介した独自のネットワークとの通信を有効にする仮想プライベートゲートウェイが含まれます。このシナリオは、ネットワークをクラウドに拡張し、さらに、VPC からインターネットに直接アクセスする必要がある場合にお勧めします。このシナリオを使用すると、スケーラブルなウェブフロントエンドを持つ多階層のアプリケーションをパブリックサブネットで実行し、IPsec AWS Site-to-Site VPN 接続で自ネットワークに接続されているプライベートサブネット内にデータを保存できます。

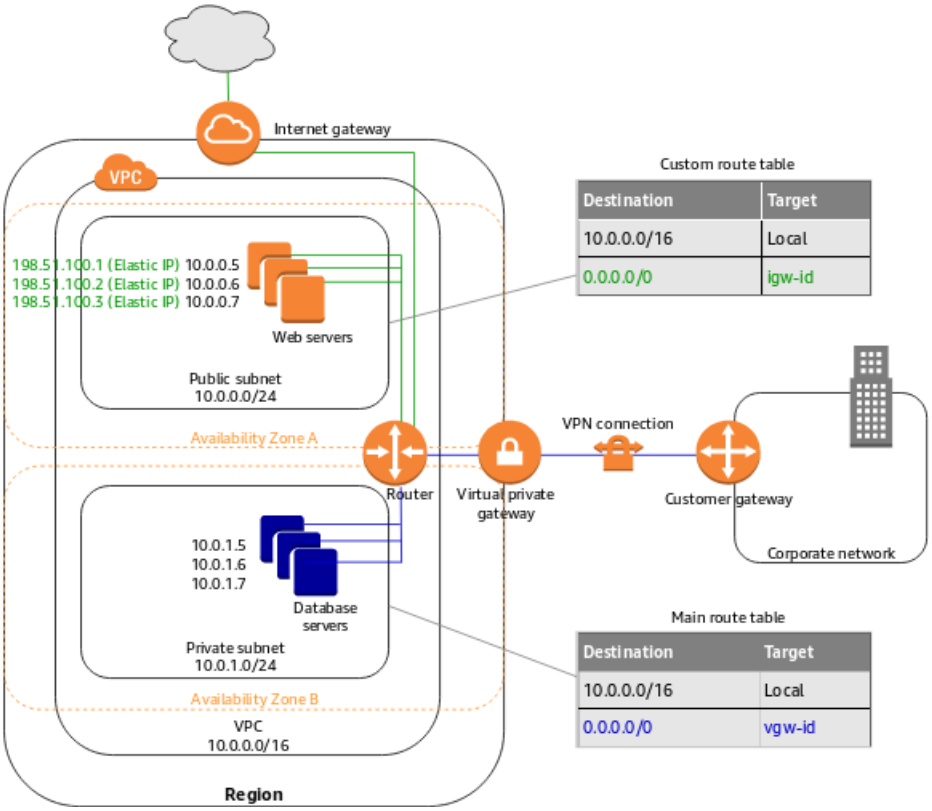
このシナリオは、オプションで IPv6 に設定することもできます。VPC ウィザードを使用して、関連する IPv6 CIDR ブロックで VPC およびサブネットを作成できます。サブネット内に起動されたインスタンスは、IPv6 アドレスを取得できます。仮想プライベートゲートウェイでは、Site-to-Site VPN 接続の IPv6 通信はサポートされていません。ただし、VPC 内のインスタンスは IPv6 で互いに通信でき、パブリックサブネット内のインスタンスは IPv6 でインターネット経由で通信できます。IPv4 アドレスと IPv6 アドレスの詳細については、「VPC の IP アドレス指定 (p. 124)」を参照してください。

EC2 インスタンスソフトウェアの管理については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Linux インスタンスでソフトウェアを管理する」を参照してください。

目次

Overview

次の図は、このシナリオの設定に重要なコンポーネントを示しています。



Important

このシナリオで、Site-to-Site VPN 接続の側でカスタマーゲートウェイデバイスを設定する方法については、AWS Site-to-Site VPN ユーザーガイドの「カスタマーゲートウェイデバイス」を参照してください。

このシナリオの設定には、以下の項目が含まれています。

- IPv4 CIDR ブロックサイズが /16 (例: 10.0.0.0/16) の Virtual Private Cloud (VPC)。65,536 個のプライベート IPv4 アドレスを提供します。
- IPv4 CIDR ブロックサイズが /24 (例: 10.0.0.0/24) のパブリックサブネット。256 個のプライベート IPv4 アドレスを提供します。パブリックサブネットは、インターネットゲートウェイへのルートが含まれているルートテーブルに関連付けられているサブネットです。
- IPv4 CIDR ブロックサイズが /24 (例: 10.0.1.0/24) の VPN 専用サブネット。256 個のプライベート IPv4 アドレスを提供します。
- インターネットゲートウェイ。VPC をインターネットおよび他の AWS 製品に接続します。

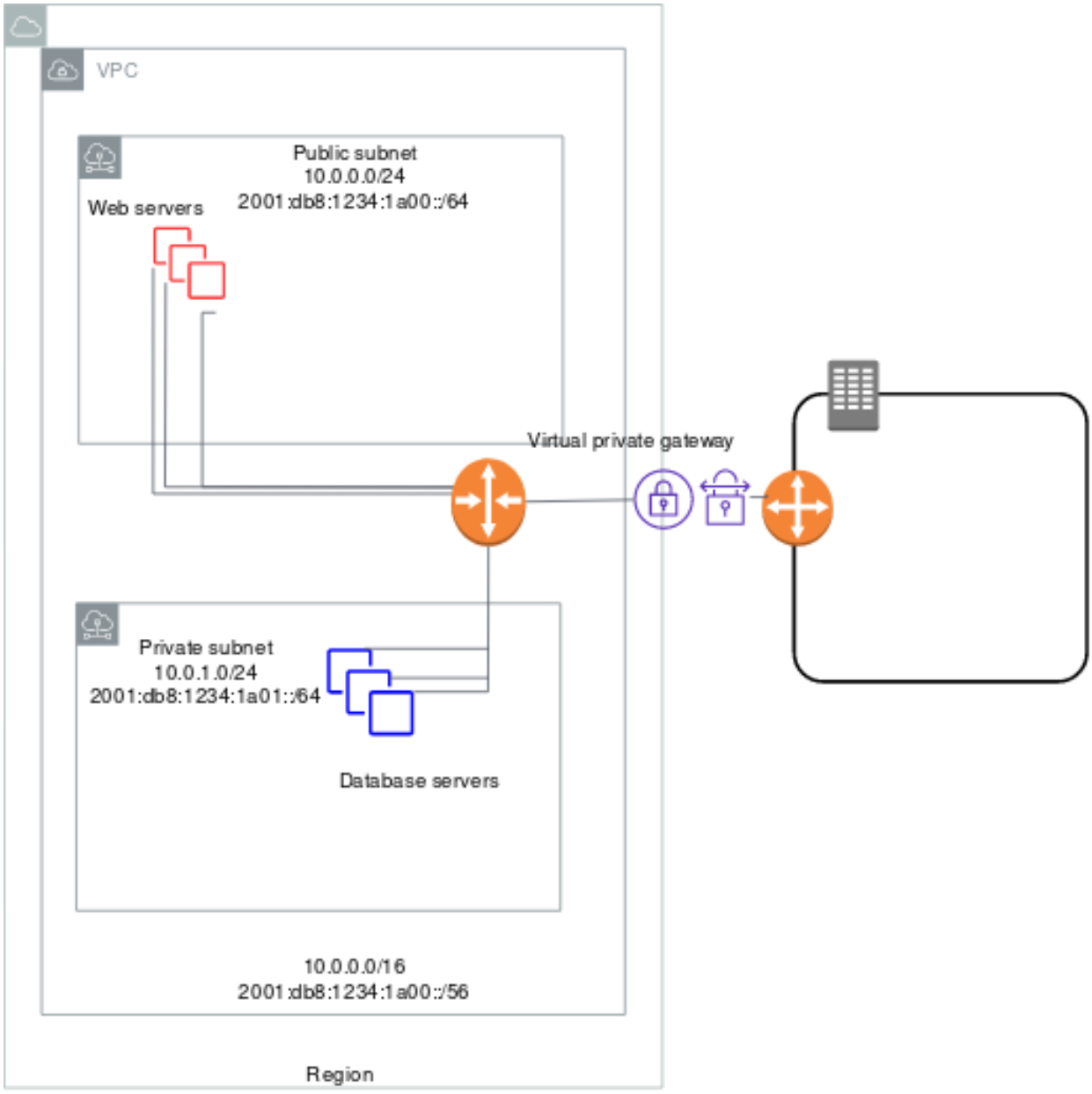
- VPC とネットワークの間の Site-to-Site VPN 接続。この Site-to-Site VPN 接続は、仮想プライベートゲートウェイとカスタマーゲートウェイで構成され、前者は Site-to-Site VPN 接続の Amazon 側、後者は Site-to-Site VPN 接続のお客様側に配置されています。
- プライベート IPv4 アドレスがサブネットの範囲内 (例: 10.0.0.5 と 10.0.1.5) であるインスタンス。インスタンスが VPC 内で相互に、かつ他のインスタンスと通信できるようにします。
- Elastic IP アドレス (例: 198.51.100.1) を持つパブリックサブネットのインスタンス。Elastic IP アドレスは、インターネットからインスタンスにアクセスできるようにするパブリック IPv4 アドレスです。インスタンスには、起動時に Elastic IP アドレスではなくパブリック IPv4 アドレスを割り当てることができます。VPN のみのサブネットのインスタンスは、インターネットからの受信トラフィックを受け取る必要がないバックエンドサーバーです。ただし、ネットワークからのトラフィックを送受信できます。
- パブリックサブネットに関連付けられているカスタムルートテーブル。このルートテーブルには、サブネットのインスタンスが VPC 内の他のインスタンスと通信できるようにするエントリと、サブネットのインスタンスがインターネットと直接通信できるようにするエントリが含まれます。
- VPN のみのサブネットに関連付けられているメインルートテーブル。ルートテーブルには、サブネットのインスタンスが VPC 内の他のインスタンスと通信できるようにするエントリと、サブネットのインスタンスがネットワークと直接通信できるようにするエントリが含まれます。

サブネットの詳細については、「VPC とサブネット (p. 106)」および「VPC の IP アドレス指定 (p. 124)」を参照してください。インターネットゲートウェイの詳細については、「インターネットゲートウェイ (p. 221)」を参照してください。AWS Site-to-Site VPN 接続の詳細については、AWS Site-to-Site VPN ユーザーガイドの「AWS Site-to-Site VPN とは」を参照してください。

IPv6 の概要

オプションでこのシナリオの IPv6 を有効にできます。上記のコンポーネントに加えて、この設定には、次に示す情報が含まれます。

- VPC に関連付けられたサイズ /56 の IPv6 CIDR ブロック (例: 2001:db8:1234:1a00::/56)。AWS の CIDR は自動的に割り当てられます。独自にアドレス範囲を選択することはできません。
- パブリックサブネットに関連付けられたサイズ /64 の IPv6 CIDR ブロック (例: 2001:db8:1234:1a00::/64)。VPC に割り当てられた範囲からサブネットの範囲を選択できます。IPv6 CIDR のサイズを選択することはできません。
- VPN 専用サブネットに関連付けられたサイズ /64 の IPv6 CIDR ブロック (例: 2001:db8:1234:1a01::/64)。VPC に割り当てられた範囲からサブネットの範囲を選択できます。IPv6 CIDR のサイズを選択することはできません。
- サブネットの範囲からのインスタンスに割り当てられていた IPv6 アドレス (例: 2001:db8:1234:1a00::1a)。
- カスタムエントリテーブル内のルートテーブルエントリ。パブリックサブネットのインスタンスが IPv6 を使用して相互通信したり、インターネット経由で直接通信したりできるようにします。
- メインルートテーブルのルートテーブル エントリ。VPN のみのサブネットのインスタンスが、IPv6 を使用して相互に通信できるようにします。



パブリックサブネット内のウェブサーバーには、次のアドレスがあります。

サーバー	IPv4 アドレス	Elastic IP アドレス	IPv6 アドレス
1	10.0.0.5	198.51.100.1	2001:db8:1234:1a00::1a
2	10.0.0.6	198.51.100.2	2001:db8:1234:1a00::2b
3	10.0.0.7	198.51.100.3	2001:db8:1234:1a00::3c

プライベートサブネット内のデータベースサーバーには、次のアドレスがあります。

サーバー	IPv4 アドレス	IPv6 アドレス
1	10.0.1.5	2001:db8:1234:1a01::1a
2	10.0.1.6	2001:db8:1234:1a01::2b
3	10.0.1.7	2001:db8:1234:1a01::3c

Routing

VPC には暗示的なルーターがあります (このシナリオの設定図を参照)。このシナリオでは、VPC ウィザードによって、VPN のみのサブネットで使用されるメインルートテーブルを更新し、カスタムルートテーブルを作成してパブリックサブネットに関連付けます。

VPN のみのサブネットのインスタンスはインターネットに直接接続することはできません。インターネット宛てのトラフィックはすべて、まず仮想プライベートゲートウェイを経由してネットワークに向かいます。そこでは、ファイアウォールと企業のセキュリティポリシーが適用されます。インスタンスが AWS 宛てのトラフィック (Amazon S3 や Amazon EC2 API へのリクエストなど) を送信すると、リクエストは仮想プライベートゲートウェイを介してネットワークに向かい、その後インターネットに進み、AWS に到達します。

> Tip
>
> ネットワークからのトラフィックで、パブリックサブネットのインスタンスの Elastic IP アドレスに向かうものはすべて、仮想プライベートゲートウェイではなく、インターネットを経由します。代わりに、ネットワークからのトラフィックでパブリックサブネットに向かうものが仮想プライベートゲートウェイを経由できるように、ルートとセキュリティグループのルールを設定することができます。

Site-to-Site VPN 接続は、静的にルーティングされた Site-to-Site VPN 接続、または動的にルーティングされた Site-to-Site VPN 接続 (BGP を使用) として設定されます。静的なルーティングを選択すると、Site-to-Site VPN 接続を作成するときに、ネットワークの IP プレフィックスを手動で入力するように求められます。動的なルーティングを選択すると、IP プレフィックスは、BGP を使用して VPC の仮想プライベートゲートウェイに自動的にアドバタイズされます。

以下の表は、このシナリオのルートテーブルを示しています。

メインルートテーブル

最初のエントリは、VPC のローカルルーティングのデフォルトエントリです。このエントリによって、VPC 内のインスタンスが IPv4 を使用して相互に通信できるようになります。2 番目のエントリは、仮想プライベートゲートウェイ (例: `vgw-1a2b3c4d`) を介してプライベートサブネットからご利用のネットワークに他のすべての IPv4 サブネットトラフィックをルーティングします。

送信先	ターゲット
`10.0.0.0/16`	ローカル
`0.0.0.0/0`	vgw-id

カスタムルートテーブル

最初のエントリは、VPC のローカルルーティングのデフォルトエントリです。このエントリによって、VPC 内のインスタンスが相互に通信できるようになります。2 番目のエントリは、インターネット

ゲートウェイ (例: igw-1a2b3c4d) を介してパブリックサブネットからインターネットに他のすべての IPv4 サブネットトラフィックをルーティングします。

送信先	ターゲット
10.0.0.0/16	ローカル
0.0.0.0/0	igw-id

代替ルーティング

プライベートサブネットのインスタンスからインターネットにアクセスする場合、パブリックサブネットでネットワークアドレス変換 (NAT) ゲートウェイまたはインスタンスを作成し、サブネットのインターネット宛てのトラフィックが NAT デバイスに向かうようにルーティングを設定することもできます。これにより、VPN のみのサブネットのインスタンスから、インターネットゲートウェイ経由でリクエストを送信できるようになります (例: ソフトウェアをアップデートする場合)。

NAT デバイスの手動設定の詳細については、「VPC の NAT デバイス (p. 236)」を参照してください。VPC ウィザードを使用した NAT デバイスの設定については、「パブリックサブネットとプライベートサブネットを持つ VPC (NAT) (p. 31)」を参照してください。

プライベートサブネットのインターネット宛てのトラフィックが NAT デバイスにアクセスできるようにするには、メインルートテーブルを次のように更新する必要があります。

最初のエントリは、VPC のローカルルーティングのデフォルトエントリです。2 番目のエントリは、独自のローカル (カスタマー) ネットワークへのサブネットトラフィックを、仮想プライベートゲートウェイにルーティングします。この例では、ローカルネットワークの IP アドレス範囲が 172.16.0.0/12 であると仮定しています。3 番目のエントリは、他のすべてのサブネットトラフィックを NAT ゲートウェイに送信します。

送信先	ターゲット
10.0.0.0/16	ローカル
172.16.0.0/12	vgw-id
0.0.0.0/0	nat-gateway-id

IPv6 のルーティング

IPv6 CIDR ブロックを VPC およびサブネットと関連付ける場合は、IPv6 トラフィックの別のルートをルートテーブルに含める必要があります。VPC 内の IPv6 通信を有効化した場合のシナリオのルートテーブルを次の表に示します。

メインルートテーブル

2 番目のエントリは、IPv6 経由で VPC 内のローカルルーティングに自動的に追加されたデフォルトルートです。

送信先	ターゲット
10.0.0.0/16	ローカル
2001:db8:1234:1a00::/56	ローカル

送信先	ターゲット
0.0.0.0/0	vgw-id

カスタムルートテーブル

2 番目のエントリは、IPv6 経由で VPC 内のローカルルーティングに自動的に追加されたデフォルトルートです。4 番目のエントリは、他のすべての IPv6 サブネットトラフィックをインターネットゲートウェイにルーティングします。

送信先	ターゲット
10.0.0.0/16	ローカル
2001:db8:1234:1a00::/56	ローカル
0.0.0.0/0	igw-id
::/0	igw-id

Security

AWS では、セキュリティグループとネットワーク ACL という 2 つの機能を使用して、VPC のセキュリティを強化できます。セキュリティグループは、インスタンスのインバウンドトラフィックとアウトバウンドトラフィックをコントロールします。ネットワーク ACL は、サブネットのインバウンドトラフィックとアウトバウンドトラフィックをコントロールします。通常、セキュリティグループで用が足りますが、VPC に追加のセキュリティレイヤーが必要な場合は、ネットワーク ACL を使用することもできます。詳細については、「」を参照してくださいAmazon VPC でのインターネットワークトラフィックのプライバシー (p. 165)

シナリオ 3 では、セキュリティグループを使用しますが、ネットワーク ACL は使用しません。ネットワーク ACL を使用する場合は、「パブリックサブネットとプライベートサブネットおよび AWS Site-to-Site VPN アクセスを持つ VPC の推奨ネットワーク ACL ルール (p. 62)」を参照してください。

VPC には、デフォルトのセキュリティグループ (p. 190)が用意されています。VPC 内に起動されるインスタンスは、起動時に別のセキュリティグループを指定しないと、デフォルトのセキュリティグループに自動的に関連付けられます。このシナリオでは、デフォルトのセキュリティグループを使用するのではなく、以下のセキュリティグループを作成することをお勧めします。

- WebServerSG: パブリックサブネットでウェブサーバーを起動するときに、このセキュリティグループを指定します。
- DBServerSG: VPN のみのサブネットでデータベースサーバーを起動するときに、このセキュリティグループを指定します。

セキュリティグループに割り当てられたインスタンスのサブネットは様々です。ただし、このシナリオでは、各セキュリティグループがインスタンスの役割の種類に対応しており、役割ごとにインスタンスが特定のサブネットに属さなければなりません。したがって、このシナリオでは、1 つのセキュリティグループに割り当てられたインスタンスはすべて、同じサブネットに属しています。

次の表では、WebServerSG セキュリティグループの推奨ルールについて説明します。このルールにより、ウェブサーバーがインターネットトラフィックを受信したり、ご利用のネットワークから SSH や RDP トラフィックを受信したりできます。また、ウェブサーバーは VPN のみのサブネットで、データベースサーバーへのリクエストの読み取り/書き込みを開始し、インターネットにトラフィックを送信できます (たとえば、ソフトウェアの更新プログラムを取得するなど)。ウェブサーバーがその他のアウトバウンド通信を開始しないため、デフォルトのアウトバウンドルールは削除されます。

Note

グループには SSH アクセスと RDP アクセスの両方、および Microsoft SQL Server アクセスと MySQL アクセスの両方が含まれます。この場合は、Linux (SSH および MySQL) または Windows (RDP および Microsoft SQL Server) に対するルールのみで十分かもしれません。

WebServerSG: 推奨ルール

インバウンド			
送信元	プロトコル	ポート範囲	コメント
0.0.0.0/0	TCP	80	任意の IPv4 アドレスからウェブサーバーへのインバウンド HTTP アクセスを許可する。
0.0.0.0/0	TCP	443	任意の IPv4 アドレスからウェブサーバーへのインバウンド HTTPS アクセスを許可する。
ネットワークのパブリック IP アドレス範囲	TCP	22	ネットワークから Linux インスタンスへのインバウンド SSH アクセス (インターネットゲートウェイ経由) を許可します。
ネットワークのパブリック IP アドレス範囲	TCP	3389	ネットワークから Windows インスタンスへのインバウンド RDP アクセス (インターネットゲートウェイ経由) を許可します。
アウトバウンド			
DBServerSG セキュリティグループの ID	TCP	1433	DBServerSG に割り当てられたデータベースサーバーへのアウトバウンド Microsoft SQL Server アクセスを許可する.
DBServerSG セキュリティグループの ID	TCP	3306	DBServerSG に割り当てられたデータベースサーバーへのアウトバウンド MySQL アクセスを許可する.
0.0.0.0/0	TCP	80	インターネットへのアウトバウンド HTTP アクセスを許可する。
0.0.0.0/0	TCP	443	インターネットへのアウトバウンド HTTPS アクセスを許可する。

次の表では、DBServerSG セキュリティグループの推奨ルールについて説明します。このルールにより、ウェブサーバーからの Microsoft SQL Server と MySQL の読み取りおよび書き込みリクエストと、ご利用のネットワークからの SSH および RDP トラフィックが許可されます。また、データベースサーバーは、インターネットへのトラフィックを開始することもできます (ルートテーブルは、そのトラフィックを仮想プライベートゲートウェイを介して送信します)。

DBServerSG: 推奨ルール

インバウンド			
送信元	プロトコル	ポート範囲	コメント
WebServerSG セキュリティグループの ID	TCP	1433	WebServerSG セキュリティグループに関連付けられたウェブサーバーからのインバウンド Microsoft SQL Server アクセスを許可する。
WebServerSG セキュリティグループの ID	TCP	3306	WebServerSG セキュリティグループに関連付けられたウェブサーバーからのインバウンド MySQL Server アクセスを許可する。
ネットワークの IPv4 アドレス範囲	TCP	22	ネットワークから Linux インスタンスへのインバウンド SSH トラフィック (仮想プライベートゲートウェイ経由) を許可します。
ネットワークの IPv4 アドレス範囲	TCP	3389	ネットワークから Windows インスタンスへのインバウンド RDP トラフィック (仮想プライベートゲートウェイ経由) を許可します。
アウトバウンド			
送信先	プロトコル	ポート範囲	コメント
0.0.0.0/0	TCP	80	仮想プライベートゲートウェイを介したインターネットへのアウトバウンド IPv4 HTTP アクセス (例: ソフトウェアアップデート) を許可する。
0.0.0.0/0	TCP	443	仮想プライベートゲートウェイを介したインターネットへのアウトバウンド IPv4 HTTPS アクセス (例: ソフトウェアアップデート) を許可する。

(オプション) VPC のデフォルトのセキュリティグループには、割り当てられたインスタンス間で相互に通信することを自動的に許可するルールがあります。そのような通信をカスタムセキュリティグループに許可するには、以下のルールを追加する必要があります。

インバウンド			
送信元	プロトコル	ポート範囲	コメント
セキュリティグループの ID	すべて	すべて	このセキュリティグループに割り当てられた他のインスタンスからのインバウンドトラフィックを許可する。
アウトバウンド			
送信先	プロトコル	ポート範囲	コメント
The ID of the security group	All	All	Allow outbound traffic to other instances assigned to this security group.

IPv6 のセキュリティグループルール

IPv6 CIDR ブロックを VPC およびサブネットと関連付ける場合は、別のルールを WebServerSG および DBServerSG セキュリティグループに追加して、インバウンドおよびアウトバウンドの IPv6 トラフィックを制御する必要があります。このシナリオでは、ウェブサーバーは、IPv6 経由ですべてのインターネットトラフィックを受信したり、IPv6 経由でローカルネットワークから SSH または RDP トラフィックを受信したりできます。また、インターネットへのアウトバウンド IPv6 トラフィックを開始することもできます。データベースサーバーは、アウトバウンド IPv6 トラフィックを開始することはできません。したがって、セキュリティグループルールを追加する必要はありません。

WebServerSG セキュリティグループの IPv6 固有ルールを次に示します (上記のルールは含まない)。

インバウンド			
送信元	プロトコル	ポート範囲	コメント
::/0	TCP	80	任意の IPv6 アドレスからウェブサーバーへのインバウンド HTTP アクセスを許可する。
::/0	TCP	443	任意の IPv6 アドレスからウェブサーバーへのインバウンド HTTPS アクセスを許可する。
ネットワークの IPv6 アドレス範囲	TCP	22	(Linux インスタンス) ネットワークからの IPv6 経由のインバウンド SSH アクセスを許可する。
ネットワークの IPv6 アドレス範囲	TCP	3389	(Windows インスタンス) ネットワークからの

			IPv6 経由のインバウンド RDP アクセスを許可する
アウトバウンド			
送信先	プロトコル	ポート範囲	コメント
::/0	TCP	HTTP	Allow outbound HTTP access to any IPv6 address.
::/0	TCP	HTTPS	Allow outbound HTTPS access to any IPv6 address.

シナリオ 3 を実装する

シナリオ 3 を実装するには、カスタマーゲートウェイに関する情報を取得し、VPC ウィザードを使用して VPC を作成します。VPC ウィザードでは、カスタマーゲートウェイと仮想プライベートゲートウェイを使用して Site-to-Site VPN 接続を作成します。

この手順には、VPC の IPv6 通信を有効化して設定するオプションのステップが含まれます。VPC 内で IPv6 を使用する場合は、上記のステップを実行する必要はありません。

カスタマーゲートウェイを準備するには

1. カスタマーゲートウェイデバイスとして使用するデバイスを決めます。詳細については、AWS Site-to-Site VPN ユーザーガイドの「カスタマーゲートウェイデバイス」を参照してください。
2. カスタマーゲートウェイデバイスの外部インターフェイスのインターネットルーティングが可能な IP アドレスを取得します。このアドレスは静的である必要があります。また、ネットワークアドレス変換 (NAT) を実行するデバイスの背後のアドレスを使用することができます。
3. 静的にルーティングされた Site-to-Site VPN 接続を作成する場合は、Site-to-Site VPN 接続を介して仮想プライベートゲートウェイにアドバタイズする内部 IP 範囲のリストを (CIDR 表記で) 取得します。詳細については、AWS Site-to-Site VPN ユーザーガイドの「ルートテーブルと VPN ルーティングの優先度」を参照してください。

IPv4 で VPC ウィザードを使用する方法については、「開始方法 (p. 11)」を参照してください。

IPv6 で VPC ウィザードを使用する方法については、「the section called “IPv6 の使用開始” (p. 15)」を参照してください。

パブリックサブネットとプライベートサブネットおよび AWS Site-to-Site VPN アクセスを持つ VPC の推奨ネットワーク ACL ルール

このシナリオでは、パブリックサブネット用のネットワーク ACL と、VPN 専用サブネット用の別のネットワーク ACL があります。次の表に、各 ACL に推奨されるルールを示します。これらのルールでは、明示的に必要なトラフィックを除き、すべてのトラフィックをブロックします。

パブリックサブネット用の ACL ルール

Inbound					
ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント

100	0.0.0.0/0	TCP	80	許可	任意の IPv4 アドレスからウェブサーバーへのインバウンド HTTP トラフィックを許可する。
110	0.0.0.0/0	TCP	443	許可	任意の IPv4 アドレスからウェブサーバーへのインバウンド HTTPS トラフィックを許可する。
120	ホームネットワークのパブリック IPv4 アドレスの範囲	TCP	22	許可	(インターネットゲートウェイを介した) ホームネットワークからウェブサーバーへのインバウンド SSH トラフィックを許可します。
130	ホームネットワークのパブリック IPv4 アドレスの範囲	TCP	3389	許可	(インターネットゲートウェイを介した) ホームネットワークからウェブサーバーへのインバウンド RDP トラフィックを許可します。
140	0.0.0.0/0	TCP	32768-65535	許可	リクエストに応答しているか、送信元がサブネットである、インターネット上のホストからのインバウンドリターントラフィックを許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。

*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv4 トラフィックを拒否します (変更不可)。

Outbound

ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント
100	0.0.0.0/0	TCP	80	許可	サブネットからインターネットへのアウトバウンド HTTP トラフィックを許可します.
110	0.0.0.0/0	TCP	443	許可	サブネットからインターネットへのアウトバウンド HTTPS トラフィックを許可します.
120	10.0.1.0/24	TCP	1433	許可	VPN のみのサブネット内のデータベースサーバーに対するアウトバウンド MS SQL アクセスを許可します. このポート番号は一例に過ぎません。別の例として、MySQL/Aurora アクセス用の 3306、PostgreSQL アクセス用の 5432、Amazon Redshift アクセス用の 5439、Oracle アクセス用の 1521 などがあります。

140	0.0.0.0/0	TCP	32768-65535	許可	インターネット上のクライアントに対するアウトバウンド IPv4 応答を許可します (例: サブネット内のウェブサーバーを訪問するユーザーに対するウェブページの提供) 。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのアウトバウンドトラフィックを拒否します (変更不可能)。

VPN のみのサブネットの ACL 設定

Inbound					
ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント
100	10.0.0.0/24	TCP	1433	許可	パブリックサブネット内のウェブサーバーから、VPN のみのサブネット内の MS SQL サーバーに対する読み取りと書き込みを許可します. このポート番号は一例に過ぎません。別の例として、MySQL/Aurora アクセス用の 3306、PostgreSQL アクセス用の

					5432、Amazon Redshift アクセス用の 5439、Oracle アクセス用の 1521 などがあります。
120	ホームネットワークのプライベート IPv4 アドレスの範囲	TCP	22	許可	(仮想プライベートゲートウェイを介した) ホームネットワークからのインバウンド SSH トラフィックを許可します。
130	ホームネットワークのプライベート IPv4 アドレスの範囲	TCP	3389	許可	(仮想プライベートゲートウェイを介した) ホームネットワークからのインバウンド RDP トラフィックを許可します。
140	ホームネットワークのプライベート IP アドレスの範囲	TCP	32768-65535	許可	(仮想プライベートゲートウェイを介した) ホームネットワークのクライアントからのインバウンドリターントラフィックを許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンドトラフィックを拒否します (変更不可能)。

Outbound

ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント
100	ホームネットワークのプライベート IP アドレスの範囲	すべて	すべて	許可	サブネットからホームネットワークへのすべてのアウトバウンドトラフィック (仮想プライベートゲートウェイ経由) を許可します。このルールはルール 120 も含んでいます。ただし、特定のプロトコルタイプとポート番号を使用して、このルールの適用範囲を絞り込むことができます。このルールの適用範囲を絞り込む場合、アウトバウンド応答がブロックされないようにするには、ネットワーク ACL にルール 120 を含める必要があります。
110	10.0.0.0/24	TCP	32768-65535	許可	パブリックサブネットのウェブサーバーに対するアウトバウンド応答を許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。

120	ホームネットワークのプライベート IP アドレスの範囲	TCP	32768-65535	許可	ホームネットワーク内のクライアントへのアウトバウンド応答 (仮想プライベートゲートウェイ経由) を許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのアウトバウンドトラフィックを拒否します (変更不可能)。

IPv6 に推奨されるネットワーク ACL ルール

IPv6 サポートを実装し、関連付けられた IPv6 CIDR ブロックを持つ VPC とサブネットを作成した場合は、別のルールをネットワーク ACL に追加して、インバウンドおよびアウトバウンド IPv6 トラフィックを制御する必要があります。

ネットワーク ACL の IPv6 固有のルールを以下に示します (上記のルールは含みません) 。

パブリックサブネット用の ACL ルール

Inbound					
ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント
150	::/0	TCP	80	許可	任意の IPv6 アドレスからのインバウンド HTTP トラフィックを許可します。
160	::/0	TCP	443	許可	任意の IPv6 アドレスからのインバウンド HTTPS トラフィックを許可します。

170	ホームネットワークの IPv6 アドレス範囲	TCP	22	許可	(インターネットゲートウェイを介した) ホームネットワークからの IPv6 経由のインバウンド SSH トラフィックを許可します。
180	ホームネットワークの IPv6 アドレス範囲	TCP	3389	許可	(インターネットゲートウェイを介した) ホームネットワークからの IPv6 経由のインバウンド RDP トラフィックを許可します。
190	::/0	TCP	1024-65535	許可	リクエストに応答しているか、送信元がサブネットである、インターネット上のホストからのインバウンドリターントラフィックを許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv6 トラフィックを拒否します (変更不可)。
Outbound					
ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント

150	::/0	TCP	80	許可	サブネットからインターネットへのアウトバウンド HTTP トラフィックを許可します.
160	::/0	TCP	443	許可	サブネットからインターネットへのアウトバウンド HTTPS トラフィックを許可します.
170	2001:db8:1234:1a00::/64	TCP	1433	許可	プライベートサブネット内のデータベースサーバーに対するアウトバウンド MS SQL アクセスを許可します. このポート番号は一例に過ぎません。別の例として、MySQL/Aurora アクセス用の 3306、PostgreSQL アクセス用の 5432、Amazon Redshift アクセス用の 5439、Oracle アクセス用の 1521 などがあります。

190	::/0	TCP	32768-65535	許可	インターネット上のクライアントに対するアウトバウンド応答を許可します (たとえば、サブネット内のウェブサーバーを訪問するユーザーに対するウェブページの提供など) 。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのアウトバウンド IPv6 トラフィックを拒否します (変更不可)。

VPN 専用サブネットの ACL ルール

Inbound					
ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント
150	2001:db8:1234:1a00::/64	TCP	1433	許可	パブリックサブネット内のウェブサーバーから、プライベートサブネット内の MS SQL サーバーに対する読み取りと書き込みを許可します. このポート番号は一例に過ぎません。別の例として、MySQL/Aurora アクセス用の 3306、PostgreSQL

					アクセス用の 5432、Amazon Redshift アクセス用の 5439、Oracle アクセス用の 1521 などがあります。
*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv6 トラフィックを拒否します (変更不可)。

Outbound

ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント
130	2001:db8:1234:1a00::/64	TCP	32768-65535	許可	パブリックサブネットに対するアウトバウンド応答 (例: プライベートサブネット内の DB サーバーと通信するパブリックサブネット内のウェブサーバーに対する応答) を許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのアウトバウンド IPv6 トラフィックを拒否します (変更不可)。

プライベートサブネットのみおよび AWS Site-to-Site VPN アクセスを持つ VPC

このシナリオの設定には、1 つのプライベートサブネットを持つ Virtual Private Cloud (VPC)、および IPsec VPN トンネルを介した独自のネットワークとの通信を有効にする仮想プライベートゲートウェイが含まれます。インターネット経由の通信を有効にするインターネットゲートウェイはありません。このシナリオは、ネットワークをインターネットに公開せずに、Amazon のインフラストラクチャを使用してネットワークをクラウドに拡張する場合にお勧めします。

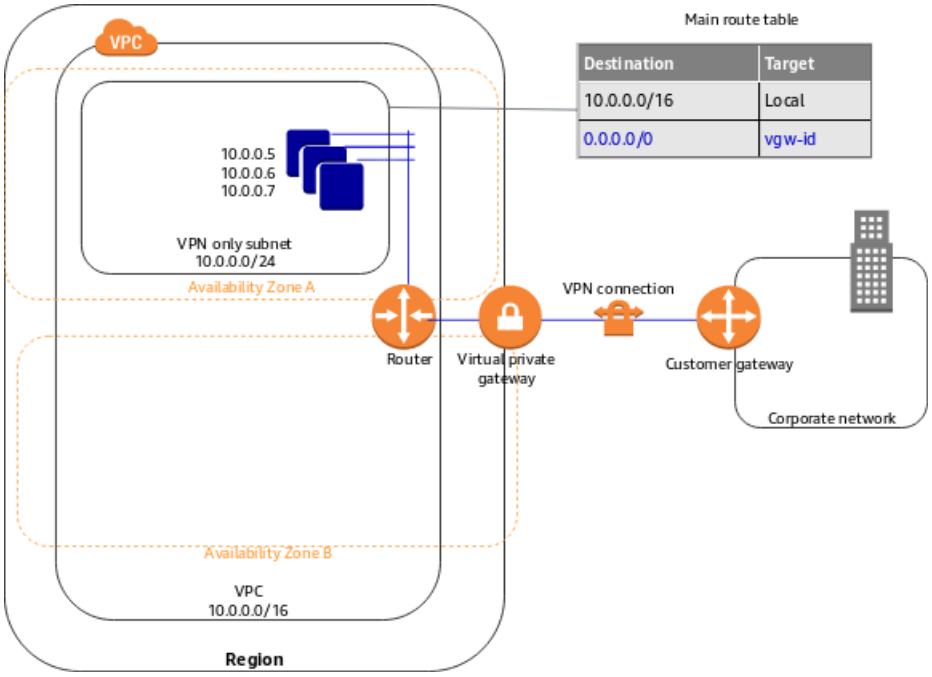
また、このシナリオは、オプションで IPv6 に設定することもできます。VPC ウィザードを使用して、関連する IPv6 CIDR ブロックで VPC およびサブネットを作成できます。サブネット内に起動されたインスタンスは、IPv6 アドレスを取得できます。仮想プライベートゲートウェイでは、AWS Site-to-Site VPN 接続の IPv6 通信はサポートされていません。ただし、VPC 内のインスタンスは IPv6 で互いに通信できます。IPv4 アドレスと IPv6 アドレスの詳細については、「VPC の IP アドレス指定 (p. 124)」を参照してください。

EC2 インスタンスソフトウェアの管理については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Linux インスタンスでソフトウェアを管理する」を参照してください。

目次

Overview

次の図は、このシナリオの設定に重要なコンポーネントを示しています。



Important

このシナリオにおいて、Site-to-Site VPN 接続側でカスタマーゲートウェイデバイスを設定するには、「カスタマーゲートウェイデバイス」を参照してください。

このシナリオの設定には、以下の項目が含まれています。

- CIDR ブロックサイズが /16 (例: 10.0.0.0/16) の Virtual Private Cloud (VPC)。65,536 個のプライベート IP アドレスを提供します。
- CIDR ブロックサイズが /24 (例: 10.0.0.0/24) の VPN のみのサブネット。256 個のプライベート IP アドレスを提供します。
- VPC とネットワークの間の Site-to-Site VPN 接続。この Site-to-Site VPN 接続は、仮想プライベートゲートウェイとカスタマーゲートウェイで構成され、前者は Site-to-Site VPN 接続の Amazon 側、後者は Site-to-Site VPN 接続のお客様側に配置されています。
- サブネット範囲のプライベート IP アドレス (例: 10.0.0.5、10.0.0.6、および 10.0.0.7) を持つインスタンス。そのインスタンスが VPC 内で相互に、および他のインスタンスと通信できるようにします。
- メインルートテーブルには、サブネットのインスタンスが VPC 内の他のインスタンスと通信できるようにするルートが含まれています。ルート伝播が有効なため、サブネット内のインスタンスがネットワークと直接通信できるルートは、メインルートテーブルに伝播されたルートとして表示されます。

サブネットの詳細については、「VPC とサブネット (p. 106)」および「VPC の IP アドレス指定 (p. 124)」を参照してください。Site-to-Site VPN 接続の詳細については、AWS Site-to-Site VPN ユーザーガイドの「AWS Site-to-Site VPN とは」を参照してください。カスタマーゲートウェイデバイスの設定の詳細については、「カスタマーゲートウェイデバイス」を参照してください。

IPv6 の概要

オプションでこのシナリオの IPv6 を有効にできます。上記のコンポーネントに加えて、この設定には、次に示す情報が含まれます。

- VPC に関連付けられたサイズ /56 の IPv6 CIDR ブロック (例: 2001:db8:1234:1a00::/56)。AWS の CIDR は自動的に割り当てられます。独自にアドレス範囲を選択することはできません。
- VPN 専用サブネットに関連付けられたサイズ /64 の IPv6 CIDR ブロック (例: 2001:db8:1234:1a00::/64)。VPC に割り当てられた範囲からサブネットの範囲を選択できます。IPv6 CIDR のサイズを選択することはできません。
- サブネットの範囲からのインスタンスに割り当てられていた IPv6 アドレス (例: 2001:db8:1234:1a00::1a)。
- メインルートテーブル内のルートテーブルエントリは、プライベートサブネットのインスタンスが IPv6 を使用して相互通信できるようにします。

Routing

VPC には暗示的なルーターがあります (このシナリオの設定図を参照)。このシナリオでは、VPC ウィザードによって、送信先が VPC 外のアドレスであるすべてのトラフィックを AWS Site-to-Site VPN 接続にルーティングするルートテーブルを作成し、そのルートテーブルをサブネットに関連付けます。

このシナリオのルートテーブルは次のとおりです。最初のエントリは、VPC のローカルルーティングのデフォルトエントリです。このエントリによって、この VPC 内のインスタンスが相互に通信できるようになります。2 番目のエントリは、他のすべてのサブネットトラフィックを仮想プライベートゲートウェイ (例: `vgw-1a2b3c4d`) にルーティングします。

送信先	ターゲット
10.0.0.0/16	ローカル
0.0.0.0/0	vgw-id

AWS Site-to-Site VPN 接続は、静的にルーティングされた Site-to-Site VPN 接続、または動的にルーティングされた Site-to-Site VPN 接続 (BGP を使用) として設定されます。静的なルーティングを選択すると、Site-to-Site VPN 接続を作成するときに、ネットワークの IP プレフィックスを手動で入力するように求められます。動的なルーティングを選択すると、IP プレフィックスは、BGP を使用して VPC に自動的にアドバタイズされます。

VPN のインスタンスはインターネットに直接接続することはできません。インターネットあてのトラフィックはすべて、まず仮想プライベートゲートウェイを経由してネットワークに向かいます。そこで、ファイアウォールと企業のセキュリティポリシーが適用されます。インスタンスが AWS 宛てのトラフィック (Amazon S3 や Amazon EC2 へのリクエストなど) を送信すると、リクエストは仮想プライベートゲートウェイを介してネットワークに向かい、その後インターネットに進み、AWS に到達します。

IPv6 のルーティング

IPv6 CIDR ブロックを VPC およびサブネットと関連付ける場合は、ルートテーブルに IPv6 トラフィックの別のルートを含めます。このシナリオのカスタムルートテーブルは次のとおりです。2 番目のエントリは、IPv6 経由で VPC 内のローカルルーティングに自動的に追加されたデフォルトルートです。

送信先	ターゲット
10.0.0.0/16	ローカル
2001:db8:1234:1a00::/56	ローカル
0.0.0.0/0	vgw-id

Security

AWS では、セキュリティグループとネットワーク ACL という 2 つの機能を使用して、VPC のセキュリティを強化できます。セキュリティグループは、インスタンスのインバウンドトラフィックとアウトバウンドトラフィックをコントロールします。ネットワーク ACL は、サブネットのインバウンドトラフィックとアウトバウンドトラフィックをコントロールします。通常、セキュリティグループで用が足りますが、VPC に追加のセキュリティレイヤーが必要な場合は、ネットワーク ACL を使用することもできます。詳細については、「」を参照してくださいAmazon VPC でのインターネットワークトラフィックのプライバシー (p. 165)

シナリオ 4 では、VPC に対してデフォルトのセキュリティグループを使用し、ネットワーク ACL は使用しません。ネットワーク ACL を使用する場合は、「プライベートサブネットのみと AWS Site-to-Site VPN アクセスを持つ VPC の推奨ネットワーク ACL ルール (p. 76)」を参照してください。

VPC に用意されているデフォルトのセキュリティグループの初期設定では、すべてのインバウンドトラフィックが拒否され、すべてのアウトバウンドトラフィックと、セキュリティグループに割り当てられているインスタンス間のすべてのトラフィックが許可されます。このシナリオでは、デフォルトのセキュリティグループにインバウンドルールを追加して、ネットワークからの SSH トラフィック (Linux) とリモートデスクトップトラフィック (Windows) を許可するためことをお勧めします。

> **Important**
>
> デフォルトのセキュリティグループでは、割り当てられたインスタンスが相互に通信するように自動的に許可されます。したがって、これを許可するためのルールを追加する必要はありません。異なるセキュリティグループを使用する場合は、これを許可するためのルールを追加する必要があります。

次の表では、VPC のデフォルトのセキュリティグループに追加する必要があるインバウンドルールについて説明します。

デフォルトのセキュリティグループ: 推奨ルール

インバウンド

送信元	プロトコル	ポート範囲	コメント
ネットワークのプライベート IPv4 アドレスの範囲	TCP	22	(Linux インスタンス) ネットワークからのインバウンド SSH トラフィックを許可する。
ネットワークのプライベート IPv4 アドレスの範囲	TCP	3389	(Windows インスタンス) ネットワークからのインバウンド RDP トラフィックを許可する。

IPv6 のセキュリティグループルール

IPv6 CIDR ブロックを VPC およびサブネットと関連付ける場合は、別のルールをセキュリティグループに追加して、インバウンドおよびアウトバウンドの IPv6 トラフィックを制御する必要があります。このシナリオでは、データベースサーバーは、IPv6 を使用した Site-to-Site VPN 接続で到達できません。したがって、セキュリティグループのルールを追加する必要はありません。

プライベートサブネットのみと AWS Site-to-Site VPN アクセスを持つ VPC の推奨ネットワーク ACL ルール

次の表に、推奨ルールを示します。これらのルールでは、明示的に必要なトラフィックを除き、すべてのトラフィックをブロックします。

Inbound

ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント
100	ホームネットワークのプライベート IP アドレスの範囲	TCP	22	許可	ホームネットワークからサブネットに対するインバウンド SSH トラフィックを許可します.
110	ホームネットワークのプライベート IP アドレスの範囲	TCP	3389	許可	ホームネットワークからサブネットに対するインバウンド RDP トラフィックを許可します.
120	ホームネットワークのプライベート IP アドレスの範囲	TCP	32768-65535	許可	送信元がサブネットであるリクエストからのインバウンドリターン トラフィックを許可します。 この範囲は一例に過ぎません。設定に使用する正しい一

					時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンドトラフィックを拒否します (変更不可能)。

Outbound

ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント
100	ホームネットワークのプライベート IP アドレスの範囲	すべて	すべて	許可	サブネットからホームネットワークに対するすべてのアウトバンドトラフィックを許可します. このルールはルール 120 も含んでいます。ただし、特定のプロトコルタイプとポート番号を使用して、このルールの適用範囲を絞り込むことができます。このルールの適用範囲を絞り込む場合、アウトバウンド応答がブロックされないようにするには、ネットワーク ACL にルール 120 を含める必要があります。

120	ホームネットワークのプライベート IP アドレスの範囲	TCP	32768-65535	許可	ホームネットワークのクライアントに対するアウトバウンド応答を許可します。 この範囲は一例に過ぎません。設定に使用する正しい一時ポートの選択の詳細については、「一時ポート (p. 212)」を参照してください。
*	0.0.0.0/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのアウトバウンドトラフィックを拒否します (変更不可能)。

IPv6 に推奨されるネットワーク ACL ルール

IPv6 がサポートされているシナリオ 4 を実装し、関連付けられた IPv6 CIDR ブロックを持つ VPC とサブネットを作成した場合は、別のルールをネットワーク ACL に追加して、インバウンドおよびアウトバウンド IPv6 トラフィックを制御する必要があります。

このシナリオでは、データベースサーバーは、IPv6 経由で VPN 通信に到達することはできません。したがって、ネットワーク ACL ルールを追加する必要はありません。以下は、サブネット間の IPv6 トラフィックを拒否するデフォルトルールです。

VPN 専用サブネットの ACL ルール

Inbound					
ルール番号	送信元 IP	プロトコル	ポート	許可/拒否	コメント
*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv6 トラフィックを拒否します (変更不可)。
Outbound					
ルール番号	送信先 IP	プロトコル	ポート	許可/拒否	コメント
*	::/0	すべて	すべて	拒否	前のルールでまだ処理されていないすべ

		てのアウトバウンド IPv6 トラフィックを拒否します (変更不可)。

VPC 設定の例

次の例を参考に VPC を作成および設定できます。

例	使用
AWS CLI を使用して IPv4 VPC とサブネットを作成する (p. 91)	AWS CLI を使用して、パブリックサブネットとプライベートサブネットを持つ VPC を作成します。
AWS CLI を使用して IPv6 VPC とサブネットを作成する (p. 96)	AWS CLI を使用して、IPv6 CIDR ブロックに関連付けられた VPC を作成し、その VPC 内に IPv6 CIDR ブロックに関連付けられたパブリックサブネットとプライベートサブネットを作成します。
パブリックサブネットとプライベートサブネットを共有する (p. 81)	複数のアカウントでプライベートサブネットとパブリックサブネットを共有します。
AWS PrivateLink および VPC ピアリングを使用するサービス (p. 81)	VPC ピアリングと AWS PrivateLink を組み合わせを使用して、プライベートサービスへのアクセスをコンシューマーに拡張します。
ミドルボックスルーティング (p. 82)	VPC に出入りするトラフィックのルーティングパスを細かく制御できます。

トランジットゲートウェイを使用して VPC を接続することもできます。

例	使用
集中型ルーター	トランジットゲートウェイを、すべての VPC、AWS Direct Connect、および AWS Site-to-Site VPN 接続を接続する集中型ルーターとして設定します。詳細については、Amazon VPC トランジットゲートウェイの「例: 集中型ルーター」を参照してください。
分離された VPC	複数の独立したルーターとしてトランジットゲートウェイを設定します。これは複数のトランジットゲートウェイを使用するのと似ていますが、ルートとアタッチメントが変わる可能性がある場合に、より高い柔軟性を提供します。詳細については、Amazon VPC トランジットゲートウェイの「例: 隔離された VPC」を参照してください。
共有サービスによる分離された VPC	共有サービスを使用する複数の分離されたルーターとしてトランジットゲートウェイを設定します。これは複数のトランジットゲートウェイを使用するのと似ていますが、ルートとアタッチメントが変わる可能性がある場合に、より高い柔軟性を提供します。詳細については、Amazon VPC Transit Gateway の「例: 共有サービスによる分離された VPC」を参照してください。

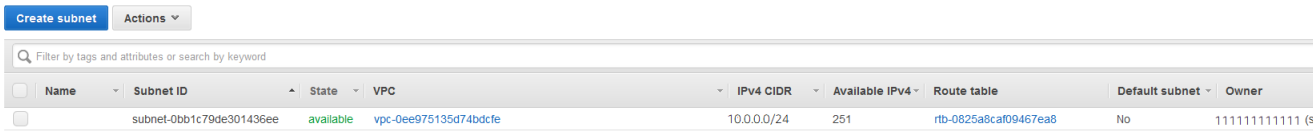
例: パブリックサブネットとプライベートサブネットの共有

インフラストラクチャ (サブネット、ルートテーブル、ゲートウェイ、CIDR 範囲など) を担当する 1 つのアカウントと、そのサブネットを使用する同じ AWS Organization にある別の複数のアカウントが必要であるシナリオについて考えてみます。VPC 所有者 (アカウント A) がルーティングインフラストラクチャ (VPC、サブネット、ルートテーブル、ゲートウェイ、ネットワーク ACL など) を作成します。アカウント D では、パブリック側アプリケーションを作成します。アカウント B とアカウント C では、インターネット接続が不要でプライベートサブネット内に配置するプライベートアプリケーションを作成します。アカウント A では、AWS Resource Access Manager を使用して、そのサブネット用にリソース共有を作成してそのサブネットを共有できます。アカウント A では、パブリックサブネットをアカウント D と共有し、プライベートサブネットをアカウント B およびアカウント C と共有します。アカウント B、アカウント C、およびアカウント D ではそのサブネット内にリソースを作成できます。各アカウントで見えるのは共有されているサブネットだけです。例えば、アカウント D で見えるのはパブリックサブネットだけです。各アカウントでは、自分のリソース (インスタンス、セキュリティグループなど) を制御できます。

アカウント A では IP インフラストラクチャ (パブリックサブネット用のルートテーブル、プライベートサブネットなど) を管理します。共有サブネット用の追加の設定は必要ないため、ルートテーブルは、共有されていないサブネットのルートテーブルと同じです。

アカウント A (アカウント ID 111111111111) は、パブリックサブネットをアカウント D (444444444444) と共有します。アカウント D では以下のサブネットが見え、[所有者] 列にはサブネットが共有されていることを表す 2 つのインジケータが表示されています。

- アカウント ID は VPC 所有者 (111111111111) の ID であり、アカウント D の ID (444444444444) とは異なっています。
- 所有者のアカウント ID の横に [共有] と表示されています。



例: AWS PrivateLink と VPC ピア接続を使用するサービス

AWS PrivateLink サービスプロバイダーは、Network Load Balancer をフロントエンドとして使用して VPC 内でサービスを実行するインスタンスを設定します。リージョン内の VPC ピアリング (VPC は同じリージョン内にある) と、リージョン間の VPC ピアリング (VPC は別のリージョン間にある) を AWS PrivateLink で使用して、VPC ピア接続間のコンシューマーへのプライベートアクセスを許可します。

リモート VPC のコンシューマーは、ピア接続間でプライベート DNS 名を使用できません。ただし、Route 53 で独自のプライベートホストゾーンを作成し、それを自分の VPC にアタッチして、同じプライベート DNS 名を使用することはできます。Amazon Route 53 Resolver でトランジットゲートウェイを使用して、接続された複数の VPC とオンプレミス環境間で PrivateLink インターフェイスエンドポイントを共有する方法については、「AWS Transit Gateway と AWS PrivateLink および Amazon Route 53 Resolver の統合」を参照してください。

次のユースケースの詳細については、「 AWS PrivateLink 経由でサービスに安全にアクセスする」を参照してください。

- SaaS アプリケーションへのプライベートアクセス
- 共有サービス
- ハイブリッドサービス
- リージョン間エンドポイントサービス
- エンドポイントサービスへのリージョン間アクセス

その他のリソース

次のトピックは、ユースケースに必要なコンポーネントの設定に役立ちます。

- VPC エンドポイントサービス
- Network Load Balancer の使用開始
- VPC ピア接続の操作
- インターフェイスエンドポイントの作成

VPC ピアリングのその他の例については、Amazon VPC ピア機能ガイドの次のトピックを参照してください。

- VPC ピア機能の設定
- サポートされていない VPC ピア接続設定

例: ミドルボックスルーティング

例えば、トラフィックをセキュリティアプライアンスにリダイレクトするなど、VPC 内のトラフィックのルーティングパスを細かく制御する場合は、VPC コンソールでミドルボックスルーティングウィザードを使用できます。ミドルボックスルーティングウィザードを使用すると、必要なルートテーブルとルート（ホップ）を自動的に作成して、必要に応じてトラフィックをリダイレクトできます。

例

- サブネット宛てのすべてのトラフィックを検査する (p. 82)
- セキュリティ VPC のゲートウェイロードバランサーの背後にあるセキュリティアプライアンス (p. 85)
- サブネット間のトラフィックを検査する (p. 87)
- 同じ VPC 内の複数のミドルボックス (p. 89)

サブネット宛てのすべてのトラフィックを検査する

インターネットゲートウェイを介して VPC にトラフィックが着信しており、EC2 インスタンスにインストールされたファイアウォールアプライアンスを使用して、サブネットを送信先（サブネット B など）とするすべてのトラフィックを検査するシナリオを考えてみます。ファイアウォールアプライアンスは、VPC のサブネット B（サブネット C）とは別のサブネットにある Amazon EC2 インスタンスにインストールおよび設定する必要があります。その後、ミドルボックスルーティングウィザードを使用して、サブネット B とインターネットゲートウェイ間のトラフィックのルートを設定できます。

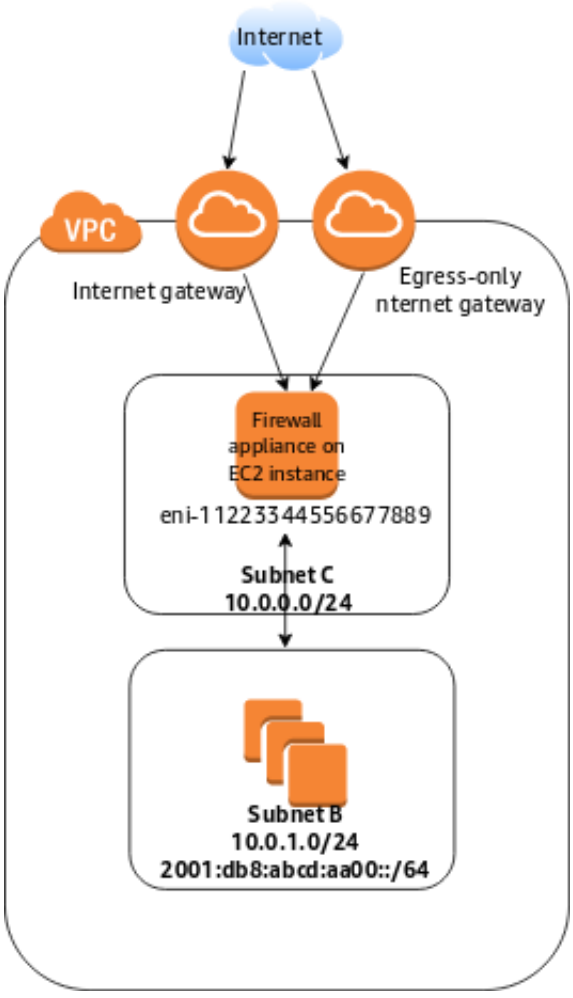
ミドルボックスルーティングウィザードは、次の操作を自動的に実行します。

- インターネットゲートウェイのルートテーブル（ルートテーブル A）、サブネット B のルートテーブル（ルートテーブル B）、サブネット C のルートテーブル（ルートテーブル C）の 3 つのルートテーブルを作成します。

- 次のセクションで説明しますが、必要なルートを新しいルートテーブルに追加してください。
- インターネットゲートウェイ、サブネット B、サブネット C に関連付けられている現在のルートテーブルの関連付けを解除します。
- ルートテーブル A をインターネットゲートウェイ (ミドルボックスルーティングウィザードの送信元)、ルートテーブル C をサブネット C (ミドルボックスルーティングウィザードのミドルボックス)、ルートテーブル B をサブネット B (ミドルボックスルーティングウィザードのデスティネーション) に関連付けます。
- ミドルボックスルーティングウィザードによって作成されたことを示すタグと、作成日を示すタグを作成します。

ミドルボックスルーティングウィザードは、既存のルートテーブルを変更しません。新しいルートテーブルを作成し、ゲートウェイおよびサブネットリソースに関連付けます。リソースが既存のルートテーブルに明示的に関連付けられている場合は、まず既存のルートテーブルの関連付けが解除され、次に新しいルートテーブルがリソースに関連付けられます。既存のルートテーブルは削除されません。

ミドルボックスルーティングウィザードを使用しない場合は、手動で設定し、サブネットとインターネットゲートウェイにルートテーブルを割り当てる必要があります。



インターネットゲートウェイルートテーブル

インターネットゲートウェイのルートテーブルには、次のルートが含まれています。

送信先	ターゲット	目的
10.0.0.0/16	ローカル	IPv4 のローカルルート
10.0.1.0/24	eni-11223344556677889	サブネット B 宛の IPv4 トラフィックをミドルボックスにルーティングする
2001:db8:1234:1a00::/56	ローカル	IPv6 のローカルルート
2001:db8:1234:1a00::/64	eni-11223344556677889	サブネット B 宛の IPv6 トラフィックをミドルボックスにルーティングする

インターネットゲートウェイと VPC の間にエッジ関連付けがあります。

ミドルボックスルーティングウィザードを使用すると、次のタグがルートテーブルに関連付けられます。

- キーを「Origin」に設定し、値を「Middlebox wizard」に設定したタグです。
- 例えば、「2021-02-18T 22:25:49 .137Z」のように、キーが「date_created」に設定され、値が作成時刻に設定されているタグ。

ミドルボックスサブネットルートテーブル

ミドルボックスサブネット（サブネット C）のルートテーブルには、次のルートが含まれます。

送信先	ターゲット	目的
10.0.0.0/16	ローカル	IPv4 のローカルルート
0.0.0.0/0	igw-id	IPv4 トラフィックをインターネットゲートウェイにルーティングする
2001:db8:1234:1a00::/56	ローカル	IPv6 のローカルルート
::/0	eigw-id	IPv6 トラフィックを Egress-only インターネットゲートウェイにルーティングする

サブネット B とサブネットの関連付けがあります。

ミドルボックスルーティングウィザードを使用すると、次のタグがルートテーブルに関連付けられます。

- キーを「Origin」に設定し、値を「Middlebox wizard」に設定したタグです。
- 例えば、「2021-02-18T 22:25:49 .137Z」のように、キーが「date_created」に設定され、値が作成時刻に設定されているタグ。

サブネットルートテーブル

ミドルボックスサブネット（サブネット C）のルートテーブルには、次のルートが含まれます。

送信先	ターゲット	目的
10.0.0.0/16	ローカル	ローカルルート
0.0.0.0/0	eni-11223344556677889	インターネット宛ての IPv4 トラフィックをミドルボックスにルーティングする
2001:db8:1234:1a00::/56	ローカル	IPv6 のローカルルート
::/0	eni-11223344556677889	インターネット宛ての IPv4 トラフィックをミドルボックスにルーティングする

サブネット C とサブネットの関連付けがあります。

ミドルボックスルーティングウィザードを使用すると、次のタグがルートテーブルに関連付けられます。

- キーを「Origin」に設定し、値を「Middlebox wizard」に設定したタグです。
- 例えば、「2021-02-18T 22:25:49 .137Z」のように、キーが「date_created」に設定され、値が作成時刻に設定されているタグ。

セキュリティ VPC のゲートウェイロードバランサーの背後にあるセキュリティアプライアンス

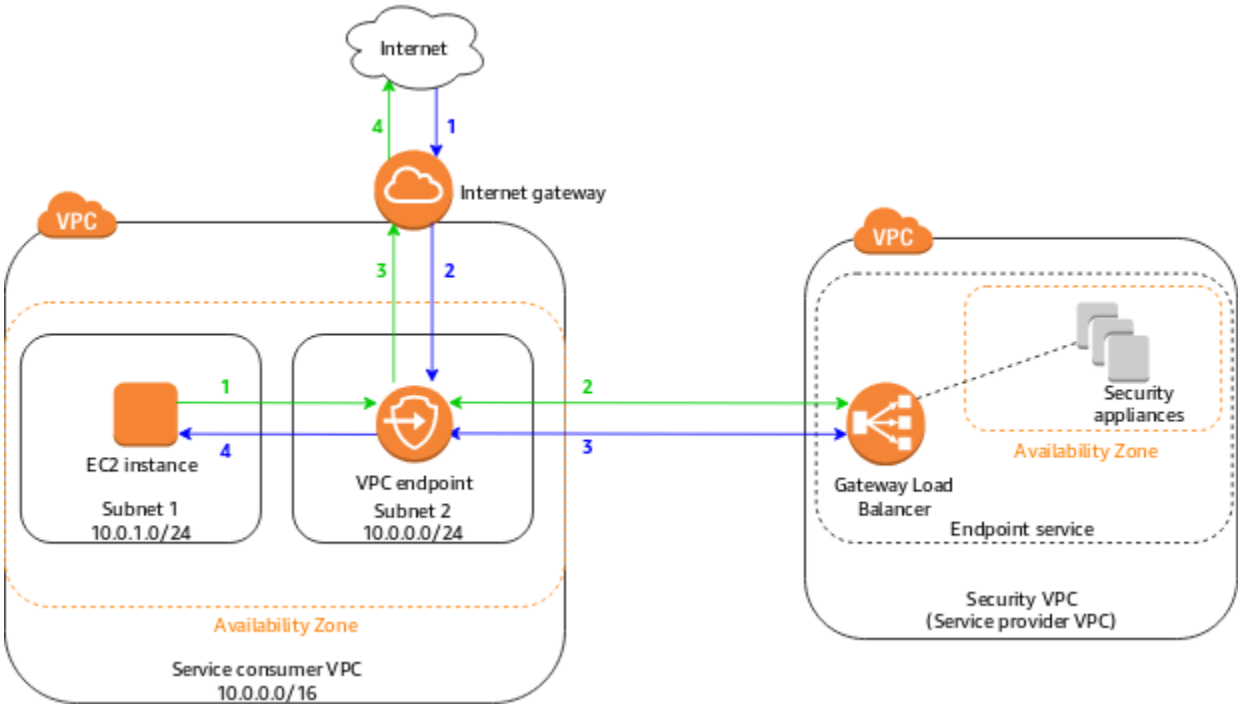
次の例では、セキュリティ VPC のGateway Load Balancer の背後に設定されたセキュリティアプライアンスのフリートを使用して、インターネットゲートウェイから VPC に入り、サブネット 1 を送信先とするトラフィックを検査します。サービスコンシューマー VPC の所有者は、VPC 内のサブネット 2 に Gateway Load Balancer エンドポイントを作成します (エンドポイントネットワークインターフェイスで表されます)。インターネットゲートウェイを経由して VPC に入るすべてのトラフィックは、まずセキュリティ VPC での検査のために Gateway Load Balancer エンドポイントにルーティングされ、その後に送信先サブネット 1 にルーティングされます。同様に、サブネット 1 から出るすべてのトラフィックは、セキュリティ VPC での検査のために Gateway Load Balancer エンドポイントにルーティングされ、その後にインターネットにルーティングされます。

ミドルボックスルーティングウィザードは、次の操作を自動的に実行します。

- ルートテーブルを作成します。
- 必要なルートを新しいルートテーブルに追加します。
- サブネットに関連付けられている現在のルートテーブルの関連付けを解除します。
- ミドルボックスルーティングウィザードが作成したルートテーブルをサブネットに関連付けます。
- ミドルボックスルーティングウィザードによって作成されたことを示すタグと、作成日を示すタグを作成します。

ミドルボックスルーティングウィザードは、既存のルートテーブルを変更しません。新しいルートテーブルを作成し、ゲートウェイおよびサブネットリソースに関連付けます。リソースが既存のルートテーブルに明示的に関連付けられている場合は、まず既存のルートテーブルの関連付けが解除され、次に新しいルートテーブルがリソースに関連付けられます。既存のルートテーブルは削除されません。

ミドルボックスルーティングウィザードを使用しない場合は、手動で設定し、サブネットとインターネットゲートウェイにルートテーブルを割り当てる必要があります。



ゲートウェイルートテーブル

インターネットゲートウェイルートテーブルには、次のルートがあります。

送信先	ターゲット	目的
10.0.0.0/16	ローカル	ローカル
10.0.1.0/24	vpc-endpoint-id	サブネット 1 を送信先とするトラフィックを Gateway Load Balancer エンドポイントにルーティングします。

ゲートウェイとエッジアソシエーションがあります。

ミドルボックスルーティングウィザードを使用すると、次のタグがルートテーブルに関連付けられます。

- キーを「Origin」に設定し、値を「Middlebox wizard」に設定したタグです。
- 例えば、「2021-02-18T 22:25:49 .137Z」のように、キーが「date_created」に設定され、値が作成時刻に設定されているタグ。

サブネット 1 ルートテーブル

サブネット 1 ルートテーブルには以下のルートがあります。

送信先	ターゲット	目的
10.0.0.0/16	ローカル	ローカルルート

送信先	ターゲット	目的
0.0.0.0/0	vpc-endpoint-id	ローカルではないトラフィックを Gateway Load Balancer エンドポイントにルーティングします。これにより、(インターネットを送信先とする) サブネットから出るすべてのトラフィックが Gateway Load Balancer エンドポイントに最初にルーティングされます。

サブネット 1 とサブネットの関連付けがあります。

ミドルボックスルーティングウィザードを使用すると、次のタグがルートテーブルに関連付けられます。

- キーを「Origin」に設定し、値を「Middlebox wizard」に設定したタグです。
- 例えば、「2021-02-18T 22:25:49 .137Z」のように、キーが「date_created」に設定され、値が作成時刻に設定されているタグ。

サブネット 2 ルートテーブル

サブネット 2 ルートテーブルには以下のルートがあります。

送信先	ターゲット	目的
10.0.0.0/16	ローカル	ローカルルート - インターネットを起点とするトラフィックについては、ローカルルートによって、サブネット 1 の送信先にルーティングされます。
0.0.0.0/0	igw-id	すべてのトラフィックをインターネットゲートウェイにルーティングする

サブネット 2 とサブネットの関連付けがあります。

ルートテーブルには、次のタグが関連付けられています。

- キーを「Origin」に設定し、値を「Middlebox wizard」に設定したタグです。
- 例えば、「2021-02-18T 22:25:49 .137Z」のように、キーが「date_created」に設定され、値が作成時刻に設定されているタグ。

サブネット間のトラフィックを検査する

VPC に複数のサブネットがあり、EC2 インスタンスにインストールされたファイアウォールアプライアンスを使用して、サブネット A と B の間のトラフィックを検査するシナリオを考えてみます。VPC 内の別のサブネット C の EC2 インスタンスに、ファイアウォールアプライアンスを設定してインストールします。アプライアンスは、サブネット A と B の間を移動するすべてのトラフィックを検査します。

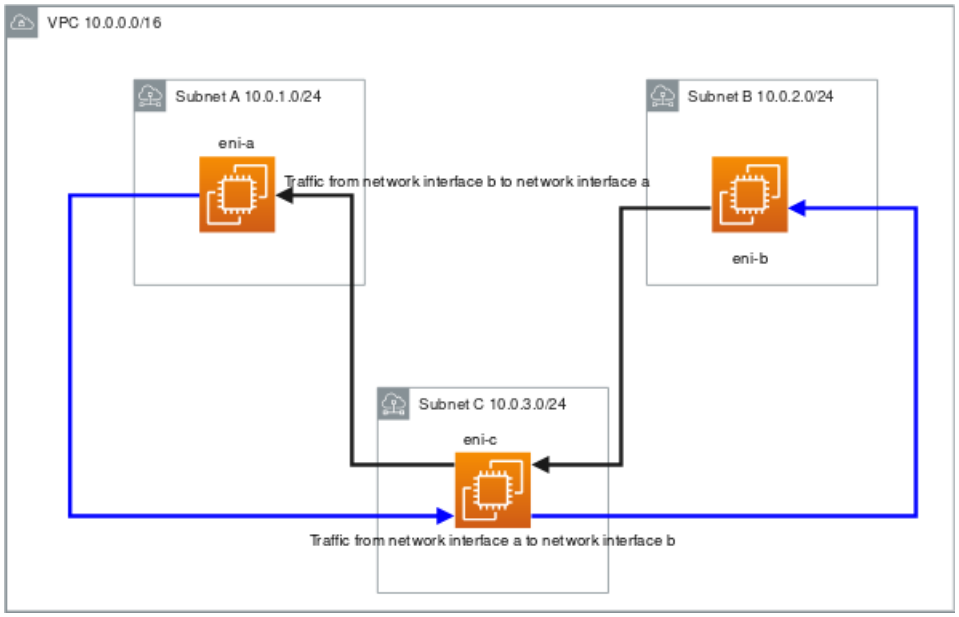
VPC とミドルボックスサブネットのメインルートを使用します。サブネット A と B には、それぞれカスタムルートテーブルがあります。

ミドルボックスルーティングウィザードは、次の操作を自動的に実行します。

- ルートテーブルを作成します。
- 必要なルートを新しいルートテーブルに追加します。
- サブネットに関連付けられている現在のルートテーブルの関連付けを解除します。
- ミドルボックスルーティングウィザードが作成したルートテーブルをサブネットに関連付けます。
- ミドルボックスルーティングウィザードによって作成されたことを示すタグと、作成日を示すタグを作成します。

ミドルボックスルーティングウィザードは、既存のルートテーブルを変更しません。新しいルートテーブルを作成し、ゲートウェイおよびサブネットリソースに関連付けます。リソースが既存のルートテーブルに明示的に関連付けられている場合は、まず既存のルートテーブルの関連付けが解除され、次に新しいルートテーブルがリソースに関連付けられます。既存のルートテーブルは削除されません。

ミドルボックスルーティングウィザードを使用しない場合は、手動で設定し、サブネットとインターネットゲートウェイにルートテーブルを割り当てる必要があります。



カスタムサブネット A ルートテーブル

サブネットのルートテーブルには、次のルートがあります。

送信先	ターゲット	目的
10.0.0.0/16	ローカル	ローカルルート
10.0.2.0/24	eni-c	サブネット B を送信先とするトラフィックをミドルボックスにルーティングする

サブネット A とサブネットの関連付けがあります。

ミドルボックスルーティングウィザードを使用すると、次のタグがルートテーブルに関連付けられます。

- キーを「Origin」に設定し、値を「Middlebox wizard」に設定したタグです。

- 例えば、「2021-02-18T 22:25:49 .137Z」のように、キーが「date_created」に設定され、値が作成時刻に設定されているタグ。

カスタムサブネット B ルートテーブル

サブネット B のルートテーブルには、次のルートがあります。

送信先	ターゲット	目的
10.0.0.0/16	ローカル	ローカルルート
10.0.1.0/24	eni-c	サブネット A を送信先とするトラフィックをミドルボックスにルーティングする

サブネット B とサブネットの関連付けがあります。

ミドルボックスルーティングウィザードを使用すると、次のタグがルートテーブルに関連付けられます。

- キーを「Origin」に設定し、値を「Middlebox wizard」に設定したタグです。
- 例えば、「2021-02-18T 22:25:49 .137Z」のように、キーが「date_created」に設定され、値が作成時刻に設定されているタグ。

メインルートテーブル

VPC とサブネット C のメインルートテーブルには、次のルートがあります。

送信先	ターゲット	目的
10.0.0.0/16	ローカル	ローカルルート

サブネット C とサブネットの関連付けがあります。

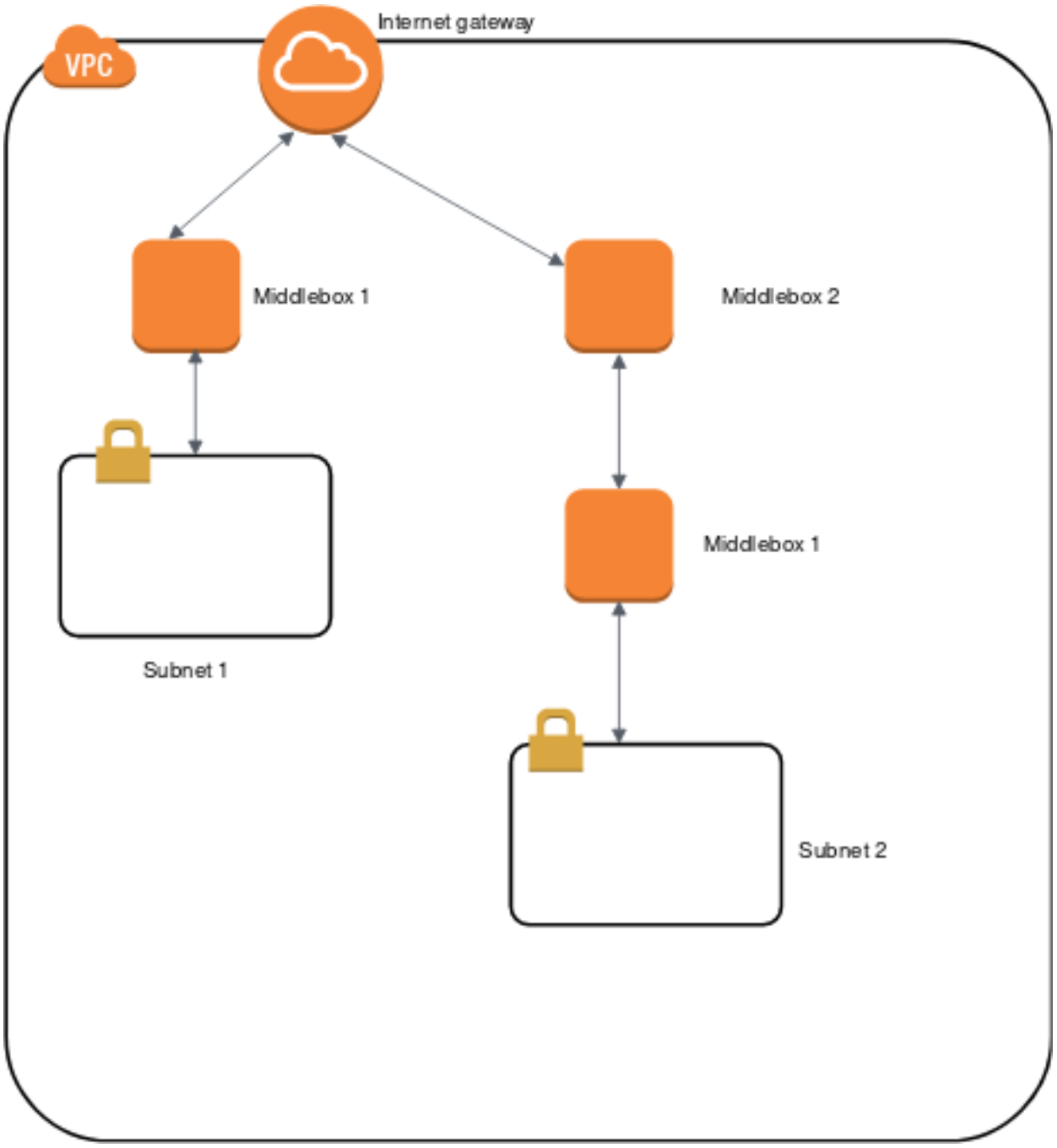
ミドルボックスルーティングウィザードを使用すると、次のタグがルートテーブルに関連付けられます。

- キーを「Origin」に設定し、値を「Middlebox wizard」に設定したタグです。
- 例えば、「2021-02-18T 22:25:49 .137Z」のように、キーが「date_created」に設定され、値が作成時刻に設定されているタグ。

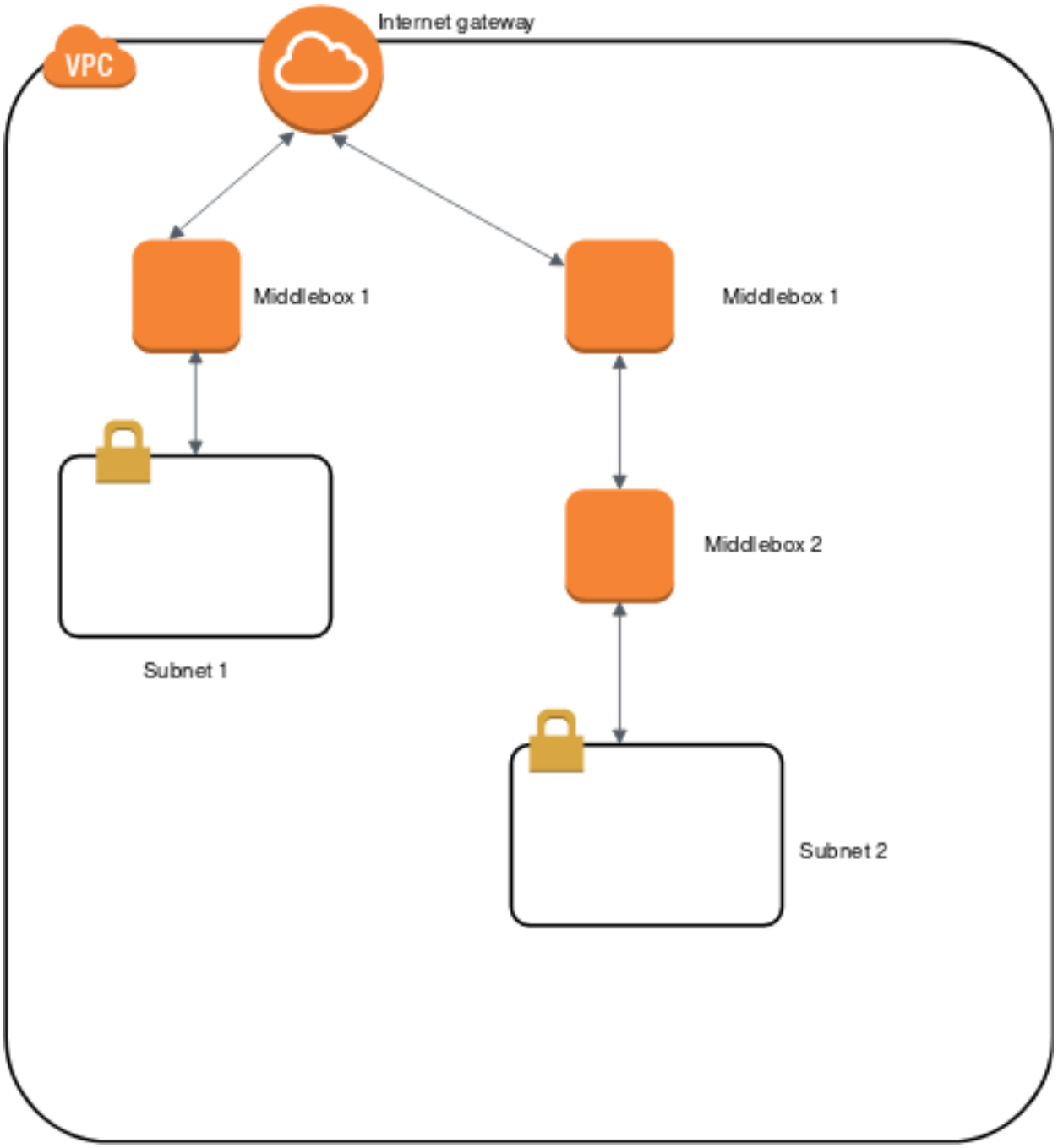
同じ VPC 内の複数のミドルボックス

同じ VPC 内の複数のサブネットのトラフィックを検査する同じミドルボックス

インターネットゲートウェイを介して VPC にトラフィックが着信しており、ミドルボックス 1 を使用してサブネット 1 を送信先とするすべてのトラフィックを検査するシナリオを考えてみます。　同じ VPC 内で、ミドルボックス 2 とミドルボックス 1 を使用して、サブネット 2 を送信先とするトラフィックを検査します。ミドルボックスに関連付けられたサブネットのルートテーブルには、インターネットゲートウェイにトラフィックをルーティングする `0.0.0.0/0` のルートが必要であるため、次の構成はサポートされていません。



この構成に同じミドルボックスを配置する場合は、ミドルボックスが両方のサブネットで同じホップ位置 (例えば、インターネットゲートウェイの後のホップ) にある必要があります。これは、ミドルボックス 2 で関連付けられたサブネットのルートテーブルに、ミドルボックス 1 のサブネットにトラフィックをルーティングする `0.0.0.0/0` のルートがあることを意味します。インターネットゲートウェイにトラフィックをルーティングする `0.0.0.0/0` のルートを持つ、ミドルボックス 1 に関連付けられたルートテーブルにルートがあります。



例: AWS CLI を使用して IPv4 VPC とサブネットを作成

次の例では、AWS CLI コマンドを使用して、IPv4 CIDR ブロックを含むデフォルトではない VPC と、VPC 内にパブリックサブネット、プライベートサブネットを作成しています。VPC およびサブネットを作成したら、パブリックサブネット内にインスタンスを起動して、接続できるようになります。開始

するには、最初に AWS CLI をインストールして設定する必要があります。詳細については、「AWS CLI のインストール」を参照してください。

次の AWS リソースを作成します。

- VPC
- 2 つのサブネット
- インターネットゲートウェイ
- ルートテーブル
- EC2 インスタンス

タスク

ステップ 1: VPC とサブネットを作成する

最初のステップは、VPC、2 つのサブネットを作成することです。この例では、VPC の CIDR ブロック `10.0.0.0/16` を使用しますが、別の CIDR ブロックを選択できます。詳細については、「」を参照してくださいVPC とサブネットのサイズ設定 (p. 109)

AWS CLI を使用して VPC およびサブネットを作成するには

1. 次の create-vpc コマンドを使用して、`10.0.0.0/16` CIDR ブロックで VPC を作成します。

   ```
   aws ec2 create-vpc --cidr-block 10.0.0.0/16 --query Vpc.VpcId --output text
   ```

   このコマンドは、新しい VPC の ID を返します。次に例を示します。

   ```
   vpc-2f09a348
   ```

2. 前述の手順の VPC ID を使用して、次の create-subnet コマンドを使用して、`10.0.1.0/24` CIDR ブロックでサブネットを作成します。

   ```
   aws ec2 create-subnet --vpc-id vpc-2f09a348 --cidr-block 10.0.1.0/24
   ```

3. VPC で、`10.0.0.0/24` CIDR ブロックを持つ 2 番目のサブネットを作成します。

   ```
   aws ec2 create-subnet --vpc-id vpc-2f09a348 --cidr-block 10.0.0.0/24
   ```

ステップ 2: サブネットをパブリックにする

VPC およびサブネットを作成した後、VPC にインターネットゲートウェイをアタッチして、カスタムルートテーブルを作成し、インターネットゲートウェイへのサブネットのルーティングを構成すると、サブネットをパブリックサブネットにすることができます。

サブネットをパブリックサブネットにするには

1. 以下の create-internet-gateway コマンドを使用して、インターネットゲートウェイを作成します。

```
aws ec2 create-internet-gateway --query InternetGateway.InternetGatewayId --output text
```

このコマンドは、新しいインターネットゲートウェイの ID を返します。次に例を示します。

```
igw-1ff7a07b
```

2. 前述の手順 の ID を使用して、次の attach-internet-gateway コマンドを使用してインターネットゲートウェイをVPCに接続します。

```
aws ec2 attach-internet-gateway --vpc-id vpc-2f09a348 --internet-gateway-
id igw-1ff7a07b
```

3. 次の create-route-table コマンドを使用して、VPC のカスタムルートテーブルを作成します。

```
aws ec2 create-route-table --vpc-id vpc-2f09a348 --query RouteTable.RouteTableId --
output text
```

このコマンドは、新しいルートテーブルの ID を返します。次に例を示します。

```
rtb-c1c8faa6
```

4. 次の create-route コマンドを使用して、すべてのトラフィック (0.0.0.0/0) がインターネットゲートウェイを指すルートをルートテーブルに作成します。

```
aws ec2 create-route --route-table-id rtb-c1c8faa6 --destination-cidr-block 0.0.0.0/0
 --gateway-id igw-1ff7a07b
```

5. (オプション) ルートが作成され有効になっていることを確認するには、以下の describe-route-tables コマンドを実行します。

```
aws ec2 describe-route-tables --route-table-id rtb-c1c8faa6
```

```
{
    "RouteTables": [
        {
            "Associations": [],
            "RouteTableId": "rtb-c1c8faa6",
            "VpcId": "vpc-2f09a348",
            "PropagatingVgws": [],
            "Tags": [],
            "Routes": [
                {
                    "GatewayId": "local",
                    "DestinationCidrBlock": "10.0.0.0/16",
                    "State": "active",
                    "Origin": "CreateRouteTable"
                },
                {
                    "GatewayId": "igw-1ff7a07b",
                    "DestinationCidrBlock": "0.0.0.0/0",
                    "State": "active",
                    "Origin": "CreateRoute"
                }
            ]
        }
    ]
}
```

6. ルートテーブルは現在、サブネットには関連付けられていません。サブネットからのトラフィックがインターネットゲートウェイにルーティングされるよう、ルートテーブルを VPC のサブネットに関連付ける必要があります。次の describe-subnets コマンドを使用して、サブネット ID を取得します。`--filter` オプションは、サブネットを新しい VPC のみに制限し、`--query` オプションは、サブネットID とその CIDR ブロックのみを返します。

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-2f09a348" --query
 "Subnets[*].{ID:SubnetId,CIDR:CidrBlock}"
```

```
[
    {
        "CIDR": "10.0.1.0/24",
        "ID": "subnet-b46032ec"
    },
    {
        "CIDR": "10.0.0.0/24",
        "ID": "subnet-a46032fc"
    }
]
```

7. 例えば `subnet-b46032ec` などのカスタムルートテーブルに関連付けるサブネットを選択し、associate-route-table コマンドを使用して関連付けることができます。このサブネットはパブリックサブネットです。

```
aws ec2 associate-route-table  --subnet-id subnet-b46032ec --route-table-id rtb-
c1c8faa6
```

8. （オプション）次の modify-subnet-attribute コマンドを使用して、サブネットに起動されたインスタンスが自動的にパブリック IP アドレスを受け取るように、サブネットのパブリック IP アドレス指定の動作を変更できます。　これを行わない場合は、起動後に Elastic IP アドレスをインスタンスに関連付けて、インスタンスがインターネットからアクセスできるようにします。

```
aws ec2 modify-subnet-attribute --subnet-id subnet-b46032ec --map-public-ip-on-launch
```

ステップ 3: サブネット内にインスタンスを起動する

サブネットがパブリックであること、および、サブネット内のインスタンスにインターネット経由でアクセスできることをテストするには、パブリックサブネット内でインスタンスを起動して接続します。最初に、インスタンスに関連付けるセキュリティグループと、インスタンスに接続するキーペアを作成する必要があります。セキュリティグループの詳細については、VPC のセキュリティグループ (p. 188) を参照してください。キーペアの詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Amazon EC2 キーペア」を参照してください。

パブリックサブネット内のインスタンスを起動して接続するには

1. キーペアを作成して、`--query` オプションと `--output` テキストオプションを使用し、`.pem` 拡張機能でプライベートキーをファイルに直接パイプします。

```
aws ec2 create-key-pair --key-name MyKeyPair --query "KeyMaterial" --output text
 > MyKeyPair.pem
```

この例では、Amazon Linux インスタンスを起動します。Linux または Mac OS X オペレーティングシステムの SSH クライアントを使用して Linux インスタンスに接続する場合は、次のコマンドを使用してプライベートキーファイルの権限を設定すると、お客様以外のユーザーはそれを読み取ることができないようになります。

```
chmod 400 MyKeyPair.pem
```

2. create-security-group コマンドを使用して、VPC にセキュリティグループを作成します。

```
aws ec2 create-security-group --group-name SSHAccess --description "Security group for
 SSH access" --vpc-id vpc-2f09a348
```

```
{
    "GroupId": "sg-e1fb8c9a"
}
```

authorize-security-group-ingress コマンドを使用して、あらゆる場所からの SSH アクセスを許可するルールを追加します。

```
aws ec2 authorize-security-group-ingress --group-id sg-e1fb8c9a --protocol tcp --
port 22 --cidr 0.0.0.0/0
```

Note

0.0.0.0/0 を使用すると、すべての IPv4 アドレスから SSH 経由でインスタンスにアクセスすることが許可されます。これは、この短期間の実習では許容されますが、本稼働環境では、特定の IP アドレスまたはアドレス範囲のみ許可してください。

3. 作成したセキュリティグループとキーペアを使用して、パブリックサブネット内でインスタンスを起動します。出力内のインスタンスのインスタンス ID をメモしておきます。

```
aws ec2 run-instances --image-id ami-a4827dc9 --count 1 --instance-type t2.micro --key-
name MyKeyPair --security-group-ids sg-e1fb8c9a --subnet-id subnet-b46032ec
```

Note

この例で、AMI は米国東部 (バージニア北部) リージョンの Amazon Linux AMI です。別のリージョンの場合、リージョン内の適した AMI の AMI ID が必要になります。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Linux AMI の検索」を参照してください。

4. インスタンスに接続するには、そのインスタンスが running 状態になっている必要があります。インスタンスの状態と IP アドレスを記述するには、次のコマンドを使用します。

```
aws ec2 describe-instances --instance-id i-0146854b7443af453 --query
 "Reservations[*].Instances[*].{State:State.Name,Address:PublicIpAddress}"
```

出力例を次に示します。

```
[
    [
        {
            "State": "running",
            "Address": "52.87.168.235"
        }
    ]
]
```

5. インスタンスが実行状態にあるときは、次のコマンドで、Linux または Mac OS X コンピュータの SSH クライアントを使用してそのインスタンスに接続できます。

```
ssh -i "MyKeyPair.pem" ec2-user@52.87.168.235
```

Windows コンピュータから接続する場合は、「PuTTY を使用した Windows から Linux インスタンスへの接続」の手順を使用します。

ステップ 4: クリーンアップする

インスタンスに接続できることを確認したあと、そのインスタンスが不要であれば終了できます。これを行うには、terminate-instances コマンドを使用します。この例で作成したそのほかのリソースを削除するには、 の順にコマンドを実行します。

1. セキュリティグループを削除する:

```
aws ec2 delete-security-group --group-id sg-e1fb8c9a
```

2. サブネットを削除する:

```
aws ec2 delete-subnet --subnet-id subnet-b46032ec
```

```
aws ec2 delete-subnet --subnet-id subnet-a46032fc
```

3. カスタムルートテーブルを削除する:

```
aws ec2 delete-route-table --route-table-id rtb-c1c8faa6
```

4. VPC からのインターネットゲートウェイのデタッチ:

```
aws ec2 detach-internet-gateway --internet-gateway-id igw-1ff7a07b --vpc-id vpc-2f09a348
```

5. インターネットゲートウェイの削除:

```
aws ec2 delete-internet-gateway --internet-gateway-id igw-1ff7a07b
```

6. VPC の削除:

```
aws ec2 delete-vpc --vpc-id vpc-2f09a348
```

例: AWS CLI を使用して IPv6 VPC とサブネットを作成

次の例では、AWS CLI コマンドを使用して、IPv6 CIDR ブロックを持つデフォルト以外の VPC 、パブリックサブネット、アウトバウンドのインターネットアクセス専用のプライベートサブネットを作成します。VPC およびサブネットを作成したら、パブリックサブネット内にインスタンスを起動して、接続できるようになります。プライベートサブネット内のインスタンスを起動し、インターネットに接続できることを確認できます。開始するには、最初に AWS CLI をインストールして設定する必要があります。詳細については、「AWS CLI のインストール」を参照してください。

次の AWS リソースを作成します。

- VPC

- 2 つのサブネット
- インターネットゲートウェイ
- ルートテーブル
- EC2 インスタンス

タスク

ステップ 1: VPC とサブネットを作成する

最初のステップは、VPC、2 つのサブネットを作成することです。この例では、VPC の IPv4 CIDR ブロック `10.0.0.0/16` を使用しますが、別の CIDR ブロックを選択することもできます。詳細については、「VPC とサブネットのサイズ設定 (p. 109)」を参照してください。

AWS CLI を使用して VPC およびサブネットを作成するには

1. IPv6 CIDR ブロック (`10.0.0.0/16`) を持つ VPC を作成し、CIDR ブロックを VPC と関連付けます。

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --amazon-provided-ipv6-cidr-block
```

返される出力で、VPC ID を書き留めておいてください。

```
{
    "Vpc": {
        "VpcId": "vpc-2f09a348",
        ...
}
```

2. VPC に関連付けられている IPv6 CIDR ブロックを得るために、VPC について説明します。

```
aws ec2 describe-vpcs --vpc-id vpc-2f09a348
```

```
{
    "Vpcs": [
        {
            ...
            "Ipv6CidrBlockAssociationSet": [
                {
                    "Ipv6CidrBlock": "2001:db8:1234:1a00::/56",
                    "AssociationId": "vpc-cidr-assoc-17a5407e",
                    "Ipv6CidrBlockState": {
                        "State": "ASSOCIATED"
                    }
                }
            ],
            ...
```

```
}
```

3. (前述のステップで返された範囲の) 10.0.0.0/24IPv4 CIDR ブロックと 2001:db8:1234:1a00::/64IPv6 CIDR ブロックを持つサブネット作成します 。

```
aws ec2 create-subnet --vpc-id vpc-2f09a348 --cidr-block 10.0.0.0/24 --ipv6-cidr-
block 2001:db8:1234:1a00::/64
```

4. 10.0.1.0/24IPv4 CIDR ブロックと 2001:db8:1234:1a01::/64IPv6 CIDR ブロックを持つ VPC に 2 番目のサブネットを作成します。

```
aws ec2 create-subnet --vpc-id vpc-2f09a348 --cidr-block 10.0.1.0/24 --ipv6-cidr-
block 2001:db8:1234:1a01::/64
```

ステップ 2: パブリックサブネットを設定する

VPC およびサブネットを作成した後、VPC にインターネットゲートウェイをアタッチして、カスタムルートテーブルを作成し、インターネットゲートウェイへのサブネットのルーティングを構成すると、サブネットをパブリックサブネットにすることができます。この例では、IPv4 トラフィックと IPv6 トラフィックをすべてインターネットゲートウェイへルーティングするルートテーブルが作成されます。

サブネットをパブリックサブネットにするには

1. インターネットゲートウェイを作成します。

```
aws ec2 create-internet-gateway
```

返される出力に、インターネットゲートウェイ ID を書き留めます。

```
{
    "InternetGateway": {
        ...
        "InternetGatewayId": "igw-1ff7a07b",
        ...
    }
}
```

2. 前のステップの ID を使用して、VPC にインターネットゲートウェイをアタッチします。

```
aws ec2 attach-internet-gateway --vpc-id vpc-2f09a348 --internet-gateway-
id igw-1ff7a07b
```

3. VPC に対してカスタムルートテーブルを作成します。

```
aws ec2 create-route-table --vpc-id vpc-2f09a348
```

返される出力で示されたルートテーブル ID を書き留めます。

```
{
    "RouteTable": {
        ...
        "RouteTableId": "rtb-c1c8faa6",
        ...
    }
}
```

4. すべての IPv6 トラフィック (`::/0`) をインターネットゲートウェイに向けるルートをルートテーブルに作成します。

```
aws ec2 create-route --route-table-id rtb-c1c8faa6 --destination-ipv6-cidr-block ::/0
 --gateway-id igw-1ff7a07b
```

   Note

   IPv4 トラフィックに対してもパブリックサブネットを使用する場合は、インターネットゲートウェイを指す `0.0.0.0/0` トラフィック のルートを別途追加する必要があります。

5. ルートが作成され有効になっていることを確認するには、ルートテーブルを記述して結果を表示できます。

```
aws ec2 describe-route-tables --route-table-id rtb-c1c8faa6
```

```
{
    "RouteTables": [
        {
            "Associations": [],
            "RouteTableId": "rtb-c1c8faa6",
            "VpcId": "vpc-2f09a348",
            "PropagatingVgws": [],
            "Tags": [],
            "Routes": [
                {
                    "GatewayId": "local",
                    "DestinationCidrBlock": "10.0.0.0/16",
                    "State": "active",
                    "Origin": "CreateRouteTable"
                },
                {
                    "GatewayId": "local",
                    "Origin": "CreateRouteTable",
                    "State": "active",
                    "DestinationIpv6CidrBlock": "2001:db8:1234:1a00::/56"
                },
                {
                    "GatewayId": "igw-1ff7a07b",
                    "Origin": "CreateRoute",
                    "State": "active",
                    "DestinationIpv6CidrBlock": "::/0"
                }
            ]
        }
    ]
}
```

6. ルートテーブルは、現時点でどのサブネットにも関連付けられていません。サブネットからのトラフィックがインターネットゲートウェイにルーティングされるよう、ルートテーブルを VPC のサブネットに関連付けます。最初に、ID を取得するためのサブネットを記述します。`--filter` オプションを使用して新しい VPC のサブネットだけを返し、`--query` オプションを使用してサブネット ID と IPv4 および IPv6 CIDR ブロックだけを返します。

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-2f09a348" --query
 "Subnets[*].
{ID:SubnetId,IPv4CIDR:CidrBlock,IPv6CIDR:Ipv6CidrBlockAssociationSet[*].Ipv6CidrBlock}"
```

```
[
    {
```

```
        "IPv6CIDR": [
            "2001:db8:1234:1a00::/64"
        ],
        "ID": "subnet-b46032ec",
        "IPv4CIDR": "10.0.0.0/24"
    },
    {
        "IPv6CIDR": [
            "2001:db8:1234:1a01::/64"
        ],
        "ID": "subnet-a46032fc",
        "IPv4CIDR": "10.0.1.0/24"
    }
]
```

7. カスタムルートテーブルに関連付けるサブネット、例えば subnet-b46032ec を選択できます。このサブネットはパブリックサブネットになります。

```
aws ec2 associate-route-table  --subnet-id subnet-b46032ec --route-table-id rtb-c1c8faa6
```

ステップ 3: Egress-Only プライベートサブネットを設定する

2 番目のサブネットを VPC に設定すると、IPv6 の Egress-Only プライベートサブネットとして使用できます。このサブネット内に起動されるインスタンスは、Egress-Only インターネットゲートウェイを通じて IPv6 経由でインターネットにアクセスできますが (例: ソフトウェアアップデートの取得)、インターネットのホストがインスタンスに到達することはできません。

サブネットを Egress-Only プライベートサブネットにするには

1. VPC の Egress-Only インターネットゲートウェイを作成します。返される出力で示されたインターネットゲートウェイ ID を書き留めます。

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-2f09a348
```

```
{
    "EgressOnlyInternetGateway": {
        "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
        "Attachments": [
            {
                "State": "attached",
                "VpcId": "vpc-2f09a348"
            }
        ]
    }
}
```

2. VPC に対してカスタムルートテーブルを作成します。返される出力で示されたルートテーブル ID を書き留めます。

```
aws ec2 create-route-table --vpc-id vpc-2f09a348
```

3. すべての IPv6 トラフィック (::/0) を Egress-Only インターネットゲートウェイに向けるルートをルートテーブルに作成します。

```
aws ec2 create-route --route-table-id rtb-abc123ab --destination-ipv6-cidr-block ::/0
 --egress-only-internet-gateway-id eigw-015e0e244e24dfe8a
```

4. ルートテーブルを VPC 内の 2 番目のサブネット (前のセクションで説明済み) に関連付けます 。このサブネットは、Egress-Only IPv6 でインターネットアクセスするプライベートサブネットになります。

```
aws ec2 associate-route-table --subnet-id subnet-a46032fc --route-table-id rtb-abc123ab
```

ステップ 4: サブネットの IPv6 アドレス動作を変更する

サブネット内で起動されたインスタンスが IPv6 アドレスを自動的に取得できるように、サブネットの IP アドレス動作を変更できます。サブネットにインスタンスを起動すると、サブネットのアドレス範囲内からインスタンスのプライマリネットワークインターフェイス (eth0) に IPv6 アドレスが 1 つ割り当てられます。

```
aws ec2 modify-subnet-attribute --subnet-id subnet-b46032ec --assign-ipv6-address-on-
creation
```

```
aws ec2 modify-subnet-attribute --subnet-id subnet-a46032fc --assign-ipv6-address-on-
creation
```

ステップ 5: パブリックサブネット内にインスタンスを起動する

パブリックサブネットがパブリックであること、および、サブネット内のインスタンスにインターネットでアクセスできることをテストするには、パブリックサブネット内でインスタンスを起動して接続します。最初に、インスタンスに関連付けるセキュリティグループと、インスタンスに接続するキーペアを作成する必要があります。セキュリティグループの詳細については、VPC のセキュリティグループ (p. 188) を参照してください。キーペアの詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Amazon EC2 キーペア」を参照してください。

パブリックサブネット内のインスタンスを起動して接続するには

1. キーペアを作成して、--query オプションと --output テキストオプションを使用し、.pem 拡張機能でプライベートキーをファイルに直接パイプします。

```
aws ec2 create-key-pair --key-name MyKeyPair --query "KeyMaterial" --output text
 > MyKeyPair.pem
```

この例では、Amazon Linux インスタンスを起動します。Linux または OS X オペレーティングシステムの SSH クライアントを使用して Linux インスタンスに接続する場合は、次のコマンドを使用してプライベートキーファイルの権限を設定すると、お客様以外のユーザーはそれを読み取ることができないようになります。

```
chmod 400 MyKeyPair.pem
```

2. create-security-group コマンドを使用して、VPC にセキュリティグループを作成します。

```
aws ec2 create-security-group --group-name SSHAccess --description "Security group for
 SSH access" --vpc-id vpc-2f09a348
```

```
{
    "GroupId": "sg-e1fb8c9a"
}
```

authorize-security-group-ingress コマンドを使用して、任意の IPv6 アドレスからの SSH アクセスを許可するルールを追加します。以下の構文が動作するのは Linux および macOS のみです。Windows で動作する構文については、AWS CLI コマンドリファレンスの例セクションを参照してください。

```
aws ec2 authorize-security-group-ingress --group-id sg-e1fb8c9a --ip-permissions
 '[{"IpProtocol": "tcp", "FromPort": 22, "ToPort": 22, "Ipv6Ranges": [{"CidrIpv6":
 "::/0"}]}]'
```

Note

::/0 を使用すると、すべての IPv6 アドレスから SSH 経由でインスタンスにアクセスできるようになります。これは、この短期間の実習では許容されますが、本稼働環境では、特定の IP アドレスまたはアドレス範囲のみがインスタンスへのアクセスを許可されます。

3. 作成したセキュリティグループとキーペアを使用して、パブリックサブネット内にインスタンスを起動します。出力内のインスタンスのインスタンス ID をメモしておきます。

```
aws ec2 run-instances --image-id ami-0de53d8956e8dcf80 --count 1 --instance-
type t2.micro --key-name MyKeyPair --security-group-ids sg-e1fb8c9a --subnet-id subnet-
b46032ec
```

Note

この例で、AMI は米国東部 (バージニア北部) リージョンの Amazon Linux AMI です。別のリージョンの場合、リージョン内に適した AMI の AMI ID が必要になります。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Linux AMI の検索」を参照してください。

4. インスタンスに接続するには、そのインスタンスが running 状態になっている必要があります。インスタンスを記述してその状態を確認し、IPv6 アドレスを書き留めておきます。

```
aws ec2 describe-instances --instance-id i-0146854b7443af453
```

出力例を次に示します。

```
{
    "Reservations": [
        {
            ...
            "Instances": [
                {
                    ...
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    ...
                    "NetworkInterfaces": {
                        "Ipv6Addresses": {
                            "Ipv6Address": "2001:db8:1234:1a00::123"
```

```
                        }
                    ...
                }
            ]
        }
    ]
}
```

5. インスタンスが実行状態にあるときは、次のコマンドで、Linux または OS X コンピューターの SSH クライアントを使用してそのインスタンスに接続できます。ローカルコンピューターに、設定済みの IPv6 アドレスが必要です。

```
ssh -i "MyKeyPair.pem" ec2-user@2001:db8:1234:1a00::123
```

Windows コンピュータから接続する場合は、「PuTTY を使用した Windows から Linux インスタンスへの接続」の手順を使用します。

ステップ 6: プライベートサブネット内にインスタンスを起動する

Egress-Only プライベートサブネット内のインスタンスがインターネットにアクセスできることをテストするには、プライベートサブネット内のインスタンスを起動し、パブリックサブネット内の踏み台インスタンスを使用してインスタンスに接続します (前のセクションで起動したインスタンスを使用できます)。最初に、インスタンスのセキュリティグループを作成する必要があります。セキュリティグループには、踏み台インスタンスが SSH を使用して接続できるようにするルール、およびインスタンスにインターネットからアクセス可能でないという確認を `ping6` コマンド (ICMPv6 トラフィック) に許可するルールが必要です。

1. create-security-group コマンドを使用して、VPC にセキュリティグループを作成します。

```
aws ec2 create-security-group --group-name SSHAccessRestricted --description "Security
 group for SSH access from bastion" --vpc-id vpc-2f09a348
```

authorize-security-group-ingress コマンドを使用して、パブリックサブネット内のインスタンスの IPv6 アドレスからのインバウンド SSH アクセスを許可するルールと、すべての ICMPv6 トラフィックを許可するルールを追加します。以下の構文が動作するのは Linux および macOS のみです。Windows で動作する構文については、AWS CLI コマンドリファレンスの例セクションを参照してください。

```
{
    "GroupId": "sg-aabb1122"
}
```

```
aws ec2 authorize-security-group-ingress --group-id sg-aabb1122 --ip-permissions
 '[{"IpProtocol": "tcp", "FromPort": 22, "ToPort": 22, "Ipv6Ranges": [{"CidrIpv6":
 "2001:db8:1234:1a00::123/128"}]}]'
```

```
aws ec2 authorize-security-group-ingress --group-id sg-aabb1122 --ip-permissions
 '[{"IpProtocol": "58", "FromPort": -1, "ToPort": -1, "Ipv6Ranges": [{"CidrIpv6":
 "::/0"}]}]'
```

2. プライベートサブネット内にインスタンスを起動します。この際、作成したセキュリティグループと、パブリックサブネット内にインスタンスを起動する際に使用したのと同一のキーペアを使用します。

```
aws ec2 run-instances --image-id ami-a4827dc9 --count 1 --instance-type t2.micro --key-
name MyKeyPair --security-group-ids sg-aabb1122 --subnet-id subnet-a46032fc
```

describe-instances コマンドを使用して、インスタンスが稼働していることを確認し、IPv6 アドレスを取得します。

3. ローカルマシンの SSH エージェント転送を設定し、パブリックサブネットのインスタンスに接続します。

   Linux の場合、次のコマンドを使用します。

   ```
   ssh-add MyKeyPair.pem
   ssh -A ec2-user@2001:db8:1234:1a00::123
   ```

   OS X の場合、次のコマンドを使用します。

   ```
   ssh-add -K MyKeyPair.pem
   ssh -A ec2-user@2001:db8:1234:1a00::123
   ```

   Windows の場合は、次の手順を使用します。Windows (PuTTY) 用に SSH エージェント転送を設定するには (p. 243)IPv6 アドレスを使用して、パブリックサブネットのインスタンスに接続します。
4. パブリックサブネットのインスタンス (踏み台インスタンス) から、IPv6 アドレスを使用して、プライベートサブネットのインスタンスに接続します。

   ```
   ssh ec2-user@2001:db8:1234:1a01::456
   ```
5. プライベートインスタンスから、ICMP が有効なウェブサイトに対して ping6 コマンドを実行して、インターネットに接続できることをテストします。次に例を示します。

   ```
   ping6 -n ietf.org
   ```

   ```
   PING ietf.org(2001:1900:3001:11::2c) 56 data bytes
   64 bytes from 2001:1900:3001:11::2c: icmp_seq=1 ttl=46 time=73.9 ms
   64 bytes from 2001:1900:3001:11::2c: icmp_seq=2 ttl=46 time=73.8 ms
   64 bytes from 2001:1900:3001:11::2c: icmp_seq=3 ttl=46 time=73.9 ms
   ...
   ```
6. インターネットのホストがプライベートサブネットのインスタンスに到達できないことをテストするには、IPv6 が有効になっているコンピューターから ping6 コマンドを使用します。タイムアウト応答が返ります。有効なレスポンスが表示される場合、インスタンスはインターネットからアクセス可能です。プライベートサブネットに関連付けられているルートテーブルを確認し、IPv6 トラフィックからインターネットゲートウェイへのルートがないことを確認します。

   ```
   ping6 2001:db8:1234:1a01::456
   ```

ステップ 7: クリーンアップ

パブリックサブネットのインスタンスに接続できること、プライベートサブネットのインスタンスがインターネットにアクセスできることを確認したら、不要になったインスタンスを終了できます。これを行うには、terminate-instances コマンドを使用します。この例で作成したそのほかのリソースを削除するには、 の順にコマンドを実行します。

1. セキュリティグループを削除する:

```
aws ec2 delete-security-group --group-id sg-e1fb8c9a
```

```
aws ec2 delete-security-group --group-id sg-aabb1122
```

2. サブネットを削除する:

```
aws ec2 delete-subnet --subnet-id subnet-b46032ec
```

```
aws ec2 delete-subnet --subnet-id subnet-a46032fc
```

3. カスタムルートテーブルを削除する:

```
aws ec2 delete-route-table --route-table-id rtb-c1c8faa6
```

```
aws ec2 delete-route-table --route-table-id rtb-abc123ab
```

4. VPC からのインターネットゲートウェイのデタッチ:

```
aws ec2 detach-internet-gateway --internet-gateway-id igw-1ff7a07b --vpc-id vpc-2f09a348
```

5. インターネットゲートウェイの削除:

```
aws ec2 delete-internet-gateway --internet-gateway-id igw-1ff7a07b
```

6. Egress-Only インターネットゲートウェイを削除する:

```
aws ec2 delete-egress-only-internet-gateway  --egress-only-internet-gateway-
id eigw-015e0e244e24dfe8a
```

7. VPC の削除:

```
aws ec2 delete-vpc --vpc-id vpc-2f09a348
```

VPC とサブネット

Amazon Virtual Private Cloud (Amazon VPC) を開始するには、VPC とサブネットを作成します。Amazon VPC の一般的な概要については、「Amazon VPC とは? (p. 1)」を参照してください。

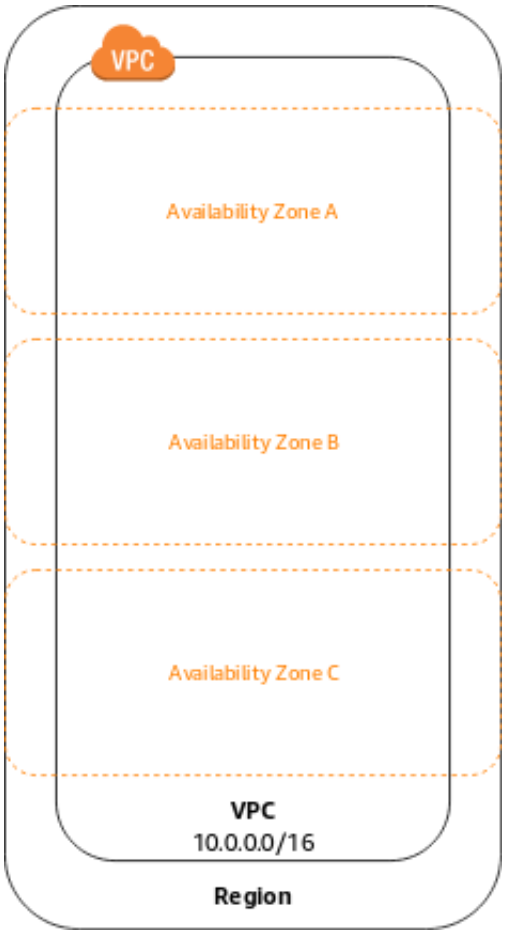
目次

VPC とサブネットの基本

Virtual Private Cloud (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 を作成するときに、その VPC に対して、IPv4 アドレスの範囲を Classless Inter-Domain Routing (CIDR) ブロックの形式で指定する必要があります (例: `10.0.0.0/16`)。これは VPC のプライマリ CIDR ブロックです。CIDR 表記の詳細については、RFC 4632 を参照してください。

VPC は、同じリージョンのアベイラビリティーゾーンすべてにおよびます。次の図は、IPv4 CIDR ブロックがある新しい VPC を示しています。

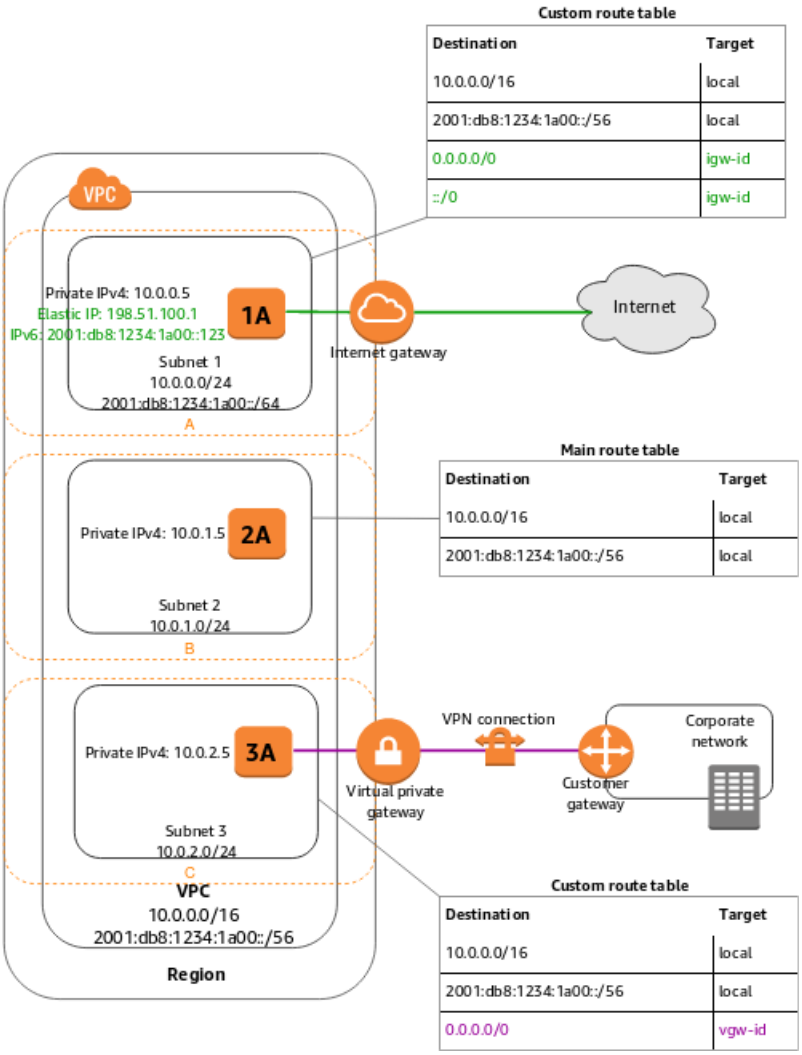


VPC を作成したら、アベイラビリティーゾーンごとに 1 つ以上のサブネットを追加します。サブネットは、VPC の IP アドレスの範囲です。起動することができますAWS特定のサブネットには、EC2 インスタンスなどの EC2 リソースをデプロイできます。サブネットを作成する際、VPC CIDR ブロックのサブセットである、サブネットの CIDR ブロックを指定します。各サブネットが完全に 1 つのアベイラビリティーゾーン内に含まれている必要があります。1 つのサブネットが複数のゾーンにまたがることはできません。個別のアベイラビリティーゾーンでインスタンスを起動することにより、1 つの場所で発生した障害からアプリケーションを保護できます。

必要に応じて、サブネットをローカルゾーンに追加できます。ローカルゾーンは、コンピューティング、ストレージ、データベース、その他の厳選したサービスをエンドユーザーの近くに配置する AWS インフラストラクチャデプロイメントです。ローカルゾーンを使用すると、エンドユーザーは 1 桁のミリ秒のレイテンシーを必要とするアプリケーションを実行できます。詳細については、Linux インスタンス用の Amazon EC2 ユーザーガイドの「リージョンとアベイラビリティーゾーン」を参照してください。

さらに、オプションで IPv6 CIDR ブロックを VPC に割り当てて、IPv6 CIDR ブロックをサブネットに割り当てることができます。

次の図は、複数のアベイラビリティーゾーンにある複数のサブネットで設定された VPC を示しています。1A、2A、および 3A は VPC のインスタンスです。IPv6 CIDR ブロックは VPC に関連付けられ、IPv6 CIDR ブロックはサブネット 1 に関連付けられます。インターネットゲートウェイはインターネットを介した通信を有効にし、仮想プライベートネットワーク (VPN) 接続は、企業ネットワークとの通信を有効にします。



サブネットのトラフィックがインターネットゲートウェイにルーティングされる場合、そのサブネットはパブリックサブネットと呼ばれます。この図では、サブネット 1 がパブリックサブネットです。パブリックサブネット内のインスタンスが IPv4 を介してインターネットと通信することが必要な場合は、そのインスタンスにパブリック IPv4 アドレスまたは Elastic IP アドレス (IPv4) が割り当てられている必要があります。パブリック IPv4 アドレスの詳細については、「パブリック IPv4 アドレス (p. 126)」を参照してください。パブリックサブネットのインスタンスに IPv6 を介してインターネットと通信させたい場合は、インスタンスに IPv6 アドレスが必要です。

インターネットゲートウェイへのルートがないサブネットは、プライベートサブネットと呼ばれます。この図では、サブネット 2 がプライベートサブネットです。

インターネットゲートウェイへのルートがなく、トラフィックが Site-to-Site VPN 接続の仮想プライベートゲートウェイにルーティングされているサブネットは、VPN のみのサブネットと呼ばれます。この図では、サブネット 3 が VPN のみのサブネットです。現在、Site-to-Site VPN 接続を介した IPv6 トラフィックはサポートされていません。

詳細については、AWS Site-to-Site VPN ユーザーガイドの「VPC 設定の例 (p. 80)」、「インターネットゲートウェイ (p. 221)」、および「AWS Site-to-Site VPN とは」を参照してください。

Note

サブネットの種類にかかわらず、サブネット内の IPv4 アドレス範囲は常にプライベートです。そのアドレスブロックがインターネットに公開されることはありません。

アカウント内に作成できる VPC とサブネットの数にはクォータがあります。詳細については、「」を参照してくださいAmazon VPC クォータ (p. 362)

VPC とサブネットのサイズ設定

Amazon VPC は IPv4 および IPv6 のアドレス指定をサポートしており、CIDR ブロックサイズのクォータはそれぞれ異なります。デフォルトでは、すべての VPC とサブネットに IPv4 CIDR が必要で、この動作は変更できません。オプションで IPv6 CIDR ブロックを VPC と関連付けることができます。

IP アドレスの割り当てについては、「VPC の IP アドレス指定 (p. 124)」を参照してください。

目次

IPv4 用の VPC とサブネットのサイズ設定

VPC を作成するときに、その VPC の IPv4 CIDR ブロックを指定する必要があります。許可されるブロックサイズは、/16 ネットマスク (65,536 個の IP アドレス) から /28 ネットマスク (16 個の IP アドレス) の間です。VPC を作成したら、VPC とセカンダリ CIDR ブロックを関連付けることができます。詳細については、「」を参照してくださいIPv4 CIDR ブロックを VPC に追加する (p. 110)

VPC を作成するときは、RFC 1918 に指定されているように、プライベート IPv4 アドレス範囲からの CIDR ブロックを指定することをお勧めします。

RFC 1918 の範囲	CIDR ブロックの例
10.0.0.0 - 10.255.255.255 (10/8 プレフィックス)	VPC は /16 以下 (10.0.0.0/16 など) にする必要があります。
172.16.0.0 - 172.31.255.255 (172.16/12 プレフィックス)	VPC は /16 以下 (172.31.0.0/16 など) にする必要があります。
192.168.0.0 - 192.168.255.255 (192.168/16 プレフィックス)	VPC を小さく (192.168.0.0/20 などに) することができます。

VPC を作成する場合、RFC 1918 に指定されているプライベート IPv4 アドレスの範囲から外れる、パブリックにルーティングできる CIDR ブロックを使うこともできますが、本書ではプライベート IPv4 アドレスを VPC の CIDR 範囲内にある IPv4 アドレスという意味で使います。

Note

他の AWS サービスで使用することを目的として VPC を作成している場合は、サービスドキュメントで、IP アドレス範囲またはネットワークコンポーネントに特定の要件があるかどうかを確認します。

サブネットの CIDR ブロックは、VPC の CIDR ブロック (VPC のサブネットが 1 つの場合)、または VPC の CIDR ブロックのサブネット (サブネットが複数の場合) と同じにすることができます。許可されているのは、/28 ネットマスクから /16 ネットマスクの間のブロックサイズです。VPC に複数のサブネットを作成する場合、サブネットの CIDR ブロックは重複できません。

例えば、CIDR ブロック 10.0.0.0/24 を持つ VPC を作成した場合、その VPC では 256 個の IP アドレスがサポートされます。この CIDR ブロックは 2 つのサブネットに分割でき、それぞれのサブネットで 128 個の IP アドレスがサポートされています。一方のサブネットでは CIDR ブロック 10.0.0.0/25 (アドレス 10.0.0.0 ~ 10.0.0.127) が、もう一方のサブネットでは CIDR ブロック 10.0.0.128/25 (アドレス 10.0.0.128 ~ 10.0.0.255) が使用されます。

インターネット上には、IPv4 サブネット CIDR ブロックの計算と作成に役立つツールがあります。「サブネット計算ツール」や「CIDR 計算ツール」などの用語を検索して、お客様のニーズに合ったツールを見つけることができます。ネットワーク技術グループが、サブネットに指定する CIDR ブロックを特定することもできます。

各サブネット CIDR ブロックの最初の 4 つの IP アドレスと最後の IP アドレスは使用できず、インスタンスに割り当てることができません。例えば、CIDR ブロック 10.0.0.0/24 を持つサブネットの場合、次の 5 つの IP アドレスが予約されます。

- 10.0.0.0: ネットワークアドレスです。
- 10.0.0.1: VPC ルーター用に AWS で予約されています。
- 10.0.0.2: で予約されています。AWSDNS サーバーの IP アドレスは、VPC ネットワーク範囲のベースにプラス 2 したものです。複数の CIDR ブロックを持つ VPC の場合、DNS サーバーの IP アドレスはプライマリ CIDR にあります。また、VPC 内のすべての CIDR ブロックに対して、各サブネットの範囲 + 2 のベースを予約します。詳細については、「」を参照してくださいAmazon DNS サーバー (p. 267)
- 10.0.0.3: 将来の利用のために AWS で予約されています。
- 10.0.0.255: ネットワークブロードキャストアドレスです。VPC ではブロードキャストがサポートされないため、このアドレスを予約します。

コマンドラインツールまたは Amazon EC2 API を使用して VPC またはサブネットを作成すると、CIDR ブロックは自動で正規形式に変更されます。例えば、CIDR ブロックに 100.68.0.18/18 を指定すると、100.68.0.0/18 の CIDR ブロックが作成されます。

IPv4 CIDR ブロックを VPC に追加する

VPC とセカンダリ IPv4 CIDR ブロックを関連付けることができます。CIDR ブロックを VPC に関連付けると、ルートが VPC ルートテーブルに自動的に追加され、VPC 内でのルーティングが可能になります (送信先は CIDR ブロックで、ターゲットは local)。

次の例では、左側の VPC に 1 つの CIDR ブロック (10.0.0.0/16) と 2 つのサブネットがあります。右側の VPC は、2 番目の CIDR ブロック (10.2.0.0/16) を追加し、2 番目の CIDR の範囲から新しいサブネットを作成した後の、同じ VPC のアーキテクチャを表します。

Destination	Target
10.0.0.0/16	local

Destination	Target
10.0.0.0/16	local
10.2.0.0/16	local

CIDR ブロックを VPC に追加する場合は、次のルールが適用されます。

- 許可されているのは、`/28` ネットマスクから `/16` ネットマスクの間のブロックサイズです。
- CIDR ブロックは、VPC に関連付けられている既存の CIDR ブロックと重複してはいけません。
- 使用できる IPv4 アドレスの範囲には制限があります。詳細については、「」を参照してくださいIPv4 CIDR ブロック関連付けの制限 (p. 112)
- 既存の CIDR ブロックのサイズを増減することはできません。
- VPC に関連付けることができる CIDR ブロックの数と、ルートテーブルに追加できるルートの数にはクォータがあります。そのため、クォータを超えると CIDR ブロックを関連付けることはできなくなります。詳細については、「」を参照してくださいAmazon VPC クォータ (p. 362)
- CIDR ブロックは、VPC ルートテーブルのいずれかのルートの送信先 CIDR 範囲と同じ、またはそれ以上に大きくすることはできません。例えば、プライマリ CIDR ブロックが `10.2.0.0/16` である VPC では、仮想プライベートゲートウェイへの送信先 `10.0.0.0/24` を持つルートテーブル内に、既存のルートがあります。`10.0.0.0/16` 範囲内のセカンダリ CIDR ブロックを関連付けるとします。既存のルートが原因で、`10.0.0.0/24` 以上の CIDR ブロックを関連付けることはできません。ただし、`10.0.0.0/25` 以下のセカンダリ CIDR ブロックを関連付けることはできます。

- ClassicLink の VPC を有効にしている場合、`10.0.0.0/16` および `10.1.0.0/16` 範囲からの CIDR ブロックを関連付けることはできますが、 `10.0.0.0/8` の範囲の他の CIDR ブロックを関連付けることはできません。
- VPC ピアリング接続の一部である VPC に IPv4 CIDR ブロックを追加する場合は、次のルールが適用されます。
  - VPC ピアリング接続が `active` の場合、ピア VPC の CIDR ブロックと重複していない VPC に CIDR ブロックを追加できます。
  - VPC ピアリング接続が `pending-acceptance` の場合、リクエスタ VPC の所有者は、アクセプタ VPC の CIDR ブロックと重複しているかどうかにかかわらず、VPC に CIDR ブロックを追加できません。アクセプタ VPC の所有者がピアリング接続を受け入れるか、またはリクエスタ VPC の所有者が VPC ピアリング接続要求を削除し、CIDR ブロックを追加してから、新しい VPC ピアリング接続を要求する必要があります。
  - VPC ピアリング接続が `pending-acceptance` の場合、アクセプタ VPC の所有者は CIDR ブロックを VPC に追加できます。セカンダリ CIDR ブロックがリクエスタ VPC の CIDR ブロックと重複している場合、VPC ピアリング接続要求は失敗し、承諾されません。
- AWS Direct Connect を使用して Direct Connect ゲートウェイ経由で複数の VPC に接続する場合、Direct Connect ゲートウェイに関連付けられた VPC 間では重複する CIDR ブロックが許可されません。Direct Connect ゲートウェイに関連付けられたいずれかの VPC に CIDR ブロックを追加する場合は、追加する CIDR ブロックが、他の関連付けられた VPC の既存の CIDR ブロックと重複しないことを確認してください。詳細については、AWS Direct Connect ユーザーガイドの「Direct Connect ゲートウェイ」を参照してください。
- CIDR ブロックは、追加または削除に伴い、以下の状態を経過します: `associating` | `associated` | `disassociating` | `disassociated` | `failing` | `failed`。CIDR ブロックは、`associated` 状態にあるときに、使用可能です。

以下の表に、VPC のプライマリ CIDR ブロックが存在する IPv4 アドレス範囲に依存する、許可および制限された CIDR ブロック関連付けの概要を示します。

IPv4 CIDR ブロック関連付けの制限

プライマリ VPC CIDR ブロックが存在する IP アドレス範囲	制限された CIDR ブロックの関連付け	許可された CIDR ブロックの関連付け
10.0.0.0/8	他の RFC 1918* の範囲 (172.16.0.0/12 から 192.168.0.0/16) からの CIDR ブロック。 プライマリ CIDR が 10.0.0.0/15 の範囲内にある場合、10.0.0.0/16 の範囲から CIDR ブロックを追加することはできません。 198.19.0.0/16 の範囲からの CIDR ブロック。	制限されていない 10.0.0.0/8 の範囲からの他の CIDR。 パブリックにルーティング可能な IPv4 CIDR ブロック (RFC 1918 以外)、または 100.64.0.0/10 の範囲からの CIDR ブロック。
172.16.0.0/12	他の RFC 1918* の範囲 (10.0.0.0/8 から 192.168.0.0/16) からの CIDR ブロック。 172.31.0.0/16 の範囲からの CIDR ブロック。 198.19.0.0/16 の範囲からの CIDR ブロック。	制限されていない 172.16.0.0/12 の範囲からの他の CIDR。 パブリックにルーティング可能な IPv4 CIDR ブロック (RFC 1918 以外)、または 100.64.0.0/10 の範囲からの CIDR ブロック。

プライマリ VPC CIDR ブロックが存在する IP アドレス範囲	制限された CIDR ブロックの関連付け	許可された CIDR ブロックの関連付け
192.168.0.0/16	他の RFC 1918* の範囲 (172.16.0.0/12 から 10.0.0.0/8) からの CIDR ブロック。 198.19.0.0/16 の範囲からの CIDR ブロック。	192.168.0.0/16 の範囲からの他の CIDR。 パブリックにルーティング可能な IPv4 CIDR ブロック (RFC 1918 以外)、または 100.64.0.0/10 の範囲からの CIDR ブロック。
198.19.0.0/16	RFC 1918* 範囲からの CIDR ブロック。	パブリックにルーティング可能な IPv4 CIDR ブロック (RFC 1918 以外)、または 100.64.0.0/10 の範囲からの CIDR ブロック。
パブリックにルーティング可能な CIDR ブロック (RFC 1918 以外)、または 100.64.0.0/10 の範囲からの CIDR ブロック。	RFC 1918* 範囲からの CIDR ブロック。 198.19.0.0/16 の範囲からの CIDR ブロック。	パブリックにルーティング可能なその他の IPv4 CIDR ブロック (RFC 1918 以外)、または 100.64.0.0/10 の範囲からの CIDR ブロック。

*RFC 1918 の範囲は、RFC 1918 で指定されているプライベート IPv4 アドレス範囲です

VPC に関連付け済みの CIDRブロックの関連付けを解除することができます。ただし、元の VPC (プライマリ CIDR ブロック) を作成した CIDR ブロックの関連付けを解除することはできません。Amazon VPC コンソールで VPC のプライマリ CIDR を表示するには、[VPC] を選択して自分の VPC を選択し、[CIDR ブロック] の最初のエントリを書き留めます。または、describe-vpcs コマンドを使用することもできます。

```
aws ec2 describe-vpcs --vpc-id vpc-1a2b3c4d
```

プライマリ CIDR は最上位の `CidrBlock` 要素で返されます。

```
{
    "Vpcs": [
        {
            "VpcId": "vpc-1a2b3c4d",
            "InstanceTenancy": "default",
            "Tags": [
                {
                    "Value": "MyVPC",
                    "Key": "Name"
                }
            ],
            "CidrBlockAssociations": [
                {
                    "AssociationId": "vpc-cidr-assoc-3781aa5e",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                },
                {
                    "AssociationId": "vpc-cidr-assoc-0280ab6b",
                    "CidrBlock": "10.2.0.0/16",
                    "CidrBlockState": {
```

```
                    "State": "associated"
                }
            }
        ],
        "State": "available",
        "DhcpOptionsId": "dopt-e0fe0e88",
        "CidrBlock": "10.0.0.0/16",
        "IsDefault": false
    }
]
}
```

IPv6 用の VPC とサブネットのサイズ設定

単一の IPv6 CIDR ブロックをアカウントの既存の VPC に関連付けるか、または新しい VPC の作成時に関連付けることができます。CIDR ブロックは `/56` の固定長プレフィックスです。Amazon の IPv6 アドレスプールから IPv6 CIDR ブロックをリクエストできます。

IPv6 CIDR ブロックと VPC を関連付けている場合、IPv6 CIDR ブロック を VPC の既存のサブネットに関連付けるか、または新しいサブネットを作成するときに関連付けることができます。サブネットの IPv6 CIDR ブロックは `/64` の固定長プレフィックスです。

例えば、VPC を作成して VPC に Amazon が提供する IPv6 CIDR ブロックを関連付けるよう指定します。VPC には `2001:db8:1234:1a00::/56` の IPv6 CIDR ブロックが割り当てられます。IP アドレスの範囲を自分で選択することはできません。サブネットを作成し、この範囲から IPv6 CIDR ブロックを関連付けることができます。例: `2001:db8:1234:1a00::/64`。

IPv6 サブネット CIDR ブロックの計算と作成に役立つツール (IPv6 Address Planner など) をインターネットで入手できます。「IPv6 サブネット計算ツール」や「IPv6 CIDR 計算ツール」などの用語を検索して、自分のニーズに合った他のツールを見つけることができます。ネットワーク技術グループが、サブネットに指定する IPv6 CIDR ブロックを特定することもできます。

サブネットから IPv6 CIDR ブロックの関連付けを解除し、VPC から IPv6 CIDR ブロックの関連付けを解除できます。VPC から IPv6 CIDR ブロックの関連付けを解除すると、IPv6 CIDR ブロックと VPC を後で再び関連付けた場合に同じ CIDR を受け取ることは期待できません。

各サブネット CIDR ブロックの最初の 4 つの IPv6 アドレスと最後の IPv6 アドレスは使用できず、インスタンスに割り当てることができません。例えば、CIDR ブロック `2001:db8:1234:1a00/64` を持つサブネットの場合、次の 5 つの IP アドレスが予約されます。

- `2001:db8:1234:1a00::`
- `2001:db8:1234:1a00::1`
- `2001:db8:1234:1a00::2`
- `2001:db8:1234:1a00::3`
- `2001:db8:1234:1a00:ffff:ffff:ffff:ffff`

サブネットのルーティング

各サブネットをルートテーブルに関連付ける必要があります。サブネットを出るアウトバウンドトラフィックに対して許可されるルートは、このテーブルによって指定されます。作成するすべてのサブネットが、VPC のメインルートテーブルに自動的に関連付けられます。この関連付けを変更し、メインルートテーブルのコンテンツを変更できます。詳細については、「」を参照してくださいVPC のルートテーブル (p. 294)

前の図では、サブネット 1 に関連付けられたルートテーブルは、すべての IPv4 トラフィック (`0.0.0.0/0`) と IPv6 トラフィック (`::/0`) をインターネットゲートウェイ (例: `igw-1a2b3c4d`) にルー

ティングします。インスタンス 1A には IPv4 Elastic IP アドレスと IPv6 アドレスがあるため、IPv4 と IPv6 の両方でインターネットからアクセスできます。

Note

(IPv4 のみ) インスタンスに関連付けられる Elastic IPv4 アドレスまたはパブリック IPv4 アドレスは、VPC のインターネットゲートウェイ経由でアクセスします。インスタンスと別のネットワーク間の AWS Site-to-Site VPN 接続経由で流れるトラフィックは、インターネットゲートウェイではなく仮想プライベートゲートウェイを通過するため、Elastic IPv4 アドレスやパブリック IPv4 アドレスにはアクセスしません。

インスタンス 2A はインターネットにアクセスできませんが、VPC の他のインスタンスにはアクセスできます。ネットワークアドレス変換 (NAT) ゲートウェイまたはインスタンスを使用して、VPC のインスタンスによる IPv4 インターネットへのアウトバウンド接続の開始を許可し、インターネットからの未承諾のインバウンド接続を拒否できます。Elastic IP アドレスは限られた数しかアロケートできないため、静的なパブリック IP アドレスを必要とするインスタンスが多く存在する場合は、NAT デバイスの使用をお勧めします。詳細については、「」を参照してくださいVPC の NAT デバイス (p. 236) IPv6 経由のインターネットへのアウトバウンドのみの通信を開始するには、送信のみのインターネットゲートウェイを使用できます。詳細については、「」を参照してくださいEgress-Only インターネットゲートウェイ (p. 228)

サブネット 3 に関連付けられたルートテーブルは、すべての IPv4 トラフィック (`0.0.0.0/0`) を仮想プライベートゲートウェイ (例: `vgw-1a2b3c4d`) にルーティングします。インスタンス 3A は Site-to-Site VPN 接続を介して企業ネットワーク内のコンピュータに到達できます。

サブネットのセキュリティ

AWS では、セキュリティグループとネットワーク ACL という 2 つの機能を使用して、VPC のセキュリティを強化できます。セキュリティグループは、インスタンスのインバウンドトラフィックとアウトバウンドトラフィックをコントロールします。ネットワーク ACL は、サブネットのインバウンドトラフィックとアウトバウンドトラフィックをコントロールします。通常、セキュリティグループで用が足りますが、VPC に追加のセキュリティレイヤーが必要な場合は、ネットワーク ACL を使用することもできます。詳細については、「」を参照してくださいAmazon VPC でのインターネットワークトラフィックのプライバシー (p. 165)

設計により、各サブネットをネットワーク ACL に関連付ける必要があります。作成するサブネットはすべて、VPC のデフォルトのネットワーク ACL に自動的に関連付けられます。この関連付けを変更し、デフォルトのネットワーク ACL のコンテンツを変更できます。詳細については、「」を参照してくださいネットワーク ACL (p. 200)

VPC またはサブネットでフローログを作成し、VPC またはサブネットでネットワークインターフェイスとの間を行き来するトラフィックをキャプチャできます。個別のネットワークインターフェイスでフローログを作成することもできます。フローログは CloudWatch Logs または Amazon S3 に公開されます。詳細については、「」を参照してくださいVPC フローログ (p. 325)

VPC とサブネットの使用

次の手順は、VPC およびサブネットを手動で作成するためです。ゲートウェイとルーティングテーブルを手動で追加する必要もあります。または、Amazon VPC ウィザードを使用して VPC とそのサブネット、ゲートウェイ、ルーティングテーブルを一度に作成できます。詳細については、「」を参照してくださいVPC 設定の例 (p. 80)

タスク

- VPC を作成する (p. 116)
- VPC を表示 (p. 117)
- VPC にサブネットを作成する (p. 117)

VPC を作成する

空の VPC は、Amazon VPC コンソールを使用して作成できます。

コンソールを使用して VPC を作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [VPC] を選択し、[Create Subnet] を選択します。
3. 必要に応じて、次の VPC の詳細を指定します。

   - [Name tag]: オプションで、VPC の名前を指定できます。これにより、`Name` というキーと指定した値を含むタグが作成されます。
   - [IPv4 CIDR ブロック]: VPC 用の IPv4 CIDR ブロックを指定します。指定できる最小の CIDR ブロックは /28 で、最大の CIDR ブロックは /16 です。RFC 1918 で規定されているプライベート (パブリックにルーティングできない) IP アドレス範囲から CIDR ブロックを指定することをお勧めします。例えば、`10.0.0.0/16` や `192.168.0.0/16` から指定します。

     Note

     パブリックにルーティング可能な IPv4 アドレスの範囲を指定できます。ただし、現在、VPC 内のパブリックにルーティング可能な CIDR ブロックからのインターネットへの直接アクセスはサポートしていません。Windows インスタンスは `224.0.0.0` から `255.255.255.255` (クラス D とクラス E の IP アドレス範囲) の VPC では正しく起動できません。
   - [IPv6 CIDR ブロック]: オプションで IPv6 CIDR ブロックを VPC と関連付けることができます。次のいずれかのオプションを選択し、[Select CIDR (CIDR の選択)] を選択します。
     - [Amazon-provided IPv6 CIDR block (Amazon が提供する IPv6 CIDR ブロック)]: Amazon の IPv6 アドレスプールから IPv6 CIDR ブロックをリクエストします。[Network Border Group] (ネットワーク境界グループ) で、AWS による IP アドレスのアドバタイズ元となるグループを選択します。
     - [IPv6 CIDR owned by me (自分が所有する IPv6 CIDR)]: (BYOIP) IPv6 アドレスプールから IPv6 CIDR ブロックを割り当てます。[Pool (プール)] で、IPv6 CIDR ブロックの割り当て元となる IPv6 アドレスプールを選択します。
   - テナンシー: テナンシーオプションを選択します。専有テナントでは、シングルテナントのハードウェアでインスタンスが確実に動作することが保証されます。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「ハードウェア専有インスタンス」を参照してください。
   - (オプション) タグを追加または削除します。

     [タグの追加] [タグの追加] を選択して、以下を実行します。
     - [キー] にはキー名を入力します。
     - [値] にキー値を入力します。

     [タグを削除] タグのキーと値の右側にある [削除] を選択します。

4. [Create] を選択します。

または、コマンドラインツールを使用できます。

コマンドラインツールを使用して VPC を作成するには

- create-vpc (AWS CLI)
- New-EC2Vpc (AWS Tools for Windows PowerShell)

コマンドラインツールを使用して VPC を記述するには

- describe-vpcs (AWS CLI)
- Get-EC2Vpc (AWS Tools for Windows PowerShell)

IP アドレスについては、「VPC の IP アドレス指定 (p. 124)」を参照してください。

VPC を作成したら、サブネットを作成できます。詳細については、「VPC にサブネットを作成する (p. 117)」を参照してください。

VPC を表示

VPC に関する詳細を表示できます。

コンソールを使用して VPC の詳細を表示するには

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、[VPCs] (VPC) を選択します。
3. VPC を選択した上で、[View Details] (詳細の表示) をクリックします。

コマンドラインツールを使用して VPC を記述するには

- describe-vpcs (AWS CLI)
- Get-EC2Vpc (AWS Tools for Windows PowerShell)

リージョン間ですべての VPC を表示するには

次の URL で Amazon EC2 グローバルビューコンソールを開きます。https://console.aws.amazon.com/ec2globalview/home

Amazon EC2 グローバルビューの使用方法については、Linux インスタンス用ユーザーガイドの 「リソースの一覧表示およびフィルタリング」を参照してください。

VPC にサブネットを作成する

VPC に新しいサブネットを追加するには、VPC の範囲からサブネットの IPv4 CIDR ブロックを指定する必要があります。サブネットが存在するアベイラビリティーゾーンを指定することができます。同じアベイラビリティーゾーン内に複数のサブネットを持つことができます。

IPv6 CIDR ブロックが VPC に関連付けられている場合は、オプションでサブネットに IPv6 CIDR ブロックを指定できます。

ローカルゾーンまたは Wavelength ゾーンにサブネットを作成するには、ゾーンを有効にする必要があります。Wavelength Zones を有効にする方法については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「ゾーンの有効化」を参照してください。

コンソールを使用してサブネットを VPC に追加するには

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、[サブネット]、[サブネットの作成] の順に選択します。
3. 必要に応じて、サブネットの詳細を指定して [作成] を選択します。
   - [Name tag]: 必要に応じてサブネットの名前を入力します。これにより、`Name` というキーと指定した値を含むタグが作成されます。
   - [VPC]: サブネットを作成する VPC を選択します。
   - [Availability Zone] (アベイラビリティーゾーン): 必要に応じてサブネットが存在するゾーンを選択するか、またはデフォルトの [No Preference] (指定なし) のままにして、AWS がアベイラビリティーゾーンを選択するようにします。

     リージョンとゾーンの詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「リージョンとゾーン」を参照してください。
   - [IPv4 CIDR ブロック]: サブネットの IPv4 CIDR ブロックを指定します。例: `10.0.1.0/24`。詳細については、「」を参照してくださいIPv4 用の VPC とサブネットのサイズ設定 (p. 109)
   - [IPv6 CIDR ブロック]: (オプション) IPv6 CIDR ブロックを VPC に関連付けている場合は、[Specify a custom IPv6 CIDR] を選択します。16 進法でキーペア値を選択するか、デフォルト値のままにします。
4. (オプション) 必要に応じて上記の手順を繰り返し、VPC でさらにサブネットを作成します。

または、コマンドラインツールを使用できます。

コマンドラインツールを使用してサブネットを追加するには

- create-subnet (AWS CLI)
- New-EC2Subnet (AWS Tools for Windows PowerShell)

サブネットを作成したら、次の手順を実行できます。

- ルーティングを設定します。サブネットをパブリックサブネットにするには、インターネットゲートウェイを VPC にアタッチする必要があります。詳細については、「」を参照してくださいインターネットゲートウェイの作成とアタッチ (p. 224) その後、カスタムルートテーブルを作成して、ルートをインターネットゲートウェイに追加できます。詳細については、「」を参照してくださいカスタムルートテーブルを作成する (p. 225) 他ルーティングのオプションについては、「VPC のルートテーブル (p. 294)」を参照してください。
- サブネットの設定を変更し、そのサブネットで起動されたすべてのインスタンスにパブリック IPv4 アドレス、IPv6 アドレス、またはその両方が割り当てられるようにします。詳細については、「」を参照してくださいサブネットの IP アドレス指定動作 (p. 127)
- 必要に応じて、セキュリティグループを作成または変更します。詳細については、「」を参照してくださいVPC のセキュリティグループ (p. 188)
- 必要に応じて、ネットワーク ACL を作成または変更します。詳細については、「」を参照してくださいネットワーク ACL (p. 200)
- サブネットを他のアカウントと共有します。詳細については、「」を参照してください??? (p. 145)

サブネットを表示する

サブネットに関する詳細を表示できます。

コンソールを使用してサブネットの詳細を表示するには

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、[Subnets] を選択します。
3. サブネットを選択して、[View Details] (詳細を表示) を選択します。

コマンドラインツールを使用してサブネットを記述するには

- describe-subnets (AWS CLI)
- Get-EC2Subnet (AWS Tools for Windows PowerShell)

リージョン間ですべてのサブネットを表示するには

次の URL で Amazon EC2 グローバルビューコンソールを開きます。https://console.aws.amazon.com/ec2globalview/home

Amazon EC2 グローバルビューの使用方法については、Linux インスタンス用ユーザーガイドの 「リソースの一覧表示およびフィルタリング」を参照してください。

VPC とセカンダリ IPv4 CIDR ブロックを関連付ける

VPC に別の IPv4 CIDR ブロックを追加できます。該当する制限 (p. 110)を必ず読んでください。

CIDR ブロックを関連付けたら、ステータスは `associating` になります。CIDR ブロックが `associated` 状態にあるときは、使用する準備ができています。

Amazon Virtual Private Cloud Console では、ページの上部にリクエストのステータスが表示されます。

コンソールを使用して CIDR ブロックを VPC に追加するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択し、その後 [Actions]、[Edit CIDRs] の順に選択します。
4. [Add IPv4 CIDR] を選択します。CIDR ブロックを入力します。例えば、`10.2.0.0/16`。[Save] を選択します。
5. [Close] を選択します。

コマンドラインツールを使用して CIDR ブロックを追加するには

- associate-vpc-cidr-block (AWS CLI)
- Register-EC2VpcCidrBlock (AWS Tools for Windows PowerShell)

必要な IPv4 CIDR ブロックを追加したら、サブネットを作成できます。詳細については、「」を参照してくださいVPC にサブネットを作成する (p. 117)

IPv6 CIDR ブロックと VPC の関連付け

IPv6 CIDR ブロックを既存の VPC と関連付けることができます。VPC には、それに関連付けられた既存の IPv6 CIDR ブロックがあってはなりません。

コンソールを使用して IPv6 CIDR ブロックを VPC に関連付けるには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。

3. VPC を選択し、その後 [Actions]、[Edit CIDRs] の順に選択します。
4. [Add IPv6 CIDR] を選択します。
5. [IPv6 CIDR block] で、次のいずれかを実行します。
   - [Amazon-provided IPv6 CIDR block] を選択して、Amazon の IPv6 アドレスプールから IPv6 CIDR ブロックをリクエストします。[Network border group] で、AWS が IP アドレスをアドバタイズするグループを選択します。
   - [IPv6 CIDR owned by me] を選択して、IPv6 アドレスプールから IPv6 CIDR ブロックを割り当てます。[Pool (プール)] で、IPv6 CIDR ブロックの割り当て元となる IPv6 アドレスプールを選択します。
6. [Select CIDR (CIDR の選択)] を選択します。
7. [Close] を選択します。

コマンドラインツールを使用して IPv6 CIDR ブロックを VPC に関連付けるには

- associate-vpc-cidr-block (AWS CLI)
- Register-EC2VpcCidrBlock (AWS Tools for Windows PowerShell)

IPv6 CIDR ブロックとサブネットの関連付け

IPv6 CIDR ブロックを VPC の既存のサブネットと関連付けることができます。サブネットには、それに関連付けられた既存の IPv6 CIDR ブロックがあってはなりません。

コンソールを使用して IPv6 CIDR ブロックをサブネットに関連付けるには

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、[Subnets] を選択します。
3. サブネットを選択後、[Subnet Actions]、[Edit IPv6 CIDRs] の順に選択します。
4. [Add IPv6 CIDR] を選択します。16 進法でサブネットのキーペアを指定し (例: 00)、チェックマークアイコンを選択してエントリを確認します。
5. [閉じる] を選択します。

または、コマンドラインツールを使用できます。

コマンドラインを使用して IPv6 CIDR ブロックをサブネットに関連付けるには

- associate-subnet-cidr-block (AWS CLI)
- Register-EC2SubnetCidrBlock (AWS Tools for Windows PowerShell)

サブネット内にインスタンスを起動する

サブネットを作成し、ルーティングを設定したら、Amazon EC2 コンソールを使用してサブネットにインスタンスを起動できます。

コンソールを使用してサブネットでインスタンスを起動するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ダッシュボードで、[Launch Instance] を選択します。
3. ウィザードの指示にしたがって操作します。AMI およびインスタンスタイプを選択し、[次の手順: インスタンスの詳細の設定] を選択します。

Note

インスタンスで IPv6 を介して通信する場合は、サポートされているインスタンスタイプを選択する必要があります。すべての現行世代のインスタンスタイプは、IPv6 アドレスをサポートしています。

4. [Configure Instance Details] ページの [Network] リストで必要な VPC を選択していることを確認し、インスタンスを起動するサブネットを選択します。このページの他のデフォルトの設定はそのままにして、[Next: Add Storage] を選択します。
5. ウィザードの次のページでは、インスタンスのストレージを設定し、タグを追加できます。[Configure Security Group] ページで、所有する既存のセキュリティグループから選択するか、ウィザードの指示にしたがって新しいセキュリティグループを作成します。完了したら、[Review and Launch] を選択します。
6. 設定を確認し、[Launch] を選択します。
7. 所有する既存のキーペアを選択するか、新しいキーペアを作成し、完了したら [Launch Instances] を選択します。

または、コマンドラインツールを使用できます。

コマンドラインツールを使用してサブネットでインスタンスを起動するには

- run-instances (AWS CLI)
- New-EC2Instance (AWS Tools for Windows PowerShell)

サブネットの削除

サブネットが不要になった場合には、それを削除することができます。サブネット内にインスタンスが存在する場合はそれをまず終了する必要があります。

コンソールを使用してサブネットを削除するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. サブネットのすべてのインスタンスを終了します。詳細については、EC2 ユーザーガイドの「インスタンスの終了」を参照してください。
3. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
4. ナビゲーションペインで、[Subnets] を選択します。
5. 削除するサブネットを選択して、[アクション]、[サブネットの削除] の順に選択します。
6. [サブネットの削除] ダイアログボックスで、[サブネットの削除] を選択します。

または、コマンドラインツールを使用できます。

コマンドラインツールを使用してサブネットを削除するには

- delete-subnet (AWS CLI)
- Remove-EC2Subnet (AWS Tools for Windows PowerShell)

VPC からの IPv4 CIDR ブロックの関連付けを解除する

VPC に複数の IPv4 CIDR ブロックが関連付けられている場合は、IPv4 CIDR ブロックと VPC の関連付けを解除できます。プライマリ IPv4 CIDR ブロックの関連付けを解除することはできません。CIDR ブロッ

ク全体の関連付けのみを解除できます。CIDR ブロックのサブセットまたは CIDR ブロックのマージされた範囲の関連付けを解除することはできません。最初に、CIDR ブロックのすべてのサブネットを削除する必要があります。

コンソールを使用して VPC から CIDR ブロックを削除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択し、[Actions]、[Edit CIDRs] の順に選択します。
4. [VPC IPv4 CIDRs] で、削除する CIDR ブロックの削除ボタン (十字) を選択します。
5. [閉じる] を選択します。

または、コマンドラインツールを使用できます。

コマンドラインツールを使用して VPC から IPv4 CIDR ブロックを削除するには

- disassociate-vpc-cidr-block (AWS CLI)
- Unregister-EC2VpcCidrBlock (AWS Tools for Windows PowerShell)

VPC またはサブネットからの IPv6 CIDR ブロックの関連付けを解除する

VPC またはサブネットで IPv6 が不要になっても、IPv4 リソースとの通信で VPC またはサブネットを引き続き使用する場合は、IPv6 CIDR ブロックの関連付けを解除できます。

IPv6 CIDR ブロックの関連付けを解除するには、まずサブネットのすべてのインスタンスに割り当てられている IPv6 アドレスの割り当てを解除する必要があります。詳細については、「」を参照してくださいインスタンスからの IPv6 アドレスの割り当て解除 (p. 131)

コンソールを使用してサブネットから IPv6 CIDR ブロックの関連付けを解除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Subnets] を選択します。
3. サブネットを選択し、[アクション]、[IPv6 CIDR の編集] の順に選択します。
4. クロスアイコンを選択して、サブネットの IPv6 CIDR ブロックを削除します。
5. [閉じる] を選択します。

コンソールを使用して VPC から IPv6 CIDR ブロックの関連付けを解除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択後、[Actions]、[Edit CIDRs] の順に選択します。
4. クロスアイコンを選択して、IPv6 CIDR ブロックを削除します。
5. [閉じる] を選択します。

> **Note**
>
> IPv6 CIDR ブロックの関連付けを解除しても、IPv6 ネットワーキングに対して設定したセキュリティグループルール、ネットワーク ACL ルール、ルートテーブルは自動的に削除されません。手動でこれらのルールまたはルートを変更するか、または削除する必要があります。

または、コマンドラインツールを使用できます。

コマンドラインツールを使用してサブネットから IPv6 CIDR ブロックの関連付けを解除するには

- disassociate-subnet-cidr-block (AWS CLI)
- Unregister-EC2SubnetCidrBlock (AWS Tools for Windows PowerShell)

コマンドラインツールを使用して VPC から IPv6 CIDR ブロックの関連付けを解除するには

- disassociate-vpc-cidr-block (AWS CLI)
- Unregister-EC2VpcCidrBlock (AWS Tools for Windows PowerShell)

VPC の削除

VPC コンソールを使用して VPC を削除するには、まず次のコンポーネントを終了または削除する必要があります。

- VPC 内のすべてのインスタンス – インスタンスを終了する方法については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「インスタンスの終了」を参照してください。
- VPC ピアリング接続
- インターフェイスエンドポイント
- NAT ゲートウェイ

VPC コンソールを使用して VPC を削除すると、次の VPC コンポーネントも削除されます。

- Subnets
- セキュリティグループ
- ネットワーク ACL
- ルートテーブル
- ゲートウェイエンドポイント
- インターネットゲートウェイ
- Egress-Only インターネットゲートウェイ
- DHCP オプション

AWS Site-to-Site VPN 接続がある場合、その接続、または VPN に関連する他のコンポーネント (カスタマーゲートウェイ、仮想プライベートゲートウェイなど) を削除する必要はありません。他の VPC でカスタマーゲートウェイを使用する予定がある場合は、AWS Site-to-Site VPN 接続とゲートウェイを保持することをお勧めします。そうしないと、新しい Site-to-Site VPN 接続を作成した後で、カスタマーゲートウェイデバイスを再度設定する必要があります。

コンソールを使用して VPC を削除するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. VPC のすべてのインスタンスを終了します。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「インスタンスの終了」を参照してください。
3. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
4. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
5. 削除する VPC を選択し、[Actions]、[Delete VPC] の順に選択します。
6. Site-to-Site VPN 接続がある場合は、それを削除するオプションを選択します。それ以外の場合、そのオプションはオフのままにしておきます。[VPC の削除] を選択します。

または、コマンドラインツールを使用できます。コマンドラインを使用して VPC を削除する場合は、最初にすべてのインスタンスを終了し、サブネット、カスタムセキュリティグループ、カスタムネットワーク ACL、カスタムルートテーブル、VPC ピア接続、エンドポイント、NAT ゲートウェイ、インターネットゲートウェイ、Egress-Only インターネットゲートウェイなど、関連リソースをすべて、削除またはデタッチする必要があります。

コマンドラインを使用して VPC を削除するには

- delete-vpc (AWS CLI)
- Remove-EC2Vpc (AWS Tools for Windows PowerShell)

VPC の IP アドレス指定

IP アドレスは、VPC のリソースの相互通信とインターネット上のリソースとの通信を有効にします。Amazon EC2 と Amazon VPC は、IPv4 と IPv6 のアドレス設定プロトコルをサポートします。インスタンスの IP アドレスの割り当ての詳細については、「Amazon EC2 インスタンスの IP アドレス指定」を参照してください。

デフォルトでは、Amazon EC2 と Amazon VPC は IPv4 アドレス設定プロトコルを使用します。VPC の作成時には IPv4 CIDR ブロック (プライベート IPv4 アドレスの範囲) を割り当てる必要があります。プライベート IPv4 アドレスには、インターネット経由では到達できません。インターネット経由でインスタンスに接続する、またはインスタンスとパブリックエンドポイントがある他の AWS のサービスとの間の通信を有効にするには、グローバルに一意なパブリック IPv4 アドレスをインスタンスに割り当てることができます。

オプションで IPv6 CIDR ブロックを VPC およびサブネットと関連付けて、そのブロックから VPC 内のリソースに IPv6 アドレスを割り当てることができます。IPv6 アドレスはパブリックであるため、インターネット経由で到達できます。

> Note
>
> インスタンスがインターネットと確実に通信できるようにするには、VPC にインターネットゲートウェイをアタッチする必要があります。詳細については、「」を参照してくださいインターネットゲートウェイ (p. 221)

VPC は、デュアルスタックモードで動作し、IPv4 または IPv6 あるいは両方を介して通信できます。IPv4 アドレスと IPv6 アドレスは互いに独立しています。VPC で IPv4 と IPv6 のルーティングとセキュリティを設定する必要があります。

次の表は、Amazon EC2 と Amazon VPC における IPv4 と IPv6 の違いをまとめたものです。

IPv4 および IPv6 の特徴と制限

IPv4	IPv6
形式は 32 ビットで、小数点以下 3 桁までの 4 つのグループです。	形式は 128 ビットで、4 桁の 16 進数の 8 つのグループです。
デフォルトですべての VPC で必要となるため、削除できません。	オプトインのみです。
VPC CIDR ブロックサイズは 16 ~ 28 となります。	VPC CIDR ブロックサイズは 56 に固定されています。
サブネット CIDR ブロックサイズは 16 ~ 28 です。	サブネット CIDR ブロックサイズは 64 に固定されています。

IPv4	IPv6
VPC では、プライベート IPv4 CIDR ブロックも選択できます。	両方をアマゾンの IPv6 アドレスのプールから VPC の IPv6 CIDR ブロックを選択します。独自の範囲を選択することはできません。
プライベート IP アドレスとパブリック IP アドレスの間には違いがあります。インターネットとの通信を有効にするには、ネットワークアドレス変換 (NAT) を介してパブリック IPv4 アドレスをプライマリプライベート IPv4 アドレスにマッピングします。	パブリック IP アドレスとプライベート IP アドレスに違いはありません。IPv6 アドレスはパブリックです。
すべてのインスタンスタイプでサポートされています。	現行世代のすべてのインスタンスタイプと、旧世代の C3、R3、I2 のインスタンスタイプでサポートされています。詳細については、「インスタンスタイプ」を参照してください。
EC2-Classic と、ClassicLink を介した VPC との EC2-Classic 接続でサポートされます。	EC2-Classic ではサポートされません。また、ClassicLink を介した VPC との EC2 Classic 接続でもサポートされません。
すべての AMI でサポートされます。	DHCPv6 用に設定された AMI で自動的にサポートされます。Amazon Linux バージョン 2016.09.0 以降と、Windows Server 2008 R2 以降は DHCPv6 用に設定されます。その他の AMI では、割り当てられた IPv6 アドレスを認識するようにインスタンスを手動で設定 (p. 139)する必要があります。
インスタンスは、そのプライベート IPv4 アドレスに対応するアマゾン提供のプライベート DNS ホスト名を受信します。必要に応じて、そのパブリック IPv4 または Elastic IP アドレスに対応するパブリック DNS ホスト名も受信します。	Amazon が提供する DNS ホスト名はサポートされません。
Elastic IPv4 アドレスがサポートされます。	Elastic IPv6 アドレスはサポートされません。
カスタマーゲートウェイ、仮想プライベートゲートウェイ、NAT デバイス、および VPC エンドポイントでサポートされています。	カスタマーゲートウェイ、仮想プライベートゲートウェイ、NAT デバイス、および VPC エンドポイントではサポートされていません。

AWS Direct Connect 接続への仮想プライベートゲートウェイ経由の IPv6 トラフィックをサポートしています。詳細については、AWS Direct Connect ユーザーガイドを参照してください。

プライベート IPv4 アドレス

プライベート IPv4 アドレス (このトピックではプライベート IP アドレスと呼ぶ) は、インターネット経由では到達できず、VPC のインスタンス間の通信で使用できます。VPC でインスタンスを起動すると、サブネットの IPv4 アドレス範囲内のプライマリプライベート IP アドレスがインスタンスのデフォルトのネットワークインターフェイス (eth0) に割り当てられます。また、各インスタンスには、インスタンスのプライベート IP アドレスに解決されるプライベート (内部) DNS ホスト名が割り当てられます。プライマリプライベート IP アドレスを指定しない場合、サブネットの範囲内で使用可能な IP アドレスが選択されます。ネットワークインターフェイスの詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Elastic Network Interface」を参照してください。

VPC で実行されているインスタンスに追加のプライベート IP アドレス (セカンダリプライベート IP アドレスと呼ばれる) を割り当てることができます。プライマリプライベート IP アドレスとは異なり、セカン

ダリプライベート IP アドレスはあるネットワークインターフェイスから別のネットワークインターフェイスへ割り当て直すことができます。プライベート IP アドレスは、インスタンスが停止して再起動するとネットワークインターフェイスに関連付けられたままになり、インスタンスが終了すると解放されます。プライマリ IP アドレスとセカンダリ IP アドレスの詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「複数の IP アドレス」を参照してください。

Note

プライベート IP アドレスは VPC の IPv4 CIDR 範囲内にある IP アドレスです。VPC のほとんどの IP アドレス範囲は、RFC 1918 で規定されているプライベート (パブリックにルーティングできない) IP アドレス範囲に入りますが、パブリックにルーティングできる CIDR ブロックを VPC に使用することはできます。VPC の IP アドレス範囲に関係なく、パブリックにルーティング可能な CIDR ブロックなど VPC の CIDR ブロックからのインターネットへの直接アクセスはサポートされていません。ゲートウェイを経由するインターネットアクセスをセットアップする必要があります。たとえば、インターネットゲートウェイ、仮想プライベートゲートウェイ、AWS Site-to-Site VPN 接続、または AWS Direct Connect をセットアップします。

パブリック IPv4 アドレス

サブネットで作成されたネットワークインターフェイスがパブリック IPv4 アドレス (このトピックではパブリック IP アドレスと呼ばれる) を自動的に受信するかどうかを判断する属性が、すべてのサブネットにあります。したがって、この属性が有効になっているサブネットに対してインスタンスを起動すると、パブリック IP アドレスがそのインスタンス用に作成されたプライマリネットワークインターフェイス (eth0) に割り当てられます。パブリック IP アドレスは、ネットワークアドレス変換 (NAT) によって、プライマリプライベート IP アドレスにマッピングされます。

VPC のインスタンスがパブリック IP アドレスを割り当てられるかどうかを制御するには、以下の方法を使用します。

- サブネットのパブリック IP アドレス属性を変更する。詳細については、「」を参照してくださいサブネットのパブリック IPv4 アドレス属性を変更する (p. 128)
- インスタンスの起動時のパブリック IP アドレス割り当てを有効または無効にする。この設定によってサブネットのパブリック IP アドレス割り当て属性は上書きされます。詳細については、「」を参照してくださいインスタンス起動時のパブリック IPv4 アドレスの割り当て (p. 129)

パブリック IP アドレスは、Amazon のパブリック IP アドレスプールにあるアドレスです。そのアドレスはお客様のアカウントとは関連付けられません。パブリック IP アドレスとインスタンスとの関連付けを解除すると、そのアドレスは解放されてプールに戻り、それ以降お客様はそのアドレスを使用できなくなります。パブリック IP アドレスの関連付けと関連付け解除は手動で実行できません。ただし、特定の場合に、こちらでパブリック IP アドレスをお客様のインスタンスから解放したり、新しいアドレスをそのインスタンスに割り当てます。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「パブリック IP アドレス」を参照してください。

状況に応じてインスタンスに割り当てたりインスタンスから削除したりできる固定パブリック IP アドレスをお客様のアカウントに割り当てる必要がある場合は、Elastic IP アドレスを使用します。詳細については、「」を参照してくださいElastic IP アドレス (p. 288)

VPC で DNS ホスト名のサポートを有効にしている場合は、パブリック IP アドレスまたは Elastic IP アドレスを受信するインスタンスには、それぞれパブリック DNS ホスト名が付与されます。パブリック DNS ホスト名を解決すると、インスタンスのパブリック IP アドレス (インスタンスのネットワークの外部の場合) およびインスタンスのプライベート IP アドレス (インスタンスのネットワーク内からの場合) となります。詳細については、「」を参照してくださいVPC の DNS サポート (p. 271)

IPv6 アドレス

オプションで、IPv6 CIDR ブロックを VPC とサブネットに関連付けることができます。詳細については、以下のトピックを参照してください。

- IPv6 CIDR ブロックと VPC の関連付け (p. 119)
- IPv6 CIDR ブロックとサブネットの関連付け (p. 120)

IPv6 CIDR ブロックが VPC およびサブネットと関連付けられていて、以下のいずれかに該当する場合、VPC 内のインスタンスには IPv6 アドレスが割り当てられます。

- サブネットは、起動時に IPv6 アドレスをインスタンスのプライマリネットワークインターフェイスに自動的に割り当てるよう設定されます。
- 起動時に IPv6 アドレスをインスタンスに手動で割り当てます。
- 起動後に IPv6 アドレスをインスタンスに割り当てます。
- 起動後に IPv6 アドレスを同じサブネットのネットワークインターフェイスに割り当て、そのネットワークインターフェイスをインスタンスにアタッチする。

起動時にインスタンスに IPv6 アドレスが割り当てられると、そのアドレスはインスタンスのプライマリネットワークインターフェイス (eth0) と関連付けられます。IPv6 アドレスとプライマリネットワークインターフェイスの関連付けは解除できます。インスタンスの IPv6 DNS ホスト名はサポートされていません。

IPv6 アドレスは、インスタンスの停止および開始時には保持され、インスタンスの終了時に解放されます。IPv6 アドレスは、別のネットワークインターフェイスに割り当てられている間は再割り当てできません。最初に割り当てを解除する必要があります。

追加の IPv6 アドレスをインスタンスに割り当てるには、インスタンスにアタッチされたネットワークインターフェイスにアドレスを割り当てます。ネットワークインターフェイスに割り当てることができる IPv6 アドレスの数と、インスタンスにアタッチできるネットワークインターフェイスの数は、インスタンスタイプごとに異なります。詳細については、Amazon EC2 ユーザーガイドの「各インスタンスタイプのネットワークインターフェイスあたりの IP アドレス」を参照してください。

IPv6 アドレスはグローバルに一意であるため、インターネット経由で到達できます。インスタンスのサブネットのルーティングを制御するか、セキュリティグループとネットワーク ACL ルールを使用することで、IPv6 アドレスを介してインスタンスに接続できるかどうかを制御できます。詳細については、「」を参照してくださいAmazon VPC でのインターネットワークトラフィックのプライバシー (p. 165)

予約済み IPv6 アドレスの範囲については、「IANA IPv6 Special-Purpose Address Registry」と「RFC4291」を参照してください。

サブネットの IP アドレス指定動作

すべてのサブネットに、そのサブネットで作成したネットワークインターフェイスをパブリック IPv4 アドレス (該当する場合は IPv6 アドレス) に割り当てるかどうかを決定する、変更可能な属性があります。これには、サブネットでインスタンスを起動したときにインスタンス用に作成されるプライマリネットワークインターフェイス (eth0) が含まれます。

サブネットの属性に関係なく、特定のインスタンスの起動時の設定によって上書きできます。詳細については、「インスタンス起動時のパブリック IPv4 アドレスの割り当て (p. 129)」および「インスタンス起動時の IPv6 アドレスの割り当て (p. 129)」を参照してください。

お客様の IP アドレスを使用する

独自のパブリック IPv4 アドレス範囲または IPv6 アドレス範囲の一部またはすべてを AWS アカウントに持ち込むことができます。引き続きアドレス範囲を所有できますが、デフォルトで AWS はこれをインターネット上でアドバタイズします。アドレス範囲を AWS に設定すると、そのアドレス範囲はアドレスプールとしてアカウントに表示されます。IPv4 アドレスプールから Elastic IP アドレスを作成し、IPv6 アドレスプールの IPv6 CIDR ブロックを VPC に関連付けることができます。

詳細については、 Linux インスタンス用 Amazon EC2 ユーザーガイドの「独自の IP アドレスを使用する (BYOIP)」をご参照ください。

IP アドレスの操作

サブネットの IP アドレス指定動作、起動中のパブリック IPv4 アドレスのインスタンスへの割り当て、インスタンスとの IPv6 アドレスの割り当てと割り当て解除を変更できます。

タスク

- サブネットのパブリック IPv4 アドレス属性を変更する (p. 128)
- サブネットのパブリック IPv6 アドレス属性を変更する (p. 128)
- インスタンス起動時のパブリック IPv4 アドレスの割り当て (p. 129)
- インスタンス起動時の IPv6 アドレスの割り当て (p. 129)
- インスタンスへの IPv6 アドレスの割り当て (p. 130)
- インスタンスからの IPv6 アドレスの割り当て解除 (p. 131)
- API とコマンドの概要 (p. 131)

サブネットのパブリック IPv4 アドレス属性を変更する

デフォルトでは、デフォルト以外のサブネットでは IPv4 パブリックアドレス属性が `false` に設定されており、デフォルトサブネットではこの属性が `true` に設定されています。例外は、Amazon EC2 インスタンス起動ウィザードによって作成されるデフォルト以外のサブネットです。このウィザードが、属性を `true` に設定します。Amazon VPC コンソールを使用してこの属性を変更できます。

サブネットのパブリック IPv4 のアドレス動作を変更するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Subnets] を選択します。
3. サブネットを選択し、[Subnet Actions] を選択して、次に [Modify auto-assign IP settings] を選択します。
4. [Enable auto-assign public IPv4 address] チェックボックスをオンにした場合、選択されたサブネット内で起動されるすべてのインスタンスに対してパブリック IPv4 アドレスがリクエストされます。必要に応じてチェックボックスをオンまたはオフにして、[Save] を選択します。

サブネットのパブリック IPv6 アドレス属性を変更する

デフォルトでは、すべてのサブネットで IPv6 アドレス属性が `false` に設定されています。Amazon VPC コンソールを使用してこの属性を変更できます。サブネットで IPv6 アドレス属性を有効にした場合、そのサブネットで作成されたネットワークインターフェイスは、サブネットの範囲から IPv6 アドレスを受け取ります。サブネットに起動されたインスタンスは、プライマリネットワークインターフェイスで IPv6 アドレスを受け取ります。

サブネットには関連付けられた IPv6 CIDR ブロックが必要です。

> Note
>
> サブネットに対して IPv6 アドレス機能を有効にすると、ネットワークインターフェイスまたはインスタンスのみが IPv6 アドレスを受け取ります (バージョン `2016-11-15` 以降の Amazon EC2 API を使用して作成された場合)。Amazon EC2 コンソールは最新の API バージョンを使用します。

サブネットのパブリック IPv6 アドレスの動作を変更するには

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

2. ナビゲーションペインで、[Subnets] を選択します。
3. サブネットを選択し、[Subnet Actions] を選択して、次に [Modify auto-assign IP settings] を選択します。
4. [Enable auto-assign IPv6 address] チェックボックスをオンにした場合、選択されたサブネット内で作成されるすべてのネットワークインターフェイスに対して IPv6 アドレスがリクエストされます。必要に応じてチェックボックスをオンまたはオフにして、[Save] を選択します。

インスタンス起動時のパブリック IPv4 アドレスの割り当て

ただし、デフォルトサブネットまたはデフォルト以外のサブネット内のインスタンスが起動中にパブリック IPv4 アドレスを割り当てられるかどうかを制御できます。

> Important
>
> 起動後に、インスタンスからパブリック IPv4 アドレスの割り当てを手動で解除することはできません。ただし、特定の場合に、アドレスが自動的に解放され、その後再利用できなくなります。自由に関連付けと関連付け解除を実行できる固定パブリック IP アドレスが必要な場合は、起動後に Elastic IP アドレスをインスタンスに関連付けます。詳細については、「」を参照してくださいElastic IP アドレス (p. 288)

起動時にパブリック IPv4 アドレスをインスタンスに割り当てるには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. [インスタンスの作成] を選択します。
3. AMI およびインスタンスタイプを選択し、[Next: Configure Instance Details] を選択します。
4. [Configure Instance Details] ページの [Network] で VPC を選択します。[Auto-assign Public IP] リストが表示されます。[Enable] または [Disable] を選択して、サブネットのデフォルトの設定をオーバーライドします。
5. ウィザードの後続ページに表示されるステップにしたがって、インスタンスのセットアップを最後まで実行します。最終ページの [Review Instance Launch] で、設定内容を確認します。[Launch] を選択してキーペアを選択し、インスタンスを起動します。
6. [Instances] ページで、新しいインスタンスを選択し、そのパブリック IP アドレスを、詳細ペインの [IPv4 Public IP] フィールドで確認します。

> Note
>
> パブリック IPv4 アドレスは、コンソールのネットワークインターフェイスのプロパティとして表示されますが、NAT によってプライマリプライベート IPv4 アドレスにマッピングされます。したがって、インスタンスのネットワークインターフェイスのプロパティを調べる場合、たとえば Windows インスタンスに対して `ipconfig` を使用しても、Linux インスタンスに対して `ifconfig` を使用しても、パブリック IP アドレスは表示されません。インスタンス内からインスタンスのパブリック IP アドレスを決定するには、インスタンスのメタデータを使用できます。詳細については、「インスタンスメタデータとユーザーデータ」を参照してください。

この機能は起動中のみ使用できます。ただし、起動時にパブリック IPv4 アドレスをインスタンスに割り当てるかどうかにかかわらず、起動後に Elastic IP アドレスをインスタンスに関連付けることができます。詳細については、「」を参照してくださいElastic IP アドレス (p. 288)

インスタンス起動時の IPv6 アドレスの割り当て

起動時に IPv6 アドレスをインスタンスに自動で割り当てます。これを行うには、関連付けられた IPv6 CIDR ブロック (p. 119)を持つ VPC とサブネットに対してインスタンスを起動する必要があります。IPv6 アドレスはサブネットの範囲から割り当てられ、プライマリネットワークインターフェイス (eth0) に割り当てられます。

起動時に IPv6 アドレスをインスタンスに自動的に割り当てるには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. [インスタンスの作成] を選択します。
3. AMI およびインスタンスタイプを選択し、[次の手順: インスタンスの詳細の設定] を選択します。

   Note

   IPv6 アドレスをサポートするインスタンスタイプを選択します。
4. [Configure Instance Details] ページで、[Network] から VPC を選択し、[Subnet] からサブネットを選択します。[Auto-assign IPv6 IP] で、[Enable] を選択します。
5. ウィザードの残りの手順に従ってインスタンスを起動します。

または、起動時にインスタンスのサブネットの範囲から固有の IPv6 アドレスを割り当てる場合は、インスタンスのプライマリネットワークインターフェイスにアドレスを割り当てることができます。

起動時にインスタンスに固有の IPv6 アドレスを割り当てるには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. [インスタンスの作成] を選択します。
3. AMI およびインスタンスタイプを選択し、[次の手順: インスタンスの詳細の設定] を選択します。

   Note

   IPv6 アドレスをサポートするインスタンスタイプを選択します。
4. [Configure Instance Details] ページで、[Network] から VPC を選択し、[Subnet] からサブネットを選択します。
5. 「ネットワークインターフェイス」のセクションを参照してください。eth0 ネットワークインターフェイスでは、[IPv6 IPs] で [Add IP] を選択します。
6. サブネットの範囲から IPv6 アドレスを入力します。
7. ウィザードの残りの手順に従ってインスタンスを起動します。

起動時にインスタンスに複数の IPv6 アドレスを割り当てることの詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「複数の IPv6 アドレスの使用」を参照してください。

インスタンスへの IPv6 アドレスの割り当て

インスタンスが VPC 内にあり、サブネットに関連付けられた IPv6 CIDR ブロック (p. 119)がある場合は、Amazon EC2 コンソールを使用して、使用するサブネットの範囲から IPv6 アドレスをインスタンスに割り当てることができます。

IPv6 アドレスをインスタンスに関連付けるには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで [Instances] を選択し、インスタンスを選択します。
3. [アクション]、[ネットワーキング]、[IP アドレスの管理] の順に選択します。
4. [IPv6 Addresses] で、[Assign new IP] を選択します。サブネットの範囲から IPv6 アドレスを指定するか、[Auto-assign] を使って IPv6 アドレスを自動的に選択することができます。
5. [Save] を選択します。

また、IPv6 アドレスをネットワークインターフェイスに割り当てることができます。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Elastic Network Interfaces」トピックの「IPv6 アドレスを割り当てる」を参照してください。

インスタンスからの IPv6 アドレスの割り当て解除

インスタンスで IPv6 アドレスが不要になった場合は、Amazon EC2 コンソールを使用してインスタンスから関連付けを解除できます。

インスタンスから IPv6 アドレスの関連付けを解除するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで [Instances] を選択し、インスタンスを選択します。
3. [アクション]、[ネットワーキング]、[IP アドレスの管理] の順に選択します。
4. [IPv6 Addresses] で、IPv6 アドレスに対して [Unassign] を選択します。
5. [Save] を選択します。

または、ネットワークインターフェイスから IPv6 アドレスの関連付けを解除することができます。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Elastic Network Interfaces」トピックの「IPv6 アドレスを割り当て解除する」を参照してください。

API とコマンドの概要

このページで説明しているタスクは、コマンドラインまたは API を使用して実行できます。コマンドラインインターフェイスの詳細および利用できる API の一覧については、「Amazon VPC にアクセスする (p. 1)」を参照してください。

起動時にパブリック IPv4 アドレスを割り当てる

- run-instances コマンド (AWS CLI) で `--associate-public-ip-address` または `--no-associate-public-ip-address` オプションを使用します。
- New-EC2Instance コマンド (AWS Tools for Windows PowerShell) で `-AssociatePublicIp` パラメータを使用します。

起動時に IPv6 アドレスを割り当てる

- run-instances コマンド (AWS CLI) で `--ipv6-addresses` オプションを使用します。
- New-EC2Instance コマンド (AWS Tools for Windows PowerShell) で `-Ipv6Addresses` パラメータを使用します。

サブネットのパブリック IP アドレス動作を変更する

- modify-subnet-attribute (AWS CLI)
- Edit-EC2SubnetAttribute (AWS Tools for Windows PowerShell)

IPv6 アドレスをネットワークインターフェイスに割り当てる

- assign-ipv6-addresses (AWS CLI)
- Register-EC2Ipv6AddressList (AWS Tools for Windows PowerShell)

ネットワークインターフェイスから IPv6 アドレスの割り当てを解除する

- unassign-ipv6-addresses (AWS CLI)
- Unregister-EC2Ipv6AddressList (AWS Tools for Windows PowerShell)

IPv6に移行する

既存の VPC が IPv4 のみに対応しており、サブネット内のリソースが IPv4 のみを使用するように設定されている場合は、既存の VPC とリソースに対して IPv6 を有効化できます。VPC は、デュアルスタックモードで動作します。IPv4 または IPv6 あるいは両方を経由して通信できます。IPv4 と IPv6 は、互いに独立して通信されます。

VPC と サブネットの IPv4 サポートを無効にすることはできません。これが、Amazon VPC と Amazon EC2 の IP アドレスシステムのデフォルト値です。

> Note
>
> この情報は、パブリックサブネットとプライベートサブネットを持つ既存の VPC があることを前提としています。IPv6 で使用する新しい VPC のセットアップについては、「the section called "IPv6 の概要" (p. 22)」を参照してください。

次の表は、VPC とサブネットで IPv6 の使用を有効にするためのステップの概要を示しています。

ステップ	コメント
ステップ 1: IPv6 CIDR ブロックを VPC およびサブネットと関連付ける (p. 135)	Amazon が提供する IPv6 CIDR ブロックを VPC およびサブネットと関連付けます。
ステップ 2: ルートテーブルを更新する (p. 136)	IPv6 トラフィックがルーティングされるようにルートテーブルを更新します。パブリックサブネットの場合、サブネットからインターネットゲートウェイに IPv6 トラフィックをすべてルーティングするルートを作成します。プライベートサブネットの場合、サブネットから Egress-only インターネットゲートウェイにインターネット経由の IPv6 トラフィックをすべてルーティングするルートを作成します。
ステップ 3: セキュリティグループルールを更新する (p. 136)	IPv6 アドレスのルールを含めて、セキュリティグループルールを更新します。これにより、IPv6 トラフィックはインスタンスに出入りできるようになります。カスタムネットワーク ACL ルールを作成して、サブネットに出入りするトラフィックの流れを制御している場合は、IPv6 トラフィックのルールを含める必要があります。
ステップ 4: インスタンスタイプを変更する (p. 137)	インスタンスタイプが IPv6 をサポートしていない場合は、インスタンスタイプを変更します。
ステップ 5: IPv6 アドレスをインスタンスに割り当てる (p. 138)	サブネットの IPv6 アドレスの範囲からインスタンスに IPv6 アドレスを割り当てます。
ステップ 6: (オプション) インスタンスに IPv6 を設定する (p. 139)	DHCPv6 を使用するように設定されていない AMI からインスタンスが起動された場合は、インスタンスに割り当てられている IPv6 アドレスが認識されるように手動でインスタンスを設定する必要があります。

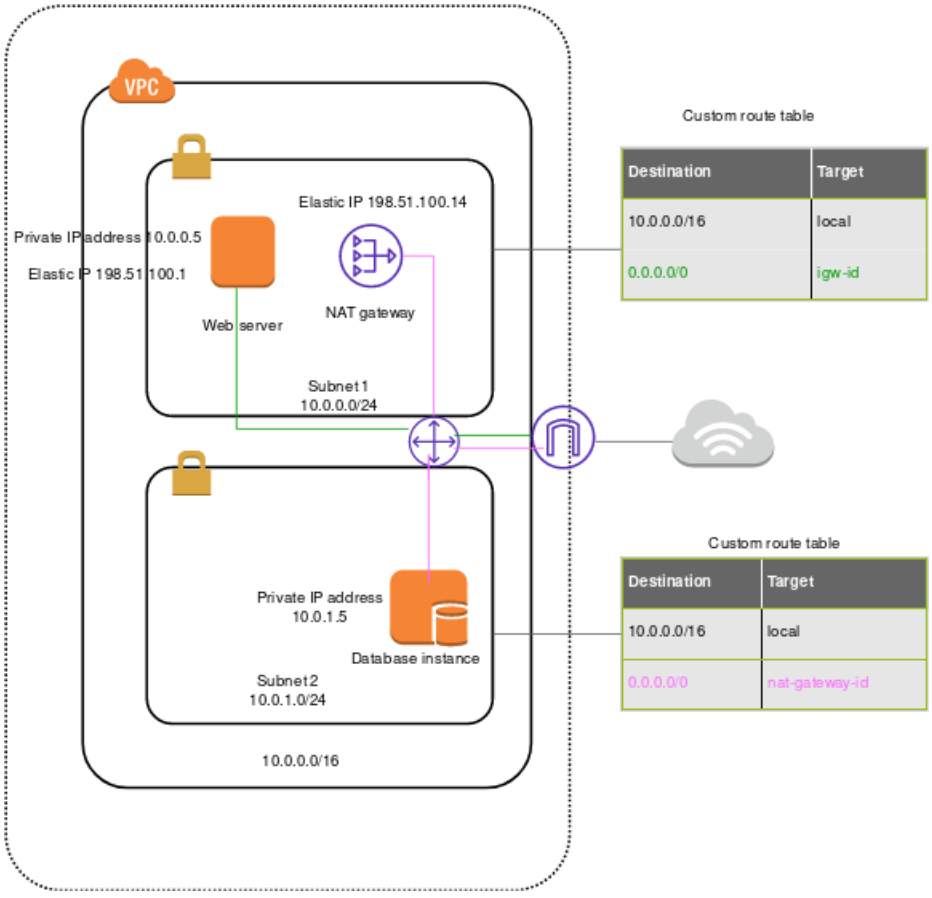
IPv6 の使用に移行する前に、Amazon VPC に対する IPv6 アドレス指定の機能に関する「IPv4 および IPv6 の特徴と制限 (p. 124)」を参照したことを確認します。

目次

- ステップ 1: IPv6 CIDR ブロックを VPC およびサブネットと関連付ける (p. 135)
- ステップ 2: ルートテーブルを更新する (p. 136)
- ステップ 3: セキュリティグループルールを更新する (p. 136)
- ステップ 4: インスタンスタイプを変更する (p. 137)
- ステップ 5: IPv6 アドレスをインスタンスに割り当てる (p. 138)
- ステップ 6: (オプション) インスタンスに IPv6 を設定する (p. 139)

例: パブリックサブネットとプライベートサブネットを持つ VPC 内で IPv6 を有効化する

この例では、VPC にパブリックサブネットとプライベートサブネットがあります。プライベートサブネットには、VPC 内の NAT ゲートウェイを経由してインターネットとアウトバウンド通信するデータベースインスタンスがあります。パブリックサブネットには、インターネットゲートウェイ経由でインターネットにアクセスするパブリックのウェブサーバーがあります。次の図は、VPC のアーキテクチャを示しています。



ウェブサーバーのセキュリティグループ (`sg-11aa22bb11aa22bb1`) には、次のインバウンドルールがあります。

タイプ	プロトコル	ポート範囲	送信元	コメント
すべてのトラフィック	すべて	すべて	sg-33cc44dd33cc44dd3	sg-33cc44dd33cc44dd3 (データベースイ

タイプ	プロトコル	ポート範囲	送信元	コメント
				ンスタンス) に関連付けられたインスタンスからのトラフィックのすべてのインバウンドアクセスを許可します。
HTTP	TCP	80	0.0.0.0/0	HTTP を介したインターネットからのインバウンドトラフィックを許可します。
HTTPS	TCP	443	0.0.0.0/0	HTTPS を介したインターネットからのインバウンドトラフィックを許可します。
SSH	TCP	22	203.0.113.123/32	ローカルコンピュータからのインバウンド SSH アクセスを許可します (例: インスタンスに接続して管理タスクを実行する必要がある場合)。

データベースインスタンスのセキュリティグループ (`sg-33cc44dd33cc44dd3`) には、次のインバウンドルールがあります。

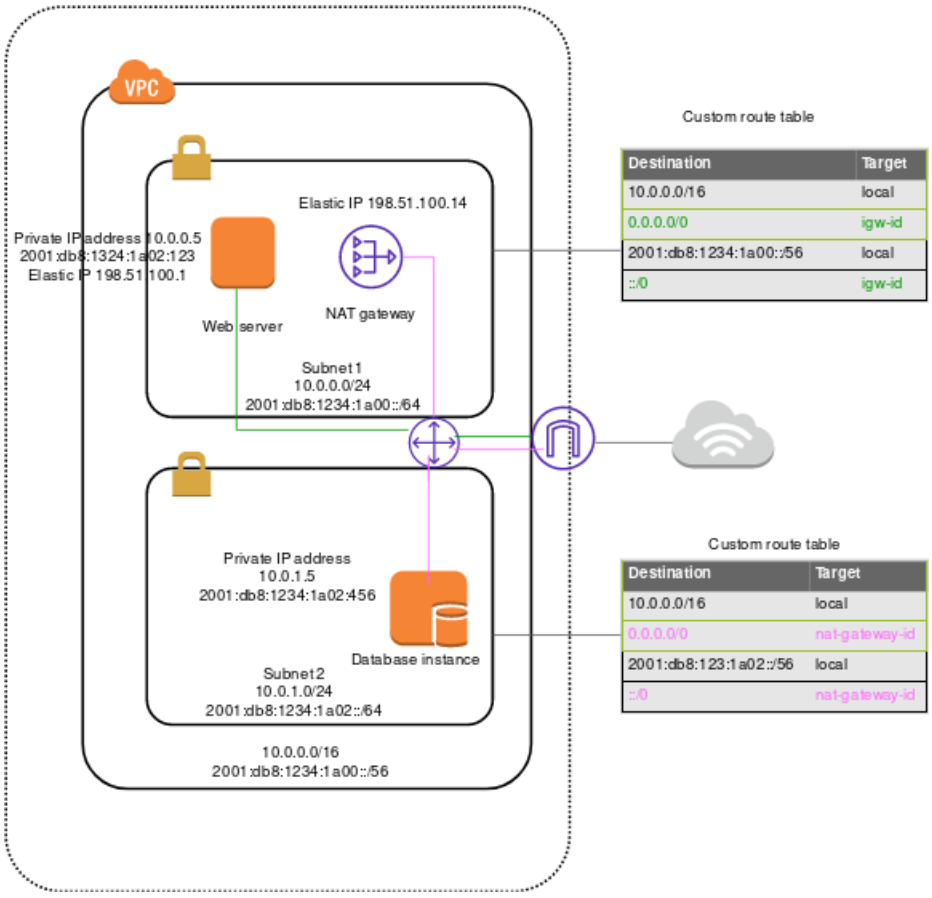
タイプ	プロトコル	ポート範囲	送信元	コメント
MySQL	TCP	3306	sg-11aa22bb11aa22bb1	sg-11aa22bb11aa22bb1 (ウェブサーバーインスタンス) に関連付けられたインスタンスからの MySQL トラフィックのインバウンドアクセスを許可します。

どちらのセキュリティグループにも、すべてのアウトバウンド IPv4 トラフィックを許可するアウトバウンドルールがデフォルトで設定されていますが、それ以外のアウトバウンドルールを設定することはできません。

ウェブサーバーは、`t2.medium` インスタンスタイプです。データベースサーバーは、`m3.large` です。

VPC とリソースを IPv6 用に有効化し、デュアルスタックモードで操作します。つまり、VPC のリソースとインターネット経由のリソースの間で、IPv6 アドレスと IPv4 アドレスの両方を使用します。

これらの手順が完了すると、VPC は次のように設定されます。



ステップ 1: IPv6 CIDR ブロックを VPC およびサブネットと関連付ける

IPv6 CIDR ブロックを VPC と関連付けたら、範囲内の `/64` の CIDR ブロックを各サブネットと関連付けます。

IPv6 CIDR ブロックを VPC と関連付けるには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択後、[Actions]、[Edit CIDRs] の順に選択します。
4. [Add IPv6 CIDR (IPv6 CIDR の追加)] を選択し、次のいずれかのオプションを選択して、[Add IPv6 CIDR (CIDR の選択)] を選択します。

   - [Amazon-provided IPv6 CIDR block (Amazon が提供する IPv6 CIDR ブロック)]: Amazon の IPv6 アドレスプールから IPv6 CIDR ブロックをリクエストします。[Network Border Group] (ネットワーク境界グループ) で、AWS による IP アドレスのアドバタイズ元となるグループを選択します。
   - [IPv6 CIDR owned by me (自分が所有する IPv6 CIDR)]: (BYOIP) IPv6 アドレスプールから IPv6 CIDR ブロックを割り当てます。[Pool (プール)] で、IPv6 CIDR ブロックの割り当て元となる IPv6 アドレスプールを選択します。

IPv6 CIDR ブロックをサブネットと関連付けるには

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、[Subnets] を選択します。
3. サブネットを選択後、[Subnet Actions]、[Edit IPv6 CIDRs] の順に選択します。
4. [Add IPv6 CIDR] を選択します。16 進法でサブネットのキーペアを指定し (例: `00`)、チェックマークアイコンを選択してエントリを確認します。
5. [閉じる] を選択します。VPC 内の他のサブネットにも同様に、上記ステップを繰り返します。

詳細については、「」を参照してくださいIPv6 用の VPC とサブネットのサイズ設定 (p. 114)

ステップ 2: ルートテーブルを更新する

パブリックサブネットの場合、ルートテーブルを更新して、IPv6 トラフィック用にインターネットゲートウェイを使用するように、インスタンス (ウェブサーバーなど) を有効にする必要があります。

プライベートサブネットの場合、ルートテーブルを更新して、IPv6 トラフィック用に Egress-only インターネットゲートウェイを使用するように、インスタンス (データベースインスタンスなど) を有効にする必要があります。

パブリックサブネット用にルートテーブルを更新するには

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、[Route Tables] を選択後、パブリックサブネットに関連付けられたルートテーブルを選択します。
3. [Routes] タブで、[Edit routes] を選択します。
4. [Add Rule (ルートの追加)] を選択します。[Destination] (送信元) で `::/0` を指定し、[Target] (ターゲット) でインターネットゲートウェイ ID を選択したら [Save changes] (変更の保存) を選択します。

プライベートサブネット用にルートテーブルを更新するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. プライベートサブネットで NAT デバイスを使用している場合、IPv6 トラフィックはサポートされません。その代わり、IPv6 経由のインターネットへのアウトバウンド通信を有効にし、インバウンド通信を無効にする場合は、プライベートサブネット用に Egress-only インターネットゲートウェイを作成します。Egress-only インターネットゲートウェイは、IPv6 トラフィックのみサポートしています。詳細については、「」を参照してくださいEgress-Only インターネットゲートウェイ (p. 228)
3. ナビゲーションペインで、[Route Tables] を選択後、プライベートサブネットに関連付けられたルートテーブルを選択します。
4. [Routes] タブで、[Edit routes] を選択します。
5. [Add Rule (ルートの追加)] を選択します。[送信先] で `::/0` を指定します。[Target] (ターゲット) で egress-only インターネットゲートウェイの ID を選択し、[Save changes] (変更の保存) を選択します。

詳細については、「」を参照してくださいルーティングオプションの例 (p. 303)

ステップ 3: セキュリティグループルールを更新する

インスタンスが IPv6 経由でトラフィックを送受信できるようにするには、IPv6 アドレスのルールを含めるようにセキュリティグループルールを更新する必要があります。

たとえば、上記の例では、ウェブサーバーのセキュリティグループ (`sg-11aa22bb11aa22bb1`) を更新し、IPv6 アドレスからのインバウンド HTTP、HTTPS、および SSH アクセスを許可するルールを追加できます。データベースのセキュリティグループのインバウンドルールを変更する必要はありません。

`sg-11aa22bb11aa22bb1` からの通信をすべて許可するルールには、IPv6 通信がデフォルトで含まれています。

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 [Security Groups] を選択後、ウェブサーバーのセキュリティグループを選択します。
3. [Inbound Rules] タブで、[Edit] を選択します。
4. ルールごとに、[別のルールの追加] を選択し、終了したら [保存] を選択します。たとえば、IPv6 経由ですべての HTTP トラフィックを許可するルールを追加するには、[タイプ] で [HTTP] を選択し、[ソース] に「`::/0`」と入力します。

デフォルトでは、IPv6 CIDR ブロックを VPC と関連付けると、すべての IPv6 トラフィックを許可するアウトバウンドルールがセキュリティグループに自動的に追加されます。ただし、セキュリティグループの元のルールを変更する場合、このアウトバウンドルールは自動的に追加されません。そのため、IPv6 トラフィック用に同等のアウトバウンドルールを追加する必要があります。詳細については、「」を参照してくださいVPC のセキュリティグループ (p. 188)

ネットワーク ACL ルールを更新する

IPv6 CIDR ブロックと VPC を関連付けると、IPv6 トラフィックを許可するように、デフォルトのネットワーク ACL にルールが自動的に追加されます。ただし、デフォルトのルールを変更していない場合に限ります。デフォルトのネットワーク ACL を変更した場合、またはサブネット間のトラフィックの流れを制御するルールを使用してカスタムネットワーク ACL を作成した場合は、IPv6 トラフィック用のルールを手動で追加する必要があります。詳細については、「」を参照してくださいネットワーク ACL (p. 200)

ステップ 4: インスタンスタイプを変更する

すべての現行世代のインスタンスタイプは、IPv6 をサポートしています。詳細については、「インスタンスタイプ」を参照してください。

インスタンスタイプが IPv6 をサポートしていない場合は、サポートされるインスタンスタイプに合わせて、インスタンスのサイズを変更する必要があります。上記の例では、データベースインスタンスは `m3.large` インスタンスタイプで、IPv6 をサポートしていません。サポートされるインスタンスタイプ (例: `m4.large`) にインスタンスタイプのサイズを変更する必要があります。

インスタンスのサイズを変更するには、互換性による制約に注意してください。詳細については、Linux インスタンス用の Amazon EC2 ユーザーガイドの「インスタンスのサイズ変更の互換性」を参照してください。このシナリオでは、データベースインスタンスが、HVM 仮想化を使用する AMI から起動された場合は、次の手順で `m4.large` インスタンスタイプにサイズ変更できます。

> **Important**
>
> インスタンスのサイズを変更するには、インスタンスを停止する必要があります。インスタンスを停止、起動を行うと、インスタンスのパブリック IPv4 アドレスは変更されます (ある場合)。インスタンスストアボリュームにデータが保存されている場合、データは消去されます。

インスタンスのサイズを変更するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[Instances] を選択後、データベースインスタンスを選択します。
3. [Actions]、[Instance State]、[Stop] の順に選択します。
4. 確認ダイアログボックスで [Yes, Stop] を選択します。

5. インスタンスが選択された状態で、[Actions]、[Instance Settings]、[Change Instance Type] の順に選択します。
6. [インスタンスタイプ] で新しいインスタンスタイプを選択し、[適用] を選択します。
7. 停止されているインスタンスを起動するには、インスタンスを選択後、[Actions]、[Instance State]、[Start] の順に選択します。確認ダイアログボックスで [Yes, Start] を選択します。

インスタンスが instance store-backed AMI の場合、以前の手順を使用してインスタンスのサイズを変更することはできません。代わりに、インスタンスから Instance Store-Backed AMI を作成後、新しいインスタンスタイプを使用して AMI から新しいインスタンスを起動することができます。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Instance Store-backed Linux AMI の作成」、および Windows 用 Amazon EC2 ユーザーガイドの「Instance Store-backed Windows AMI の作成」を参照してください。

互換性の制約がある場合は、新しいインスタンスタイプに移行できない場合があります。たとえば、PV 仮想化を使用する AMI から起動されたインスタンスの場合、PV 仮想化と IPv6 の両方をサポートしているインスタンスタイプは C3 のみです。このインスタンスタイプは、ニーズに適さない場合があります。この場合、ソフトウェアをベース HVM AMI に再インストールし、新しいインスタンスを起動する必要がある場合があります。

新しい AMI からインスタンスを起動する場合は、起動時に IPv6 アドレスをインスタンスに割り当てることができます。

ステップ 5: IPv6 アドレスをインスタンスに割り当てる

インスタンスタイプが IPv6 をサポートしていることを確認したら、Amazon EC2 コンソールを使用して IPv6 アドレスをインスタンスに割り当てることができます。IPv6 アドレスは、インスタンスのプライマリネットワークインターフェイス (eth0) に割り当てられます。

IPv6 アドレスをインスタンスに割り当てるには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. インスタンスを選択後、[Actions]、[Networking]、[Manage Private IP Addresses] の順に選択します。
4. [IPv6 Addresses] で、[Assign new IP] を選択します。サブネットの範囲から IPv6 アドレスを入力するか、デフォルト値の `Auto-Assign` のままにして IPv6 アドレスを自動的に割り当てることができます。
5. [Yes, Update] を選択します。

また、新しいインスタンスを起動する場合 (インスタンスタイプを変更できず、AMI を新しく作成した場合など) は、起動時に IPv6 アドレスを割り当てることができます。

起動時に IPv6 アドレスをインスタンスに割り当てるには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. AMI および IPv6 と互換のあるインスタンスタイプを選択後、[Next: Configure Instance Details] を選択します。
3. [Configure Instance Details] ページで、[Network] の VPC を選択後、[Subnet] のサブネットを選択します。[Auto-assign IPv6 IP] で、[Enable] を選択します。
4. ウィザードの残りの手順に従ってインスタンスを起動します。

その IPv6 アドレスを使用してインスタンスに接続できます。ローカルコンピュータから接続する場合は、ローカルコンピュータに IPv6 アドレスがあり、IPv6 を使用するように設定されていることを確認しま

す。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Linux インスタンスへの接続」および Windows インスタンス用 Amazon EC2 ユーザーガイドの「Windows インスタンスへの接続」を参照してください。

ステップ 6: (オプション) インスタンスに IPv6 を設定する

Amazon Linux 2016.09.0 以降、Windows Server 2008 R2 以降、または Ubuntu Server 2018 以降のバージョンを使用してインスタンスを起動した場合、インスタンスは IPv6 に設定され、追加の手順は必要ありません。

別の AMI からインスタンスを起動した場合は、IPv6 および DHCPv6 用に設定されていない可能性があります。つまり、インスタンスに割り当てる IPv6 アドレスはプライマリネットワークインターフェイスでは自動的に認識されません。

Linux で DHCPv6 を検証するには

ping6 コマンドを次のように使用します。

```
$ ping6 ipv6.google.com
```

Windows で DHCPv6 を検証するには

ping コマンドを次のように使用します。

```
C:\> ping -6 ipv6.google.com
```

インスタンスがまだ設定されていない場合は、以下の手順に示すように、手動で設定できます。

手動設定 (オペレーティングシステム別)

- Amazon Linux (p. 139)
- Ubuntu (p. 140)
- RHEL/CentOS (p. 142)
- Windows (p. 144)

Amazon Linux

Amazon Linux インスタンスを設定するには

1. インスタンスのパブリック IPv4 アドレスを使用して、インスタンスに接続します。
2. インスタンスの最新ソフトウェアパッケージを取得する

   ```
   sudo yum update -y
   ```

3. 任意のテキストエディタを使用して、`/etc/sysconfig/network-scripts/ifcfg-eth0` を開き、次の行を見つけ出します。

   ```
   IPV6INIT=no
   ```

   その行を次のように置き換えます。

   ```
   IPV6INIT=yes
   ```

次の 2 行を追加し、変更を保存します。

```
DHCPV6C=yes
DHCPV6C_OPTIONS=-nw
```

4. `/etc/sysconfig/network` を開き、以下の行を削除して変更を保存します。

```
NETWORKING_IPV6=no
IPV6INIT=no
IPV6_ROUTER=no
IPV6_AUTOCONF=no
IPV6FORWARDING=no
IPV6TO4INIT=no
IPV6_CONTROL_RADVD=no
```

5. `/etc/hosts` を開き、コンテンツを以下のように置き換え、変更を保存します。

```
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost6 localhost6.localdomain6
```

6. インスタンスを再起動します。インスタンスに再接続し、`ifconfig` コマンドを使用して、IPv6 アドレスがプライマリネットワークインターフェイスで認識されることを確認します。

Ubuntu

ネットワークインターフェイスに割り当てられた IPv6 アドレスを動的に認識するよう Ubuntu インスタンスを設定できます。インスタンスに IPv6 アドレスがない場合、この設定により、インスタンスの起動時間が最大で 5 分長くなる可能性があります。

目次

Ubuntu Server 16

これらのステップは、ルートユーザーとして実行する必要があります。

Ubuntu Server 16 インスタンスを設定するには

1. インスタンスのパブリック IPv4 アドレスを使用して、インスタンスに接続します。
2. `/etc/network/interfaces.d/50-cloud-init.cfg` ファイルのコンテンツを表示します。

```
cat /etc/network/interfaces.d/50-cloud-init.cfg
```

```
# This file is generated from information provided by
# the datasource.  Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

ループバックネットワークデバイス (lo) が設定されたことを確認し、ネットワークインターフェイスの名前をメモします。この例で、ネットワークインターフェイスの名前は eth0 です。この名前は、インスタンスタイプによって異なる場合があります。

3. ファイル /etc/network/interfaces.d/60-default-with-ipv6.cfg を作成し、以下の行を追加します。必要に応じて、eth0 を上記のステップで取得したネットワークインターフェイスの名前に置き換えます。

```
iface eth0 inet6 dhcp
```

4. インスタンスを再起動するか、次のコマンドを実行してネットワークインターフェイスを再起動します。必要に応じて、eth0 をネットワークインターフェイスの名前に置き換えます。

```
sudo ifdown eth0 ; sudo ifup eth0
```

5. インスタンスに再接続し、ifconfig コマンドを使用して、IPv6 アドレスがネットワークインターフェイスで構成されていることを確認します。

ユーザーデータを使用するよう IPv6 を設定するには

- 新しい Ubuntu インスタンスを起動し、インスタンスに割り当てられた IPv6 アドレスが、起動中に次のユーザーデータを指定して、ネットワークインターフェイスで自動的に設定されることを確認します。

```
#!/bin/bash
echo "iface eth0 inet6 dhcp" >> /etc/network/interfaces.d/60-default-with-ipv6.cfg
dhclient -6
```

この場合、インスタンスに接続して IPv6 アドレスを設定する必要はありません。

詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「起動時に Linux インスタンスでコマンドを実行する」を参照してください。

Ubuntu Server 14

Ubuntu Server 14 を使用している場合は、デュアルスタックネットワークインターフェイスの再起動時に発生する既知の問題 (再起動すると、インスタンスが到達不可能な場合にタイムアウトの延長が発生する) の回避策を含める必要があります。

これらのステップは、ルートユーザーとして実行する必要があります。

Ubuntu Server 14 インスタンスを設定するには

1. インスタンスのパブリック IPv4 アドレスを使用して、インスタンスに接続します。
2. 次のものが含まれるように /etc/network/interfaces.d/eth0.cfg ファイルを編集します。

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
    up dhclient -6 $IFACE
```

3. インスタンスを再起動します。

```
sudo reboot
```

4. インスタンスに再接続し、`ifconfig` コマンドを使用して、IPv6 アドレスがネットワークインターフェイスで構成されていることを確認します。

DHCPv6 クライアントを起動する

または、追加の設定を実行せずにすぐにネットワークインターフェイスの IPv6 アドレスを表示するには、インスタンスの DHCPv6 クライアントを開始できます。ただし、IPv6 アドレスは再起動後にネットワークインターフェイスで永続化されません。

Ubuntu で DHCPv6 クライアントを起動するには

1. インスタンスのパブリック IPv4 アドレスを使用して、インスタンスに接続します。
2. DHCPv6 クライアントを起動する

```
sudo dhclient -6
```

3. `ifconfig` コマンドを使用して、IPv6 アドレスがプライマリネットワークインターフェイスで認識されることを確認します。

RHEL/CentOS

RHEL 7.4 および CentOS 7 以降は、cloud-init を使用してネットワークインターフェイスを設定し、`/etc/sysconfig/network-scripts/ifcfg-eth0` ファイルを生成します。カスタム`cloud-init`設定ファイルを作成して DHCPv6 を有効にできます。それにより、再起動するたびに DHCPv6 を有効にする設定を持つ `ifcfg-eth0` ファイルを生成できます。

> **Note**
>
> 既知の問題により、最新バージョンの cloud-init-0.7.9 で RHEL/CentOS 7.4 を使用している場合、これらのステップを実行すると、再起動後にインスタンスへの接続が失われる可能性があります。回避策として、`/etc/sysconfig/network-scripts/ifcfg-eth0` ファイルを手動で編集できます。

cloud-init を使用して RHEL/CentOS インスタンスを設定するには

1. インスタンスのパブリック IPv4 アドレスを使用して、インスタンスに接続します。
2. 任意のテキストエディタを使用して、たとえば次のようなカスタムファイルを作成します。

```
/etc/cloud/cloud.cfg.d/99-custom-networking.cfg
```

3. 次の行をファイルに追加し、変更を保存します。

```
network:
  version: 1
  config:
  - type: physical
    name: eth0
    subnets:
      - type: dhcp
      - type: dhcp6
```

4. 選択したテキストエディタを使用して、次の行を、`/etc/sysctl.d` の下のインターフェイス固有のファイルに追加します。Consistent Network Device Naming を無効にした場合、network-interface-name は `ethX`、またはセカンダリインターフェイスです。

```
net.ipv6.conf.network-interface-name.accept_ra=1
```

次の例では、ネットワークインターフェイスは `en5` です。

```
net.ipv6.conf.en5.accept_ra=1
```

5. インスタンスを再起動します。
6. インスタンスに再接続し、ifconfig コマンドを使用して、IPv6 アドレスがネットワークインターフェイスで構成されていることを確認します。

または、以下の手順を使用して、`/etc/sysconfig/network-scripts/ifcfg-eth0` ファイルを直接参照できます。この方法は、cloud-init をサポートしていない以前のバージョンの RHEL および CentOS で使用する必要があります。

RHEL/CentOS インスタンスを設定するには

1. インスタンスのパブリック IPv4 アドレスを使用して、インスタンスに接続します。
2. 任意のテキストエディタを使用して、`/etc/sysconfig/network-scripts/ifcfg-eth0` を開き、次の行を見つけ出します。

```
IPV6INIT="no"
```

その行を次のように置き換えます。

```
IPV6INIT="yes"
```

次の 2 行を追加し、変更を保存します。

```
DHCPV6C=yes
NM_CONTROLLED=no
```

3. `/etc/sysconfig/network` を開き、以下の行を以下のように追加または修正して、変更を保存します。

```
NETWORKING_IPV6=yes
```

4. 次のコマンドを実行して、インスタンスのネットワークを再起動します。

```
sudo service network restart
```

ifconfig コマンドを使用して、IPv6 アドレスがプライマリネットワークインターフェイスで認識されることを確認できます。

RHEL 6 または CentOS 6 をトラブルシューティングするには

ネットワーキングを再起動して IPv6 アドレスが取得できないエラーが発生する場合は、`/etc/sysconfig/network-scripts/ifup-eth` を開いて以下の行 (デフォルトでは 327 行) を見つけます。

```
if /sbin/dhclient "$DHCLIENTARGS"; then
```

`$DHCLIENTARGS` を囲む引用符を削除して、変更を保存します。インスタンスでネットワーキングを再起動します。

```
sudo service network restart
```

Windows

Windows Server 2003 および Windows Server 2008 SP2 で IPv6 を設定するには、以下の手順を使用します。

IPv6 を確実に IPv4 に対して優先させるには、Microsoft サポートページ https://support.microsoft.com/en-us/help/929852/how-to-disable-ipv6-or-its-components-in-windows から Prefer IPv6 over IPv4 in prefix policies という名前の修正をダウンロードします。

Windows Server 2003 で IPv6 を有効化して設定するには

1. describe-instances AWS CLI コマンドを使用するか、Amazon EC2 コンソールで、インスタンスの [IPv6 IP] フィールドにチェックマークを付けて、インスタンスの IPv6 アドレスを取得します。
2. インスタンスのパブリック IPv4 アドレスを使用して、インスタンスに接続します。
3. インスタンス内から、[Start]、[Control Panel]、[Network Connections]、[Local Area Connection] の順に選択します。
4. [Properties] 、[Install] の順に選択します。
5. [Protocol]、[Add] の順に選択します。[Network Protocol] リストで、[Microsoft TCP/IP version 6]、[OK] の順に選択します。
6. コマンドプロンプトを起動後、ネットワークシェルを開きます。

   ```
   netsh
   ```
7. インターフェイス IPv6 コンテキストに切り替えます。

   ```
   interface ipv6
   ```
8. 次のコマンドを使用して、IPv6 アドレスをローカルエリア接続に追加します。IPv6 アドレスの値をインスタンスの IPv6 アドレスに置き換えます。

   ```
   add address "Local Area Connection" "ipv6-address"
   ```

   以下に例を示します。

   ```
   add address "Local Area Connection" "2001:db8:1234:1a00:1a01:2b:12:d08b"
   ```
9. ネットワークシェルを終了します。

   ```
   exit
   ```
10. `ipconfig` コマンドを使用して、IPv6 アドレスがローカルエリア接続で認識されることを確認します。

Windows Server 2008 SP2 で IPv6 を有効化して設定するには

1. describe-instances AWS CLI コマンドを使用するか、Amazon EC2 コンソールで、インスタンスの [IPv6 IP] フィールドにチェックマークを付けて、インスタンスの IPv6 アドレスを取得します。
2. インスタンスのパブリック IPv4 アドレスを使用して、Windows インスタンスに接続します。
3. [Start]、[Control Panel] の順に選択します。
4. [Network and Sharing Center] を開いた後、[Network Connections] を開きます。
5. [Local Area Network] (ネットワークインターフェイス用) を右クリックし、[Properties] を選択します。
6. [Internet Protocol Version 6 (TCP/IPv6)] チェックボックスをオンにして、[OK] を選択します。

7. ローカルエリアネットワークの [properties] ダイアログボックスを再度開きます。[Internet Protocol Version 6 (TCP/IPv6)]、[Properties] の順に選択します。
8. [Use the following IPv6 address] を選択して、以下の作業を行います。
   - [IPv6 Address] で、ステップ 1 で取得した IPv6 アドレスを入力します。
   - [Subnet prefix length] で、`64` と入力します。
9. [OK] を選択して、[properties] ダイアログボックスを閉じます。
10. コマンドプロンプトを開きます。`ipconfig` コマンドを使用して、IPv6 アドレスがローカルエリア接続で認識されることを確認します。

共有 VPC の操作

VPC 共有を使用すると、複数の AWS アカウントで、Amazon EC2 インスタンス、Amazon Relational Database Service (RDS) データベース、Amazon Redshift クラスター、AWS Lambda 関数などのアプリケーションリソースを、共有および一元管理される Virtual Private Clouds (VPC) 内に作成できます。このモデルでは、VPC を所有するアカウント (所有者) は、 で同じ組織に属する他のアカウント (参加者) と 1 つまたは複数のサブネットを共有しますAWS Organizations サブネットが共有されると、参加者は共有しているサブネット内にある自分のアプリケーションリソースを表示、作成、変更、および削除できます。参加者は、他の参加者または VPC 所有者に属するリソースを表示、変更、または削除することはできません。

VPC を共有して、同じ信頼境界内にある高度な相互接続を必要とするアプリケーションに、VPC 内の暗黙的なルーティングを活用できます。これにより、作成および管理する VPC の数が減り、課金とアクセスコントロールに別のアカウントを使用できます。 AWS PrivateLink 、Transit Gateway、VPC ピアリングなどの接続機能を使用して共有の Amazon VPC に相互接続することで、ネットワークトポロジーをさらに簡素化できます。VPC 共有の利点の詳細については、「VPC 共有: 複数のアカウントと VPC 管理への新しいアプローチ」を参照してください。

目次

共有 VPC の前提条件

組織の管理アカウントで、リソース共有を有効にしておく必要があります。リソース共有の有効化の詳細については、AWS RAM ユーザーガイドの「AWS Organizations での共有の有効化」を参照してください。

サブネットを共有する

デフォルト以外のサブネットを組織内の他のアカウントと共有できます。サブネットを共有するには、まず共有するサブネットを使用するリソース共有と、そのサブネットを共有する AWS アカウント、組織単位、または組織全体を作成します。リソース共有の作成の詳細については、AWS RAM ユーザーガイドの「リソース共有の作成」を参照してください。

コンソールを使用してサブネットを共有するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Subnets] を選択します。
3. サブネットを選択してから、[Actions (アクション)]、[Share subnet (サブネットの共有)] の順に選択します。
4. リソース共有を選択してから、[Share subnet (サブネットの共有)] を選択します。

AWS CLI を使用してサブネットを共有するには

create-resource-share および associate-resource-share コマンドを使用します。

アベイラビリティーゾーン間でのサブネットのマッピング

リソースがリージョンの複数のアベイラビリティーゾーンに分散されるようにするために、アベイラビリティーゾーンは各 アカウントの名前に個別にマッピングされます。たとえば、`us-east-1a` アカウントのアベイラビリティーゾーン AWS の場所は、別の `us-east-1a` アカウントのアベイラビリティーゾーン AWS の場所と異なる可能性があります。

VPC 共有のためにアカウント間でアベイラビリティーゾーンを調整するには、アベイラビリティーゾーンの一意で一貫性のある識別子である AZ ID を使用する必要があります。たとえば、`use1-az1` は `us-east-1` リージョンのアベイラビリティーゾーンの 1 つです。アベイラビリティーゾーン ID により、アカウント間でリソースの場所を区別できます。詳細については、AWS RAM ユーザーガイドの「リソースの AZ ID」を参照してください。

共有サブネットの共有を解除する

所有者は、いつでも参加者との共有サブネットの共有を解除できます。所有者が共有サブネットの共有を解除した後、以下のルールが適用されます。

- 既存の参加者リソースは非共有サブネットで引き続き実行される。
- 参加者は非共有サブネットに新しいリソースを作成できない。
- 参加者はサブネット内のリソースを変更、定義、削除できる。
- 参加者のリソースがまだ非共有サブネットにある場合、所有者は共有サブネットまたは共有サブネット VPC を削除できない。所有者は、参加者が非共有サブネット内のすべてのリソースを削除した後でのみ、サブネットまたは共有サブネット VPC を削除できます。

コンソールを使用してサブネットの共有を解除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Subnets] を選択します。
3. サブネットを選択してから、[Actions (アクション)]、[Share subnet (サブネットの共有)] の順に選択します。
4. [Actions (アクション)]、[Stop sharing (共有の停止)] の順に選択します。

AWS CLI を使用してサブネットの共有を解除するには

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

共有サブネットの所有者の識別

参加者は、Amazon VPC コンソールまたはコマンドラインツールを使用して、共有しているサブネットを表示できます。

コンソールを使用してサブネット所有者を識別するには

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、[Subnets] を選択します。[Owner (所有者)] 列にサブネットの所有者が表示されます。

AWS CLI を使用してサブネット所有者を識別するには

describe-subnets および describe-vpcs コマンドを使用します。これらの出力に所有者の ID が含まれます。

共有サブネットのアクセス許可

所有者アクセス権限

VPC 所有者は、サブネット、ルートテーブル、ネットワーク ACL、ピアリング接続、ゲートウェイエンドポイント、インターフェイスエンドポイント、Amazon Route 53 Resolver エンドポイント、インターネットゲートウェイ、NAT ゲートウェイ、仮想プライベートゲートウェイ、Transit Gateway アタッチメントなど、VPC レベルのすべてのリソースの作成、管理、削除を担当します。

VPC 所有者は、参加者が作成したセキュリティグループを含む参加者リソースを変更または削除することはできません。VPC 所有者は、トラブルシューティングと監査を容易にするために、すべてのネットワークインターフェイスの詳細と、参加者リソースにアタッチされているセキュリティグループを表示できます。VPC 所有者は、トラフィックのモニタリングまたはトラブルシューティング用に、VPC、サブネット、またはネットワークインターフェイスレベルのフローログのサブスクリプションを作成できます。

参加者のアクセス許可

共有 VPC の参加者は、Amazon EC2 インスタンス、Amazon RDS データベース、ロードバランサーなどのリソースの作成、管理、削除を担当します。参加者は、他の参加者アカウントに属するリソースを表示したり変更したりはできません。参加者は、ルートテーブルの詳細と、共有しているサブネットにアタッチされているネットワーク ACL を表示できます。ただし、ルートテーブル、ネットワーク ACL、サブネットなど、VPC レベルのリソースは変更できません。参加者は、セキュリティグループ ID を使用して、他の参加者または所有者に属するセキュリティグループを参照できます。参加者は、自分が所有するインターフェイスのフローログのサブスクリプションのみを作成できます。参加者は、プライベートホストゾーンを共有 VPC に直接関連付けることはできません。参加者が VPC に関連付けられたプライベートホストゾーンの動作を制御する必要がある場合は、2 つのオプションがあります。

- 参加者は、プライベートホストゾーンを作成し、VPC 所有者と共有できます。プライベートホストゾーンを共有する方法の詳細については、Amazon Route 53 デベロッパーガイド の「別の AWS アカウントで作成した Amazon VPC とプライベートホストゾーンの関連付け」を参照してください。
- VPC 所有者は、既に所有者が VPC に関連付けているプライベートホストゾーンを制御する、クロスアカウント IAM ロールを作成できます。所有者は、ロールを引き受けるのに必要な権限を、参加者のアカウントに付与できます。詳細については、AWS Identity and Access Management ユーザーガイドの「IAM チュートリアル: AWS アカウント間の IAM ロールを使用したアクセスの委任」を参照してください。そうして参加者のアカウントがロールを引き受けることができ、所有者が委任したロールの権限によって、プライベートホストゾーンを制御できます。

所有者と参加者の請求と計測

共有 VPC では、各参加者は、Amazon EC2 インスタンス、Amazon Relational Database Service データベース、Amazon Redshift クラスター、AWS Lambda 関数などのアプリケーションリソースに対して料金を支払います。参加者はアベイラビリティーゾーンのデータ転送、VPC ピアリング接続を介するデータ転送、AWS Direct Connect ゲートウェイを介するデータ転送に対しても料金を請求されます。VPC 所有者は、NAT ゲートウェイ、仮想プライベートゲートウェイ、トランジットゲートウェイ、 AWS

PrivateLink 、および VPC エンドポイントでのデータ処理とデータ転送に対して時間単位料金が課金されます (該当する場合) 。同じアベイラビリティーゾーン内のデータ転送 (AZ ID で一意に識別される) は、通信リソースを所有しているアカウントにかかわらず無料です。

Limitations

VPC 共有の使用には、以下の制限があります。

- 所有者は、 の同じ組織内にある他のアカウントまたは組織単位とのみサブネットを共有できますAWS Organizations
- 所有者は、デフォルトの VPC 内にあるサブネットを共有できません。
- 参加者は、VPC を共有する他の参加者または VPC 所有者が所有するセキュリティグループを使用してリソースを起動することはできません。
- VPC のデフォルトセキュリティグループは所有者に属しているため、参加者はデフォルトのセキュリティグループを使用してリソースを起動することはできません。
- 所有者は、他の参加者が所有するセキュリティグループを使用してリソースを起動することはできません。
- 参加者が共有サブネットでリソースを起動するときは、デフォルトのセキュリティグループに頼らずに、自分のセキュリティグループをリソースにアタッチする必要があります。参加者は、VPC 所有者に属しているため、デフォルトのセキュリティグループを使用することはできません。
- 参加者は、所有していない VPC に Route53 Resolver エンドポイントを作成することはできません。インバウンドエンドポイントなどの VPC レベルのリソースを作成できるのは VPC 所有者のみです。
- VPC タグ、および共有 VPC 内のリソースのタグは、参加者と共有されません。
- サブネット所有者のみが、共有サブネットにトランジットゲートウェイをアタッチできます。参加者はできません。
- 参加者は、共有 VPC 内に Application Load Balancer および Network Load Balancer を作成できますが、共有されていないサブネットで実行されているターゲットを登録することはできません。
- Gateway Load Balancer の作成時に、共有サブネットを選択できるのはサブネット所有者のみです。参加者はできません。
- サービスクォータはアカウントごとに適用されます。

VPC を拡張する

サブネットなどの VPC リソースを世界中の複数の場所でホストできます。これらの場所は、リージョン、アベイラビリティーゾーン、Local Zones、および Wavelength ゾーンで構成されます。リージョンはそれぞれ、地理的に離れた領域です。

- アベイラビリティーゾーンは、各リージョン内の複数の独立した場所です。
- Local Zones を使用すると、コンピューティングやストレージなどのリソースをエンドユーザーに近い複数の場所に配置できます。
- AWS Outposts では、ネイティブの AWS のサービス、インフラストラクチャ、運用モデルをほぼすべてのデータセンター、コロケーションスペース、オンプレミスの施設で利用できます。
- Wavelength Zones を使用すると、デベロッパーは 5G デバイスやエンドユーザーに非常に低いレイテンシーを提供するアプリケーションを構築できます。Wavelength は、標準の AWS コンピューティングおよびストレージサービスを通信事業者の 5G ネットワークのエッジにデプロイします。

AWS は、最新の高可用性のデータセンターを運用しています。しかし、非常にまれですが、同じ場所にあるインスタンスすべての可用性に影響する障害が発生することもあります。すべてのインスタンスを 1 か所でホストしている場合、そのような障害が起きると、すべてのインスタンスが利用できなくなります。

最適なデプロイを確認するには、AWS Wavelength に関するよくある質問を参照してください。

Local Zones で VPC リソースを拡張する

AWS Local Zones では、リソースをエンドユーザーの近くに配置できるほか、使い慣れた API とツールセットを使用して AWS リージョンのサービス全般にシームレスに接続できます。ローカルゾーンの割り当てを持つ新しいサブネットを作成して、VPC リージョンを拡張できます。ローカルゾーンにサブネットを作成すると、VPC はそのローカルゾーンに拡張されます。

ローカルゾーンを使用するには、3 つのステップで構成されるプロセスに従います。

- まず、ローカルゾーンにオプトインします。
- 次に、ローカルゾーン内にサブネットを作成します。
- 最後に、ローカルゾーンサブネットで選択したリソースを起動し、アプリケーションとエンドユーザーを近づけます。

ネットワーク境界グループは、 がパブリック IP アドレスをアドバタイズAWSするアベイラビリティーゾーンまたは Local Zones の一意のセットです。

IPv6 アドレスを持つ VPC を作成する場合、Amazon が提供するパブリック IP アドレスのセットを VPC に割り当てるだけでなく、アドレスをグループに制限するアドレスのネットワーク境界グループを設定することもできます。ネットワーク境界グループを設定すると、IP アドレスはネットワーク境界グループ間を移動できません。`us-west-2` ネットワーク境界グループには、4 つの米国西部 (オレゴン) アベイラビリティーゾーンが含まれます。`us-west-2-lax-1` ネットワーク境界グループには、ロサンゼルスの Local Zones が含まれます。

Local Zones には、以下の規則が適用されます。

- ローカルゾーンのサブネットは、アベイラビリティーゾーンサブネットと同じルーティングルール (ルートテーブル、セキュリティグループ、ネットワーク ACL など) に従います。
- Local Zones は、Amazon Virtual Private Cloud Console、AWS CLI または API を使用してサブネットに割り当てることができます。
- ローカルゾーンで使用するパブリック IP アドレスをプロビジョニングする必要があります。アドレスを割り当てるときに、IP アドレスのアドバタイズ元の場所を指定できます。これをネットワークボーダーグループと呼びます。このパラメータを設定して、アドレスをこの場所に制限することができます。IP アドレスをプロビジョニングした後は、ローカルゾーンと親リージョンの間で IP アドレスを移動できません (例えば、`us-west-2-lax-1a` から `us-west-2`)。
- Amazon が提供する IPv6 IP アドレスをリクエストし、それらのアドレスを新しい VPC または既存の VPC のネットワーク境界グループに関連付けることができます。

  Note

  IPv6 のサポートはロサンゼルスの Local Zones のみになります。
- アウトバウンドインターネットトラフィックは、あるローカルゾーンから そのローカルゾーンを離れます。

Linux での Local Zones 操作方法の詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Local Zones」を参照してください。Windows での Local Zones 操作方法の詳細については、Windows インスタンス用 Amazon EC2 ユーザーガイドの「Local Zones」を参照してください。どちらのガイドにも利用可能な Local Zones および各 Local Zones で起動できるリソースの一覧が記載されています。

インターネットゲートウェイに関する考慮事項

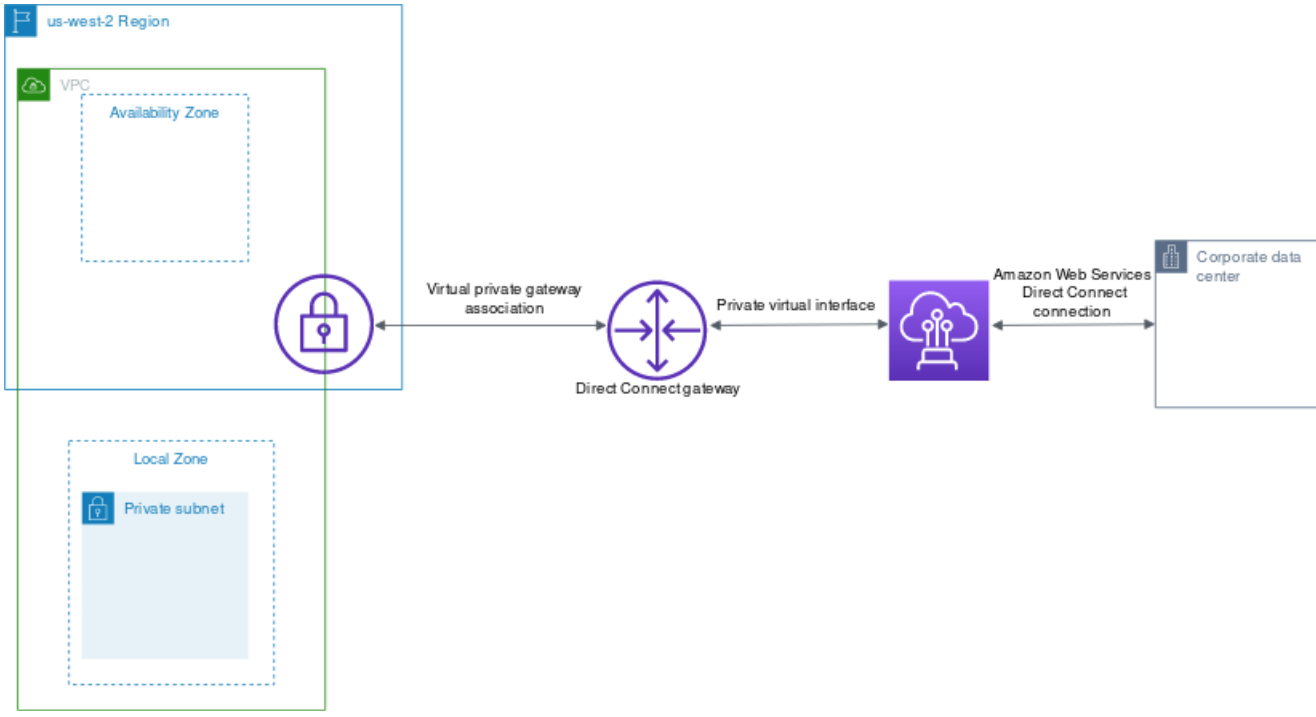
Local Zones で (親リージョンの) インターネットゲートウェイを使用する場合は、次のことを考慮してください。

- elastic IP アドレスまたは Amazon の自動割り当てパブリック IP アドレスを使用して、Local Zonesでインターネットゲートウェイを使用できます。関連付ける elastic IP アドレスには、ローカルゾーンのネットワーク境界グループが含まれている必要があります。詳細については、「the section called “Elastic IP アドレス” (p. 288)」を参照してください。

  リージョンに設定されている elastic IP アドレスを関連付けることはできません。
- Local Zones で使用される elastic IP アドレスは、リージョン内の elastic IP アドレスと同じクォータを持ちます。詳細については、「the section called “Elastic IP アドレス (IPv4)” (p. 362)」を参照してください。
- ローカルゾーンリソースに関連付けられたルートテーブルでは、インターネットゲートウェイを使用できます。詳細については、「the section called “インターネットゲートウェイへのルーティング” (p. 303)」を参照してください。

Direct Connect ゲートウェイを使用した Local Zones へのアクセス

オンプレミスのデータセンターがローカルゾーン内のリソースにアクセスできるようにするシナリオを考えてみましょう。ローカルゾーンに関連付けられた VPC の仮想プライベートゲートウェイを使用して、Direct Connect ゲートウェイに接続します。Direct Connect ゲートウェイは、リージョン内の AWS Direct Connect ロケーションに接続します。オンプレミスのデータセンターには、AWS Direct Connect の場所への AWS Direct Connect 接続があります。



この構成には、次のリソースを使用します。

- ローカルゾーンサブネットに関連付けられた VPC の仮想プライベートゲートウェイ。Amazon Virtual Private Cloud Console のサブネットの詳細ページ、または describe-subnets を使用してサブネットの VPC を表示できます。

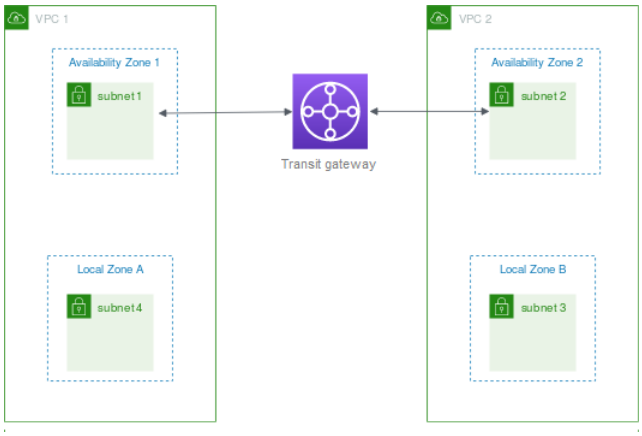
  仮想プライベートゲートウェイの作成方法の詳細については、AWS Site-to-Site VPN ユーザーガイドの「ターゲットゲートウェイを作成する」を参照してください。

- Direct Connect 接続。AWS では、ロサンゼルスの Local Zones に対するレイテンシーパフォーマンスを最適化するために、次のいずれかの場所を使用することをお勧めします。
  - カリフォルニア州ロサンゼルス、エルセグンドにある T5 (AWS では、ロサンゼルスのローカルゾーンへのレイテンシーを最小にするため、このロケーションを推奨します)
  - CoreSite LA1、カリフォルニア州ロサンゼルス
  - Equinix LA3、カリフォルニア州エルセグンド

  接続の注文方法については、AWS Direct Connect ユーザーガイドの「クロスコネクト」を参照してください。
- Direct Connect ゲートウェイ Direct Connect ゲートウェイの作成方法の詳細については、AWS Direct Connect ユーザーガイドの「Direct Connect ゲートウェイを作成する」を参照してください。
- VPC を Direct Connect ゲートウェイに接続するための仮想プライベートゲートウェイの関連付け。仮想プライベートゲートウェイの関連付け作成方法については、AWS Direct Connect ユーザーガイドの「仮想プライベートゲートウェイの関連付けと関連付けの解除」を参照してください。
- AWS Direct Connect ロケーションからオンプレミスのデータセンターへの接続のプライベート仮想インターフェイス。Direct Connect ゲートウェイの作成方法の詳細については、AWS Direct Connect ユーザーガイドの「Direct Connect ゲートウェイへのプライベート仮想インターフェイスの作成」を参照してください。

ローカルゾーンのサブネットをトランジットゲートウェイに接続する

ローカルゾーンでサブネットにトランジットゲートウェイアタッチメントを作成することはできません。次の図は、親アベイラビリティーゾーンから、ローカルゾーンのサブネットをトランジットゲートウェイに接続するようにネットワークを設定する方法を示しています。Local Zones にサブネットを作成し、親アベイラビリティーゾーンにサブネットを作成します。親アベイラビリティーゾーンのサブネットをトランジットゲートウェイに接続し、各 VPC のルートテーブルに、他の VPC の CIDR 宛のトラフィックをトランジットゲートウェイアタッチメントのネットワークインターフェイスにルーティングするルートを作成します。



このシナリオでは、次のリソースを作成します。

- 各親アベイラビリティーゾーンのサブネット。詳細については、「the section called “VPC にサブネットを作成する” (p. 117)」を参照してください。
- トランジットゲートウェイ。詳細については、「Amazon VPC Transit Gateway」の「Transit Gateway の作成」を参照してください。
- 親アベイラビリティーゾーンを使用する各 VPC のトランジットゲートウェイアタッチメント。詳細については、「Amazon VPC Transit Gateway」の「VPC へのTransit Gateway の作成」を参照してください。

- トランジットゲートウェイアタッチメントに関連付けられたトランジットゲートウェイルートテーブル。詳細については、「Amazon VPC Transit Gateway」の「Transit Gateway ルートテーブル」を参照してください。
- VPC ごとに、VPC ルートテーブルに、他の VPC の CIDR を送信先とし、トランジットゲートウェイアタッチメントのネットワークインターフェイスの ID を対象としたエントリ。トランジットゲートウェイアタッチメントのネットワークインターフェイスを検索するには、ネットワークインターフェイスの説明で、トランジットゲートウェイアタッチメントの ID を検索します。詳細については、「the section called “トランジットゲートウェイのルーティング” (p. 307)」を参照してください。

VPC 1 のルートテーブルの例を次に示します。

送信先	ターゲット
VPC 1 CIDR	*####*
VPC 2 CIDR	*vpc1-attachment-network-interface-id*

VPC 2 のルートテーブルの例を次に示します。

送信先	ターゲット
VPC 2 CIDR	*####*
VPC 1 CIDR	*vpc2-attachment-network-interface-id*

トランジットゲートウェイのルートテーブルの例を次に示します。各 VPC の CIDR ブロックがトランジットゲートウェイルートテーブルに伝播されます。

CIDR	Attachment	ルートタイプ
VPC 1 CIDR	*VPC 1 ########*	伝播済み
VPC 2 CIDR	*VPC 2 ########*	伝播済み

VPC リソースを Wavelength Zones に拡張する

AWS Wavelength では、開発者はモバイルデバイスおよびエンドユーザー向けに、非常にレイテンシーが低いアプリケーションを構築できます。Wavelength は、標準の AWS コンピューティングおよびストレージサービスを通信事業者の 5G ネットワークのエッジにデプロイします。開発者は、Amazon Virtual Private Cloud (VPC) を 1 つ以上の Wavelength ゾーンに拡張し、Amazon Elastic Compute Cloud (EC2) インスタンスなどの AWS リソースを使用して、非常に低いレイテンシーを必要とするアプリケーションを実行し、リージョン内の AWS サービスに接続できます。

Wavelength Zones を使用するには、まずゾーンにオプトインする必要があります。次に、Wavelength ゾーンにサブネットを作成します。Amazon EC2 インスタンス、Amazon EBS ボリューム、Amazon VPC サブネット、および Carrier Gateway を Wavelength Zones に作成できます。Amazon EC2 Auto Scaling、Amazon EKS クラスター、Amazon ECS クラスター、Amazon EC2 Systems Manager、Amazon CloudWatch、AWS CloudTrail、AWS CloudFormation など、EC2、EBS、および VPC と調整または連携しているサービスを使用することもできます。Wavelength のサービスは、Amazon DynamoDB や Amazon RDS などのサービスに簡単にアクセスできるように、信頼性の高い高帯域幅接続を介して AWS リージョンに接続されている VPC の一部です。

Wavelength Zones には、次の規則が適用されます。

- VPC でサブネットを作成し、それを Wavelength ゾーンに関連付けると、VPC は Wavelength Zone まで拡張されます。
- デフォルトでは、Wavelength ゾーンにまたがる VPC で作成するすべてのサブネットは、ローカルルートを含むメイン VPC ルートテーブルを継承します。
- Wavelength ゾーンのサブネットで EC2 インスタンスを起動するときは、そのインスタンスにキャリア IP アドレスを割り当てます。キャリアゲートウェイは、インターフェイスからインターネット、またはモバイルデバイスへのトラフィックに、そのアドレスを使用します。キャリアゲートウェイは NAT を使用してアドレスを変換し、トラフィックを送信先に送信します。通信キャリアネットワークからのトラフィックは、キャリアゲートウェイを経由します。
- VPC ルートテーブル、または Wavelength ゾーンのサブネットルートテーブルのターゲットを、キャリアゲートウェイに設定できます。キャリアゲートウェイは、特定の場所のキャリアネットワークからのインバウンドトラフィックと、キャリアネットワークおよびインターネットへのアウトバウンドトラフィックを許可します。Wavelength ゾーンでのルーティングオプションの詳細については、AWS Wavelength 開発者ガイドの「ルーティング」を参照してください。
- Wavelength Zones のサブネットには、IPv4 アドレス、DHCP オプションセット、ネットワーク ACL など、アベイラビリティーゾーンのサブネットと同じネットワークコンポーネントがあります。
- Wavelength ゾーンでサブネットへのトランジットゲートウェイアタッチメントを作成することはできません。代わりに、親アベイラビリティーゾーンのサブネットを介して添付ファイルを作成し、トランジットゲートウェイを介して目的の送信先にトラフィックをルーティングします。例については、次のセクションを参照ください。

複数の Wavelength ゾーンに関する考慮事項

同じ VPC 内の異なる Wavelength ゾーンにある EC2 インスタンスは、相互に通信することができません。Wavelength ゾーン間の通信が必要な場合、AWS ではWavelength ゾーンごとに 1 つずつ、複数の VPC を使用することをお勧めします。中継ゲートウェイを使用して VPC に接続できます。この設定により、Wavelength ゾーンのインスタンス間で通信が可能になります。

Wavelength ゾーン間のトラフィックは、AWS リージョンを介してルーティングされます。詳細については、「AWS Transit Gateway」を参照してください。

次の図は、2 つの異なる Wavelength ゾーンのインスタンスが通信できるようにネットワークを設定する方法を示しています。2 つの Wavelength ゾーン (Wavelength ゾーン A と Wavelength ゾーン B) があります。通信を有効にするには、次のリソースを作成する必要があります。

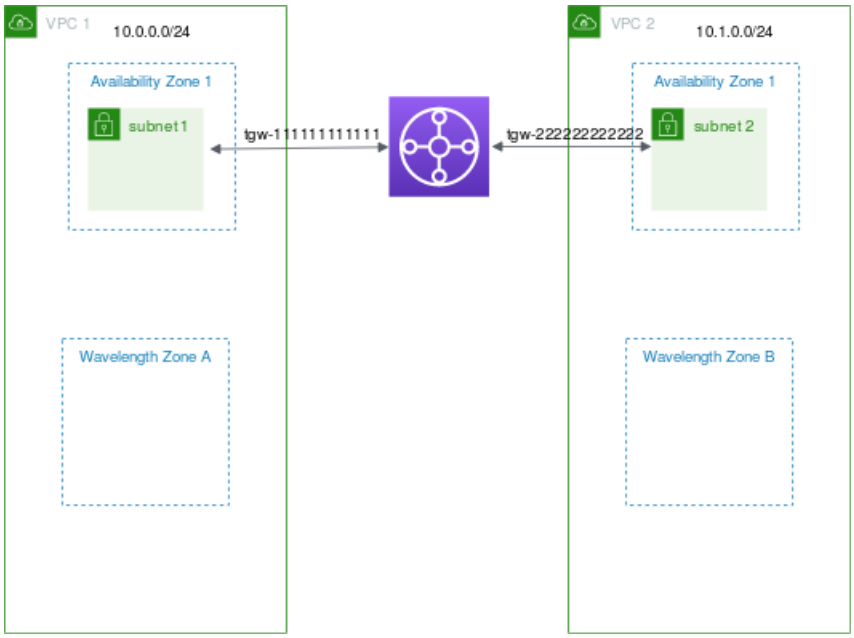
- 各 Wavelength ゾーンについて、その Wavelength ゾーンの親アベイラビリティーゾーンであるアベイラビリティーゾーン内のサブネット。この例では、サブネット 1 とサブネット 2 を作成します。サブネットの作成の詳細については、「the section called “VPC にサブネットを作成する” (p. 117)」を参照してください。describe-availability-zones を使用して、親ゾーンを検索します。
- トランジットゲートウェイ。VPC に接続するトランジットゲートウェイ。トランジットゲートウェイの作成方法の詳細については、Amazon VPC Transit Gatewaysの「トランジットゲートウェイの作成」を参照してください。
- Wavelength ゾーンの親アベイラビリティーゾーン内のトランジットゲートウェイへの VPC ごとの VPC アタッチメント。詳細については、「Amazon VPC トランジットゲートウェイ」の「VPC へのトランジットゲートウェイの作成」を参照してください。
- トランジットゲートウェイルートテーブル内の各 VPC のエントリ。トランジットゲートウェイルートの作成方法の詳細については、Amazon VPC Transit Gateways ガイドの「トランジットゲートウェイルートテーブル」を参照してください。
- VPC ごとに、他の VPC CIDR を送信先とし、トランジットゲートウェイ ID をターゲットとする VPC ルートテーブル内のエントリ。詳細については、「the section called “トランジットゲートウェイのルーティング” (p. 307)」を参照してください。

  この例では、VPC 1 のルートテーブルには次のエントリがあります。

送信先	ターゲット
10.1.0.0/24	tgw-22222222222222222

VPC 2 のルートテーブルには、次のエントリがあります。

送信先	ターゲット
10.0.0.0/24	tgw-22222222222222222



AWS Outposts のサブネット

AWS Outposts では、同じ AWS ハードウェアインフラストラクチャ、サービス、API、ツールを提供、オンプレミスやクラウドでアプリケーションを構築して実行することができます。AWS Outposts はオンプレミスのアプリケーションやシステムに対し低レイテンシーのアクセスを必要とするワークロード、データをローカルに保存および処理する必要があるワークロードに最適です。AWS Outposts の詳細については、AWS Outpostsを参照してください。

Amazon VPC は、AWS リージョンのすべてのアベイラビリティーゾーンにまたがります。Outposts を親リージョンに接続すると、アカウント内の既存および新規作成された VPC はすべて、リージョン内のすべてのアベイラビリティーゾーンおよび関連付けられている Outpost ロケーションにまたがります。

AWS Outposts には以下のルールが適用されます。

- サブネットは、1 つの Outpost の場所に存在する必要があります。
- ローカルゲートウェイは、VPC とオンプレミスネットワーク間のネットワーク接続を処理します。ローカルゲートウェイの詳細については、AWS Outposts ユーザーガイドの「ローカルゲートウェイ」を参照してください。
- アカウントが AWS Outposts に関連付けられている場合は、サブネットの作成時に Outpost ARN を指定して、サブネットを Outpost に割り当てます。

- デフォルトでは、Outpost に関連付けられた VPC で作成するすべてのサブネットは、ローカルゲートウェイルートを含むメイン VPC ルートテーブルを継承します。また、カスタムルートテーブルを VPC 内のサブネットに明示的に関連付けて、オンプレミスネットワークにルーティングする必要があるすべてのトラフィックのネクストホップターゲットとしてローカルゲートウェイを設定することもできます。

デフォルト VPC とデフォルトサブネット

2013 年 12 月 4 日より後に AWS アカウントを作成した場合、EC2-VPC のみサポートされます。この場合、AWS リージョンごとにデフォルト VPC が用意されています。デフォルト VPC は使用できる状態になっているため、独自の VPC の作成および設定は不要です。すぐにデフォルト VPC に Amazon EC2 インスタンスの起動を開始できます。デフォルトの VPC では、Elastic Load Balancing、Amazon RDS、および Amazon EMR などのサービスを使用することもできます。

デフォルト VPC は、すぐに使用を開始する場合や、ブログやシンプルなウェブサイトなど、パブリックインスタンスを起動する場合に適しています。デフォルト VPC のコンポーネントは、必要に応じて変更できます。特定の要件 (例: 推奨される CIDR ブロック範囲やサブネットサイズを使用する) に適したデフォルト以外の VPC を作成する場合は、「シナリオの例 (p. 80)」を参照してください。

目次

デフォルト VPC のコンポーネント

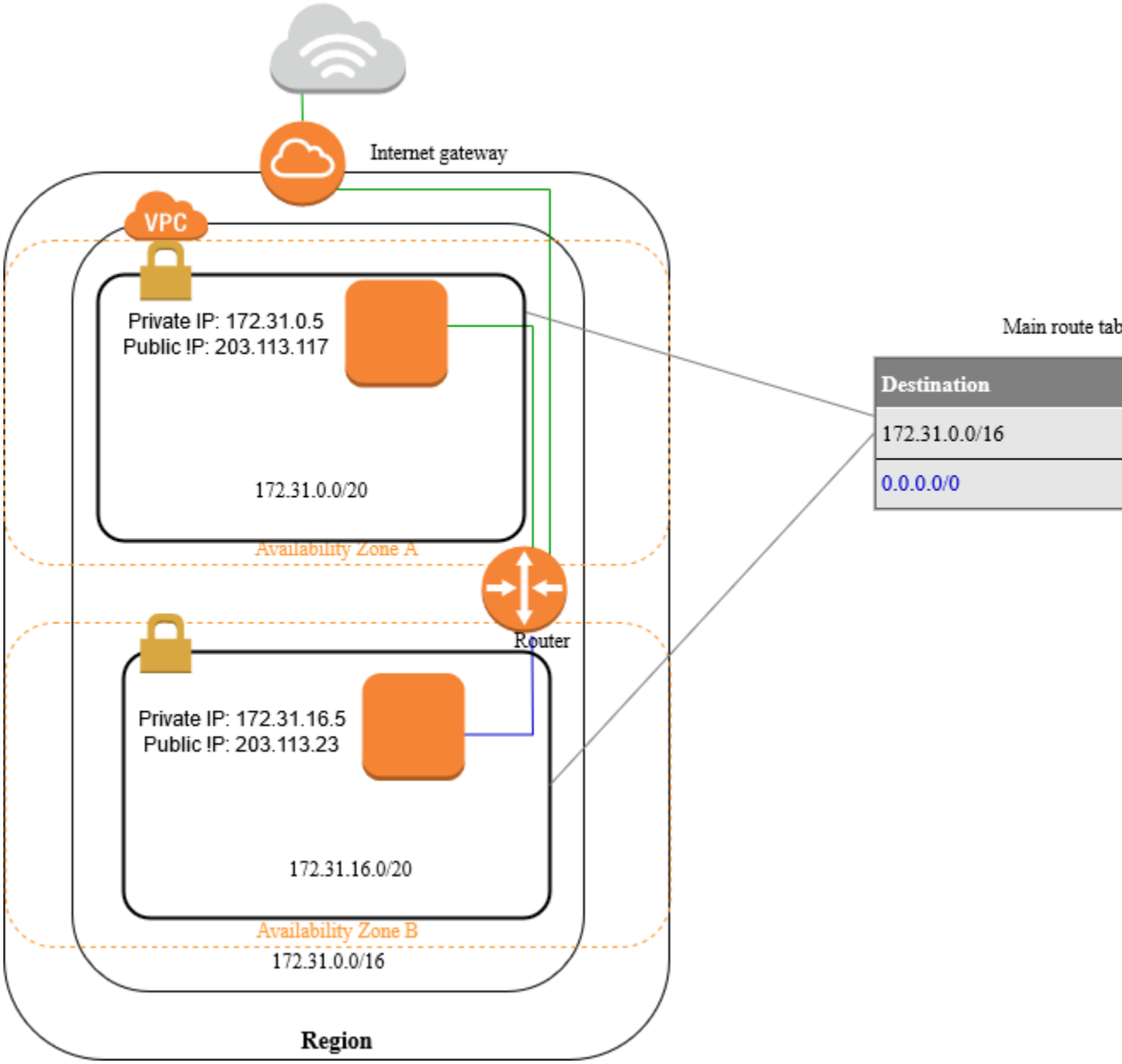
デフォルト VPC を作成するとき、Amazon 側で次の設定を行います。

- サイズ `/16` の IPv4 CIDR ブロック (`172.31.0.0/16`) の VPC を作成する。これは、最大 65,536 個のプライベート IPv4 アドレスを提供します。
- 各アベイラビリティーゾーンに、サイズ `/20` のデフォルトサブネットを作成する。この場合は、サブネットあたり最大 4,096 個のアドレスが作成され、その中のいくつかは Amazon が使用するように予約されています。
- インターネットゲートウェイ (p. 221)を作成して、デフォルト VPC に接続する。
- すべてのトラフィック (`0.0.0.0/0`) をインターネットゲートウェイにポイントさせるルートをメインルートテーブルに追加します。
- デフォルトのセキュリティグループを作成し、デフォルト VPC に関連付ける。
- デフォルトのネットワークアクセスコントロールリスト (ACL) を作成し、デフォルト VPC に関連付ける。
- デフォルト VPC を備えた AWS アカウントに、デフォルトの DHCP オプションセットを関連付けます。

> **Note**
>
> Amazonは、ユーザーに代わって上記のリソースを作成します。ユーザーがこれらのアクションを実行するわけではないため、IAM ポリシーはこれらのアクションに適用されません。たとえ

ば、CreateInternetGateway を呼び出す機能を拒否する IAM ポリシーがあり、CreateDefaultVpc を呼び出した場合でも、デフォルト VPC 内のインターネットゲートウェイが作成されます。

次の図は、デフォルト VPC に対して設定する重要なコンポーネントを示します。



デフォルト VPC は、他の VPC と同じように使用できます。

- デフォルト以外のサブネットを追加します。
- メインルートテーブルを変更します。
- ルートテーブルを追加します。
- 追加セキュリティグループを関連付けます。
- デフォルトのセキュリティグループのルールを更新します。
- AWS Site-to-Site VPN 接続を追加します。
- さらに多くの IPv4 CIDR ブロックを追加します。

- Direct Connect ゲートウェイを使用して、リモートリージョン内の VPC にアクセスします。Direct Connect ゲートウェイオプションの詳細については、AWS Direct Connect ユーザーガイドの「Direct Connect ゲートウェイ」を参照してください。

デフォルトサブネットは、他のサブネットと同じように (カスタムルートテーブルの追加、ネットワーク ACL の設定など) 使用できます。また、EC2 インスタンスを起動するときに、特定のデフォルトサブネットを指定することもできます。

オプションで IPv6 CIDR ブロックをデフォルト VPC と関連付けることができます。詳細については, VPC とサブネットの使用 (p. 115).

デフォルトサブネット

デフォルトでは、デフォルトサブネットはパブリックサブネットに指定されています。メインルートテーブルがインターネット用のサブネットのトラフィックをインターネットゲートウェイに送信するためです。デフォルトサブネットをプライベートサブネットにするには、送信元 0.0.0.0/0 からインターネットゲートウェイへのルートを削除します。ただし、この操作を行った場合、そのサブネットで実行されている EC2 インスタンスすべてがインターネットにアクセスできなくなります。

デフォルトサブネット内に起動する各インスタンスは、パブリック IPv4 アドレスとプライベート IPv4 アドレスの両方、およびパブリックとプライベート DNS ホスト名の両方を受け取ります。デフォルト VPC 内のデフォルト以外のサブネット内に起動するインスタンスは、パブリック IPv4 アドレスまたはパブリック DNS ホスト名を受け取りません。サブネットのデフォルトのパブリック IP アドレス指定の動作は変更できます。詳細については、「」を参照してくださいサブネットのパブリック IPv4 アドレス属性を変更する (p. 128)

AWS によって、リージョンに新しいアベイラビリティーゾーンが追加される場合があります。ほとんどの場合、数日以内に、このアベイラビリティーゾーン内でデフォルト VPC の新しいデフォルトサブネットが自動的に作成されます。ただし、デフォルト VPC への変更を行った場合、新しいデフォルトサブネットは追加されません。新しいアベイラビリティーゾーンでデフォルトサブネットが必要な場合は、独自に作成できます。詳細については、「」を参照してくださいデフォルトのサブネットを作成する (p. 162)

アベイラビリティーおよびサポートされているプラットフォーム

2013 年 12 月 4 日以降に AWS アカウントを作成した場合は、EC2-VPC のみをサポートし、各 AWS リージョンにデフォルト VPC があります。したがって、デフォルト以外の VPC を作成し、インスタンスの起動時にそれを指定しない限り、インスタンスは、デフォルト VPC 内に起動されます。

2013 年 3 月 18 日より前に AWS アカウントを作成した場合、以前使用したことがあるリージョンでは、EC2-Classic と EC2-VPC の両方がサポートされます。使用したことがないリージョンでは EC2-VPC のみがサポートされます。この場合、AWS リソースを作成しなかった各リージョンには、デフォルト VPC が作成されます。デフォルト以外の VPC を作成して、新しいリージョンでインスタンスを起動するときにそれを指定する場合を除き、インスタンスは、該当リージョンのデフォルト VPC で起動されます。ただし、以前使用したことがあるリージョン内にインスタンスを起動する場合、そのインスタンスは EC2-Classic 内に起動されます。

2013 年 3 月 18 日から 2013 年 12 月 4 日の間に AWS アカウントを作成した場合は、EC2-VPC のみサポートされている場合があります。または、使用したリージョンの一部では、EC2-Classic および EC2-VPC の両方をサポートしていることがあります。AWS アカウントの各リージョンがサポートするプラットフォームを確認する場合は、「サポートされているプラットフォームの検出 (p. 159)」を参照してください。各リージョンでデフォルト VPC が有効になった時期については、Amazon VPC の AWS フォーラムの「お知らせ: デフォルトの VPC 機能セットのリージョンを有効にする」を参照してください。

AWS アカウントが EC2-VPC のみをサポートしている場合は、その AWS アカウントに関連付けられている IAM アカウントも EC2-VPC のみをサポートし、AWS アカウントと同じデフォルト VPC を使用します。

AWS アカウントが EC2-Classic と EC2-VPC のいずれもサポートしている場合は、新しい AWS アカウントを作成するか、以前に使用したことがないリージョンでインスタンスを起動することができます。そのため、EC2-Classic 内にインスタンスを起動するシンプルさを備えた EC2-VPC の利点を活用できる場合があります。デフォルト VPC が存在しないリージョンで EC2-Classic がサポートされている場合、このリージョンにデフォルト VPC を追加するには、VPC のよくある質問で「既存の EC2 アカウントでどうしてもデフォルト VPC が使いたいのです。なにか方法はありますか？」(デフォルト VPC のよくある質問)を参照してください。

サポートされているプラットフォームの検出

AWS アカウントが両方のプラットフォームをサポートしているかどうか、またはデフォルト VPC があるかどうかを確認するには、Amazon EC2 コンソールまたはコマンドラインを使用できます。

Amazon EC2 コンソールを使用してプラットフォームのサポート有無を確認するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションバーで、右上のリージョンセレクターを使用して、リージョンを選択します。
3. Amazon EC2 コンソールダッシュボードの [アカウントの属性] の下にある [サポートされているプラットフォーム] を探します。そこに値が 2 つ (`EC2` と `VPC`) あれば、どちらのプラットフォームにもインスタンスを起動できます。値が 1 つ (`VPC`) なら、EC2-VPC にのみインスタンスを起動できます。

   例えば、次の図は、アカウントが EC2-VPC プラットフォームのみをサポートしていること、および識別子 `vpc-1a2b3c4d` を持つデフォルト VPC があることを示しています。

   デフォルト VPC を削除すると、[Default VPC] 値として `None` が表示されます。

コマンドラインを使用してプラットフォームのサポート有無を確認するには

- describe-account-attributes (AWS CLI)
- Get-EC2AccountAttribute (AWS Tools for Windows PowerShell)

出力の `supported-platforms` 属性は、EC2 インスタンスを起動できるプラットフォームを示します。

デフォルト VPC とデフォルトサブネットの表示

デフォルト VPC およびデフォルトサブネットを表示するには、Amazon VPC コンソールまたはコマンドラインを使用します。

Amazon VPC コンソールを使用して、デフォルト VPC とデフォルトサブネットを表示するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。

3. [Default VPC] 列で、[Yes] の値を探します。デフォルト VPC の ID をメモしておきます。
4. ナビゲーションペインで、[Subnets] を選択します。
5. 検索バーで、デフォルト VPC の ID を入力します。デフォルト VPC のサブネットが返ります。
6. どのサブネットがデフォルトサブネットかを確認するには、[Default Subnet] 列で [Yes] の値を探します。

コマンドラインを使用してデフォルト VPC を記述するには

- describe-vpcs を使用する (AWS CLI)
- Get-EC2Vpc を使用する (AWS Tools for Windows PowerShell)

このコマンドを使用するときは、`isDefault` フィルタの値を `true` に設定します。

コマンドラインを使用してデフォルトサブネットを記述するには

- describe-subnets を使用する (AWS CLI)
- Get-EC2Subnet を使用する (AWS Tools for Windows PowerShell)

このコマンドを使用するときは、`vpc-id` フィルタの値をデフォルト VPC の ID に設定します。出力で、`DefaultForAz` フィールドは、デフォルトサブネットの `true` に設定されます。

EC2 インスタンスをデフォルト VPC 内に起動する

サブネットを指定せずに EC2 インスタンスを起動すると、そのインスタンスはデフォルト VPC のデフォルトサブネット内に自動的に起動されます。デフォルトでは、アベイラビリティーゾーンが選択され、インスタンスは、そのアベイラビリティーゾーンに対応するサブネットから起動されます。また、インスタンスのアベイラビリティーゾーンを選択することもできます。そのためには、対応するデフォルトサブネットをコンソールで選択するか、サブネットまたはアベイラビリティーゾーンを AWS CLI で指定します。

コンソールを使用した EC2 インスタンスの起動

EC2 インスタンスをデフォルト VPC 内に起動するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. EC2 ダッシュボードから、[Launch Instance] を選択します。
3. ウィザードの指示にしたがって操作します。AMI を選択し、インスタンスタイプを選択します。[Review and Launch] を選択すると、ウィザードの残りの部分のデフォルト設定を確定できます。これにより、[Review Instance Launch] ページが直接表示されます。
4. 設定を確認します。[Instance Details] セクションで、[Subnet] のデフォルトは [No preference (default subnet in any Availability Zone)] です。つまり、インスタンスは、選択したアベイラビリティーゾーンのデフォルトサブネット内に起動されていることを意味します。また、[Edit instance details] を選択し、特定のアベイラビリティーゾーンのデフォルトサブネットを選択します。
5. [Launch] を選択し、キーペアを選択してインスタンスを起動します。

コマンドラインを使用して EC2 インスタンスを起動する

次のいずれかのコマンドを使用して、EC2 インスタンスを起動できます。

- run-instances (AWS CLI)
- New-EC2Instance (AWS Tools for Windows PowerShell)

デフォルト VPC で EC2 インスタンスを起動するには、サブネットやアベイラビリティーゾーンを指定しないでこれらのコマンドを使用します。

EC2 インスタンスをデフォルト VPC の特定のサブネット内に起動するには、サブネット ID またはアベイラビリティーゾーンを指定します。

デフォルトサブネットとデフォルト VPC の削除

デフォルトサブネットやデフォルト VPC は、他のサブネットや VPC と同様、削除できます。詳細については、「」を参照してくださいVPC とサブネットの使用 (p. 115) デフォルトサブネットやデフォルト VPC を削除する場合、インスタンスを起動する別の VPC のサブネットを明示的に指定する必要があります。これは、EC2-Classic でインスタンスを起動できないためです。別の VPC がない場合は、デフォルト以外の VPC とデフォルト以外のサブネットを作成する必要があります。詳細については、「」を参照してくださいVPC を作成する (p. 116)

デフォルト VPC を削除した場合は、新しく作成することができます。詳細については、「」を参照してくださいデフォルトの VPC を作成する (p. 161)

デフォルトサブネットを削除した場合は、新しく作成できます。詳細については、「デフォルトのサブネットを作成する (p. 162)」を参照してください。新しいデフォルトサブネットが想定どおりに動作することを確認するには、サブネット属性を変更して、そのサブネットで起動されたインスタンスにパブリック IP アドレスを割り当てます。詳細については、「」を参照してくださいサブネットのパブリック IPv4 アドレス属性を変更する (p. 128) アベイラビリティーゾーンごとに 1 つだけデフォルトサブネットを持つことができます。デフォルト以外の VPC でデフォルトサブネットを作成することはできません。

デフォルトの VPC を作成する

デフォルト VPC を削除した場合は、新しく作成することができます。以前の削除したデフォルト VPC を復元することはできません。また、デフォルト以外の既存の VPC をデフォルト VPC としてマーキングすることはできません。アカウントが EC2-Classic をサポートしている場合は、これらの手順を使用して、EC2-Classic をサポートするリージョンにデフォルト VPC を作成することはできません。

デフォルト VPC を作成する場合、各アベイラビリティーゾーンのデフォルトサブネットなど、デフォルト VPC の標準コンポーネント (p. 156)を使用して作成されます。独自のコンポーネントを指定することはできません。新しいデフォルト VPC では、サブネット CIDR ブロックが、以前のデフォルト VPC と同じアベイラビリティーゾーンにマッピングされない場合があります。たとえば、CIDR ブロック (`172.31.0.0/20`) を持つサブネットが、以前のデフォルト VPC の `us-east-2a` に作成されていた場合、新しいデフォルト VPC では `us-east-2b` に作成される場合があります。

デフォルト VPC がすでに該当リージョンに作成されている場合は、新しく作成することはできません。

Amazon VPC コンソールを使用してデフォルト VPC を作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. [Actions]、[Create Default VPC] の順に選択します。
4. [Create] を選択します。確認画面を閉じます。

コマンドラインを使用してデフォルト VPC を作成するには

create-default-vpc AWS CLI コマンドを使用できます。このコマンドには、入力パラメータがありません。

```
aws ec2 create-default-vpc
```

出力例を次に示します。

```
{
    "Vpc": {
        "VpcId": "vpc-3f139646",
        "InstanceTenancy": "default",
        "Tags": [],
        "Ipv6CidrBlockAssociationSet": [],
        "State": "pending",
        "DhcpOptionsId": "dopt-61079b07",
        "CidrBlock": "172.31.0.0/16",
        "IsDefault": true
    }
}
```

または、New-EC2DefaultVpc Tools for Windows PowerShell コマンド、または CreateDefaultVpc Amazon EC2 API アクションを使用することもできます。

デフォルトのサブネットを作成する

アベイラビリティーゾーンにデフォルトサブネットがない場合は、これを作成できます。たとえば、デフォルトサブネットを削除したか、AWS に新しく追加されたアベイラビリティーゾーンでデフォルトサブネットがデフォルト VPC 内に自動的に作成されなかった場合、デフォルトサブネットを作成できます。

デフォルトサブネットを作成すると、そのサイズはデフォルト VPC で次に利用可能な連続領域の /20 IPv4 CIDR ブロックになります。以下のルールが適用されます。

- CIDR ブロックを独自に指定することはできません。
- 削除済みのデフォルトサブネットは復元できません。
- デフォルトサブネットは、アベイラビリティーゾーンごとに 1 つに限ります。
- デフォルト以外の VPC でデフォルトサブネットを作成することはできません。

デフォルト VPC のアドレス空間が足りなくてサイズが /20 の CIDR ブロックを作成できない場合、リクエストは失敗します。追加のアドレス空間が必要な場合は、IPv4 CIDR ブロックを VPC に追加する (p. 110)ことができます。

IPv6 CIDR ブロックをデフォルト VPC に関連付けている場合、新しいデフォルトサブネットは IPv6 CIDR ブロックを自動的に受け取りません。代わりに、デフォルトサブネットを作成した後で IPv6 CIDR ブロックを関連付けることができます。詳細については、「」を参照してくださいIPv6 CIDR ブロックとサブネットの関連付け (p. 120)

AWS Management Console を使用してデフォルトのサブネットを作成することはできません。

AWS CLI を使用してデフォルトのサブネットを作成するには

create-default-subnet AWS CLI コマンドを使用し、サブネットを作成する先のアベイラビリティーゾーンを指定します。

```
aws ec2 create-default-subnet --availability-zone us-east-2a
```

出力例を次に示します。

```
{
    "Subnet": {
        "AvailabilityZone": "us-east-2a",
        "Tags": [],
        "AvailableIpAddressCount": 4091,
        "DefaultForAz": true,
        "Ipv6CidrBlockAssociationSet": [],
        "VpcId": "vpc-1a2b3c4d",
        "State": "available",
        "MapPublicIpOnLaunch": true,
        "SubnetId": "subnet-1122aabb",
        "CidrBlock": "172.31.32.0/20",
        "AssignIpv6AddressOnCreation": false
    }
}
```

AWS CLI を設定する方法の詳細については、AWS Command Line Interface ユーザーガイドを参照してください。

または、New-EC2DefaultSubnet Tools for Windows PowerShell コマンド、または CreateDefaultSubnet Amazon EC2 API アクションを使用することができます。

Amazon Virtual Private Cloud でのセキュリティ

AWS では、クラウドのセキュリティが最優先事項です。AWS のお客様は、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャから利点を得られます。

セキュリティは、AWS とお客様の間の共有責任です。責任共有モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ - AWS は、AWS クラウドで AWS のサービスを実行するインフラストラクチャを保護する責任を負います。また、AWS は、使用するサービスを安全に提供します。AWS コンプライアンスプログラムの一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。Amazon Virtual Private Cloud に適用されるコンプライアンスプログラムの詳細については、コンプライアンスプログラムによる AWS 対象範囲内のサービス」を参照してください。
- クラウド内のセキュリティ – お客様の責任は、使用する AWS のサービスに応じて判断されます。また、お客様は、お客様のデータの機密性、企業の要件、および適用可能な法律および規制などの他の要因についても責任を担います。

このドキュメントは、Amazon VPC を使用する際の責任共有モデルの適用方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するように Amazon VPC を設定する方法について説明します。また、Amazon VPC リソースのモニタリングや保護に役立つ他の AWS のサービスの使用方法についても説明します。

目次

Amazon Virtual Private Cloud のデータ保護

AWS 責任共有モデルは Amazon Virtual Private Cloud のデータ保護に適用されます。このモデルで説明されているように、AWS は、 AWS クラウド のすべてを実行するグローバルインフラストラクチャを保護する責任を担います。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理

を維持する責任があります。このコンテンツには、使用する AWS サービスのセキュリティ設定および管理タスクが含まれます。データプライバシーの詳細については、「データプライバシーのよくある質問」を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログの「The AWS Shared Responsibility Model and GDPR」を参照してください。

データを保護するため、 AWS アカウント の認証情報を保護し、AWS Identity and Access Management (IAM) を使用して個々のユーザーアカウントをセットアップすることをお勧めします。この方法により、それぞれの職務を遂行するために必要な許可のみを各ユーザーに付与できます。また、以下の方法でデータを保護することをお勧めします。

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 以降が推奨されています。
- AWS CloudTrail で API とユーザーアクティビティログをセットアップします。
- AWS 暗号化ソリューションを、AWS サービス内のすべてのデフォルトのセキュリティ管理と一緒に使用します。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これにより、Amazon S3 に保存される個人データの検出と保護が支援されます。
- コマンドラインインターフェイスまたは API を使用して AWS にアクセスするときに FIPS 140-2 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。使用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-2」を参照してください。

顧客の E メールアドレスなどの機密情報やセンシティブ情報は、タグや [名前] フィールドなどの自由形式のフィールドに配置しないことを強くお勧めします。これは、コンソール、API、AWS CLI、または AWS SDK によって Amazon VPC や他の AWS のサービスを使用する場合も同様です。タグまたは名前に使用する自由形式のフィールドに入力したデータは、請求ログまたは診断ログに使用できます。外部サーバーへの URL を指定する場合は、そのサーバーへのリクエストを検証するための認証情報を URL に含めないことを強くお勧めします。

Amazon VPC でのインターネットワークトラフィックのプライバシー

Amazon Virtual Private Cloud では、次の機能を使用して、仮想プライベートクラウド (VPC) のセキュリティを強化し、モニタリングできます。

- セキュリティグループ: セキュリティグループは、関連付けられた Amazon EC2 インスタンスのファイアウォールとして動作し、インバウンドトラフィックとアウトバウンドトラフィックの両方をインスタンスレベルでコントロールします。インスタンスを起動する際、作成した 1 つ以上のセキュリティグループに関連付けることができます。VPC 内のインスタンスごとに異なるセキュリティグループのセットに割り当てることができます。インスタンスを起動するときにセキュリティグループを指定しないと、そのインスタンスは VPC のデフォルトのセキュリティグループに自動的に関連付けられます。詳細については、「」を参照してくださいVPC のセキュリティグループ (p. 188)
- ネットワークアクセスコントロールリスト (ACL): ネットワーク ACL は、関連付けられたサブネットのファイアウォールとして動作し、インバウンドトラフィックとアウトバウンドトラフィックの両方をサブネットレベルでコントロールします。詳細については、「」を参照してくださいネットワーク ACL (p. 200)
- フローログ: フローログは、 のネットワークインターフェイスとの間で行き来する IP トラフィックに関する情報をキャプチャします。VPC、サブネット、または個々のネットワークインターフェイスのフローログを作成できます。フローログデータは、CloudWatch Logs または Amazon S3 に発行され、過度に制限されているか制限のないセキュリティグループとネットワーク ACL ルールを診断するうえで役立ちます。詳細については、「」を参照してくださいVPC フローログ (p. 325)
- トラフィックのミラーリング: Amazon EC2 インスタンスの Elastic Network Interface からネットワークトラフィックをコピーできます。その後、トラフィックを帯域外セキュリティアプライアンスおよびモ

ニタリングアプライアンスに送信できます。詳細については、「トラフィックミラーリングガイド」を参照してください。

AWS Identity and Access Management (IAM) を使用すると、組織内の誰がセキュリティグループ、ネットワーク ACL、およびフローログを作成/管理できるようにするかをコントロールできます。たとえば、ネットワーク管理者にそのアクセス許可を付与し、インスタンスの起動のみが必要な作業員には付与しないようにできます。詳細については、「」を参照してくださいAmazon VPC の Identity and Access Management (p. 169)

Amazon セキュリティグループとネットワーク ACL は、次の Amazon サービスとの間で送受信されるトラフィックをフィルタリングしません。

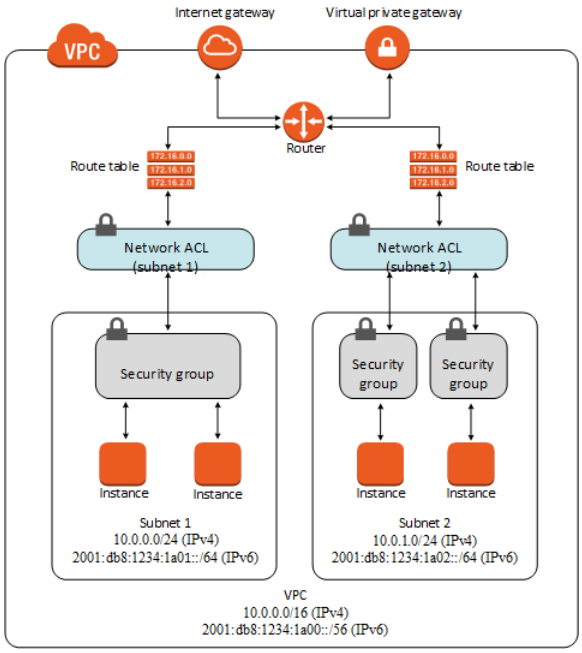
- Amazon ドメインネームサービス (DNS)
- Amazon Dynamic Host Configuration Protocol (DHCP)
- Amazon EC2 インスタンスメタデータ。
- Amazon Windows ライセンスアクティベーション
- Amazon Time Sync Service のご紹介
- デフォルト VPC ルーターの予約済み IP アドレス

セキュリティグループとネットワーク ACL を比較する

次の表は、セキュリティグループとネットワーク ACL の基本的な違いをまとめたものです。

セキュリティグループ	ネットワーク ACL
インスタンスレベルで動作します。	サブネットレベルで動作します。
ルールの許可のみがサポートされます	ルールの許可と拒否がサポートされます
ステートフル: ルールに関係なく、返されたトラフィックが自動的に許可されます	ステートレス: 返されたトラフィックがルールによって明示的に許可されます
トラフィックを許可するかどうかを決める前に、すべてのルールを評価します	トラフィックを許可するかどうかを決定する際に、最も低い番号のルールから順にルールを処理します。
インスタンスの起動時に誰かがセキュリティグループを指定した場合、または後でセキュリティグループをインスタンスに関連付けた場合にのみ、インスタンスに適用されます。	関連付けられているサブネット内のすべてのインスタンスに自動的に適用されます (そのため、セキュリティグループのルールの許容範囲が広すぎる場合は、保護レイヤーを追加する必要があります) 。

次の図は、セキュリティグループおよびネットワーク ACL が提供するセキュリティレイヤーを示しています。たとえば、インターネットゲートウェイからのトラフィックは、ルーティングテーブルのルートを使用して適切なサブネットにルーティングされます。サブネットに対してどのトラフィックが許可されるかは、そのサブネットに関連付けられているネットワーク ACL のルールによってコントロールされます。インスタンスに対してどのトラフィックが許可されるかは、その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によってコントロールされます。



セキュリティグループのみを使用してインスタンスを保護できます。ただし、ネットワーク ACL は追加の防御レイヤーとして追加できます。例については、「例: サブネットのインスタンスへのアクセス制御 (p. 217)」を参照してください。

転送時の暗号化

AWS では、すべてのタイプの EC2 インスタンス間において安全でプライベートな接続を提供しています。さらに、一部のインスタンスタイプでは、基盤となる Nitro System ハードウェアのオフロード機能を使用して、インスタンス間の転送中のトラフィックを自動的に暗号化します。詳細については、「Linux インスタンス用の Amazon EC2 ユーザーガイド」の「転送中の暗号化」を参照してください。

Amazon VPC のインフラストラクチャセキュリティ

マネージドサービスである Amazon VPC は、ホワイトペーパー「アマゾン ウェブ サービス: セキュリティプロセスの概要」に記載されている AWS グローバルネットワークセキュリティの手順で保護されています。

AWS が公開した API 呼び出しを使用して、ネットワーク経由で Amazon VPC にアクセスします。クライアントでは、Transport Layer Security (TLS) 1.0 以降がサポートされている必要があります。TLS 1.2 以降が推奨されています。また、Ephemeral Diffie-Hellman (DHE) や Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) などの Perfect Forward Secrecy (PFS) を使用した暗号スイートもクライアントでサポートされている必要があります。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

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

ネットワークの隔離

仮想プライベートクラウド (VPC) は、AWS クラウド内の論理的に隔離された領域にある仮想ネットワークです。ワークロードまたは組織エンティティ単位でインフラストラクチャを隔離するには、個別の VPC を使用します。

サブネットは、ある範囲の IP アドレスが示す VPC 内の領域です。インスタンスを起動する場合には、VPC 内のあるサブネットにおいて起動することになります。サブネットを使用すると、単一の VPC 内で多階層ウェブアプリケーションの各階層 (ウェブサーバー、アプリケーションサーバーおよびデータベースサーバーなど) を隔離できます。インターネットからの直接アクセスを認めるべきでないインスタンスには、プライベートサブネットを使用します。

パブリックインターネットを介してトラフィックを送信せずに VPC から Amazon EC2 API を呼び出すには、 AWS PrivateLink を使用します。

ネットワークトラフィックの制御

EC2 インスタンスへのネットワークトラフックを制御するには、以下のオプションを検討します。

- the section called “セキュリティグループ” (p. 188) を使用してサブネットへのアクセスを制限します。この方法を使うと、たとえば、社内ネットワークのアドレス範囲に属するアドレスからのトラフィックのみ認めるといったことができます。
- VPC へのネットワークアクセスをコントロールするための主要なメカニズムとして、セキュリティグループを活用します。必要に応じて、ネットワーク ACL を控えめに使用して、ステートレスできめの粗いネットワークコントロールを提供します。セキュリティグループは、ステートフルなパケットフィルタ処理を実行でき、他のセキュリティグループを参照するルールを作成できるため、ネットワーク ACL よりも汎用性があります。ただし、ネットワーク ACL は、トラフィックの特定のサブセットを拒否したり、高レベルのサブネットガードレールを提供したりするための、セカンダリコントロールとして効果的です。また、ネットワーク ACL はサブネット全体に適用されるため、多層防御として使用して、正しいセキュリティグループなしでインスタンスが意図せずに起動される事態に備えることができます。
- インターネットからの直接アクセスを認めるべきでないインスタンスには、プライベートサブネットを使用します。プライベートサブネット内にあるインスタンスからのインターネットアクセスに、要塞ホストまたは NAT ゲートウェイを使用する。
- 最小限必要なネットワークルートを使用して Amazon VPC サブネットルートテーブルを設定します。例えば、インターネットゲートウェイへのルートがあるサブネットに、インターネットへの直接アクセスを必要とする Amazon EC2 インスタンスのみを配置し、仮想プライベートゲートウェイへのルートがあるサブネットに、内部ネットワークへの直接アクセスを必要とする Amazon EC2 インスタンスのみを配置します。
- 追加のセキュリティグループまたはネットワークインターフェイスを使用して、Amazon EC2 インスタンス管理トラフィックを通常のアプリケーショントラフィックとは別に制御および監査することをご検討ください。このアプローチにより、顧客は変更管理用の特別な IAM ポリシーを実装できるため、セキュリティグループルールや自動化されたルール検証スクリプトに対する変更の監査が容易になります。複数のネットワークインターフェイスでは、ホストベースのルーティングポリシーを作成したり、サブネットに割り当てられたネットワークインターフェイスに基づいて異なる VPC サブネットルーティングルールを活用したりするなど、ネットワークトラフィックを制御するための追加のオプションも提供されます。
- AWS Virtual Private Network または AWS Direct Connectを使用して、リモートネットワークから VPC へのプライベート接続を確立する。詳細については、ネットワークから Amazon VPC への接続オプションを参照してください。
- VPC フローログを使用して、インスタンスに到達するトラフィックをモニタリングします。
- AWS Security Hubを使用して、インスタンスからの意図しないネットワークアクセスを確認する。

Amazon VPC は、各 Amazon EC2 インスタンスへのネットワークアクセスを制限することに加えて、追加のネットワークセキュリティコントロールの実装をサポートしています。詳細については、「ネットワークの保護」を参照してください。

Amazon VPC の Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全にコントロールするために役立つ AWS のサービスです。IAM 管理者は、誰を認証 (サインイン) し、誰に Amazon VPC リソースの使用を承認する (アクセス許可を付与する) かを制御します。IAM は、追加料金なしで使用できる AWS のサービスです。

目次

Audience

AWS Identity and Access Management (IAM) の用途は、Amazon VPC で行う作業によって異なります。

サービスユーザー – ジョブを実行するために Amazon VPC サービスを使用する場合は、管理者から必要なアクセス許可と認証情報が与えられます。さらに多くの Amazon VPC 機能を使用して作業を行う場合は、追加のアクセス許可が必要になることがあります。アクセスの管理方法を理解すると、管理者から適切なアクセス許可をリクエストするのに役に立ちます。Amazon VPC の機能にアクセスできない場合は、「Amazon VPC の ID とアクセスのトラブルシューティング (p. 183)」を参照してください。

サービス管理者 – 社内の Amazon VPC リソースを担当している場合は、通常、Amazon VPC へのフルアクセスがあります。管理者は、従業員にアクセスを許可する Amazon VPC 機能とリソースを決定します。サービスユーザーのアクセス許可を変更するリクエストを IAM 管理者に送信します。IAM の基本概念については、このページの情報を確認します。会社で Amazon VPC を使用して IAM を利用する方法の詳細については、「Amazon VPC で IAM を使用する方法 (p. 173)」を参照してください。

IAM 管理者 – 管理者は、Amazon VPC へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。ポリシーの例を表示するには、「Amazon VPC ポリシーの例 (p. 177)」を参照してください。

ID で認証する

認証は、アイデンティティ認証情報を使用して AWS にサインインする方法です。AWS Management Console を使用したサインインの詳細については、IAM ユーザーガイドの「IAM ユーザーまたはルートユーザーとしての AWS Management Console へのサインイン」を参照してください。

AWS アカウント のルートユーザーもしくは IAM ユーザーとして、または IAM ロールを引き受けることによって認証を受ける (AWS にサインインする) 必要があります。会社のシングルサインオン認証を使用することも、Google や Facebook を使用してサインインすることもできます。このような場合、管理者は以前に IAM ロールを使用して ID フェデレーションを設定しました。他の会社の認証情報を使用して AWS にアクセスした場合、ロールを間接的に割り当てられています。

AWS Management Console に直接サインインするには、ルートユーザーの E メールまたは IAM ユーザー名とパスワードを使用します。ルートユーザーまたは IAM ユーザーのアクセスキーを使用して AWS にプログラム的にアクセスできます。AWS は、ユーザーの認証情報を使用してリクエストに暗号的で署名するための SDK とコマンドラインツールを提供します。AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。これには、インバウンド API リクエストを認証するためのプロトコル、署名バージョン 4 を使用します。リクエストの認証の詳細については、AWS の全般リファレンスの「署名バージョン 4 の署名プロセス」を参照してください。

使用する認証方法を問わず、追加のセキュリティ情報の提供を要求される場合もあります。例えば、AWS では多要素認証 (MFA) を使用してアカウントのセキュリティを高めることを推奨しています。詳細については、IAM ユーザーガイドの「AWS での多要素認証 (MFA) の使用」を参照してください。

AWS アカウント ルートユーザー

AWS アカウント を初めて作成する場合は、このアカウントのすべての AWS サービスとリソースに対して完全なアクセス権限を持つシングルサインインアイデンティティで始めます。このアイデンティティは AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることによってアクセスできます。強くお勧めしているのは、日常的なタスクには、それが管理者タスクであっても、ルートユーザーを使用しないことです。代わりに、最初の IAM ユーザーを作成するためにのみ、ルートユーザーを使用するというベストプラクティスに従います。その後、ルートユーザーの認証情報を安全な場所に保管し、それらを使用して少数のアカウントおよびサービス管理タスクのみを実行します。

IAM ユーザーとグループ

IAM ユーザーは、単一のユーザーまたはアプリケーションに対する特定の許可を持つ AWS アカウント 内のアイデンティティです。IAM ユーザーは、ユーザー名とパスワード、アクセスキーのセットなど、長期的な認証情報を持つことができます。アクセスキーを生成する方法の詳細については、IAM ユーザーガイドの「IAM ユーザーのアクセスキーの管理」を参照してください。IAM ユーザーにアクセスキーを生成するとき、必ずキーペアを表示して安全に保存してください。後になって、シークレットアクセスキーを回復することはできません。新しいアクセスキーペアを生成する必要があります。

IAM グループは、IAM ユーザーのコレクションを指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、一度に複数のユーザーに対してアクセス許可を指定できます。多数の組のユーザーがある場合、グループを使用すると管理が容易になります。例えば、IAMAdminsという名前のグループを設定して、そのグループに IAM リソースを管理するアクセス許可を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の特定の人またはアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時的な認証情報が利用できます。詳細については、IAM ユーザーガイドの「IAM ユーザーの作成が適している場合 (ロールではなく)」を参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可を持つ、 AWS アカウント 内のアイデンティティです。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。ロールを切り替えることによって、AWS Management Console で IAM ロールを一時的に引き受けることができます。ロールを引き受けるには、AWS CLI または AWS API オペレーションを呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、IAM ユーザーガイドの IAM ロールの使用を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます。

- 一時的な IAM ユーザーアクセス許可 – IAM ユーザーは、特定のタスクに対して複数の異なるアクセス許可を一時的に IAM ロールで引き受けることができます。
- フェデレーティッドユーザーアクセス – IAM ユーザーを作成する代わりに、AWS Directory Service、エンタープライズユーザーディレクトリ、またはウェブ ID プロバイダーからの既存のアイデンティティを使用できます。このようなユーザーはフェデレーティッドユーザーと呼ばれます。AWS では、ID プロ

バイダーを通じてアクセスがリクエストされたとき、フェデレーティッドユーザーにロールを割り当てます。フェデレーティッドユーザーの詳細については、IAM ユーザーガイドのフェデレーティッドユーザーとロールを参照してください。

- クロスアカウントアクセス – IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを別のアカウントの人物 (信頼済みプリンシパル) に許可できます。ロールは、クロスアカウントアクセスを許可する主な方法です。ただし、一部の AWS のサービスでは、(ロールをプロキシとして使用する代わりに) リソースにポリシーを直接アタッチできます。クロスアカウントアクセスでのロールとリソースベースのポリシーの違いの詳細については、IAM ユーザーガイドの IAM ロールとリソースベースのポリシーとの相違点を参照してください。
- クロスサービスアクセス – 一部の AWS のサービスは、AWS の他のサービスの機能を使用します。例えば、サービスで呼び出しを行う場合、そのサービスでは Amazon EC2 でアプリケーションを実行したり、Amazon S3 にオブジェクトを保存したりするのが一般的です。サービスは、呼び出し元プリンシパルのアクセス許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
  - プリンシパル許可 – IAM ユーザーまたはロールを使用して AWS でアクションを実行する場合、そのユーザーはプリンシパルとみなされます。ポリシーは、プリンシパルにアクセス許可を付与します。一部のサービスを使用する場合、別のサービスで別のアクションをトリガーするアクションを実行することがあります。この場合、両方のアクションを実行するためのアクセス許可が必要です。アクションがポリシーで追加の依存アクションを必要とするかどうかを確認するには、サービス認証リファレンスの「Amazon Elastic Compute Cloud のアクション、リソース、および条件キー」をご参照ください。
  - サービスロール – サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの「AWS のサービスにアクセス権限を委任するロールの作成」を参照してください。
  - サービスリンクロール – サービスリンクロールは、AWS のサービスにリンクされたサービスロールの一種です。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは、IAM アカウント内に表示され、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。
- Amazon EC2 で実行されているアプリケーション – EC2 インスタンスで実行され、AWS CLI または AWS API リクエストを作成しているアプリケーションの一時的な認証情報を管理するには、IAM ロールを使用できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時認証情報を取得することができます。詳細については、IAM ユーザーガイドの「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用してアクセス許可を付与する」を参照してください。

IAM ロールまたは IAM ユーザーを使用するべきかどうかについては、IAM ユーザーガイドの IAM ユーザーの作成が適している場合 (ロールではなく) を参照してください。

ポリシーを使用してアクセスを管理する

AWS でのアクセスは、ポリシーを作成し、それらを IAM アイデンティティまたは AWS リソースにアタッチすることで制御できます。ポリシーは AWS のオブジェクトであり、ID やリソースに関連付けて、これらのアクセス許可を定義します。ルートユーザーまたは IAM ユーザーとしてサインインすることも、IAM ロールを引き受けることもできます。その後リクエストを行うと、AWS が関連するアイデンティティベースまたはリソースベースのポリシーを評価します。ポリシーでのアクセス許可により、リクエストが許可されるか拒否されるかが決まります。大半のポリシーは JSON ドキュメントとして AWS に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、IAM ユーザーガイドの JSON ポリシー概要を参照してください。

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

すべての IAM エンティティ (ユーザーまたはロール) は、アクセス許可のない状態からスタートします。言い換えると、デフォルト設定では、ユーザーは何もできず、自分のパスワードを変更することすらできません。何かを実行するアクセス許可をユーザーに付与するには、管理者がユーザーにアクセス許可ポリシーをアタッチする必要があります。また、管理者は、必要なアクセス許可があるグループにユーザーを追加できます。管理者がグループにアクセス許可を付与すると、そのグループ内のすべてのユーザーにこれらのアクセス許可が付与されます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションのアクセス許可を定義します。例えば、`iam:GetRole` アクションを許可するポリシーがあるとします。このポリシーがあるユーザーは、AWS Management Console、AWS CLI、または AWS API からロールの情報を取得できます。

ID ベースのポリシー

アイデンティティベースのポリシーは、IAM user ユーザー、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件を制御します。アイデンティティベースのポリシーを作成する方法については、IAM ユーザーガイドの「IAM ポリシーの作成」を参照してください。

アイデンティティベースのポリシーは、さらにインラインポリシーまたは管理ポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、 AWS アカウント 内の複数のユーザー、グループ、およびロールにアタッチできるスタンドアロンポリシーです。管理ポリシーには、AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。管理ポリシーまたはインラインポリシーのいずれかを選択する方法については、IAM ユーザーガイドの管理ポリシーとインラインポリシーの比較を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースにアタッチする JSON ポリシードキュメントです。リソースベースのポリシーの例は、IAM ロールの信頼ポリシーおよび Amazon S3 バケットポリシーです。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、ポリシーは、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件を定義します。リソースベースのポリシーで、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または AWS のサービスを含めることができます。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーで IAM の AWS 管理ポリシーを使用することはできません。

アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、AWS WAF、および Amazon VPC は、ACL をサポートするサービスの例です。ACL の詳細については、Amazon Simple Storage Service デベロッパーガイドの「アクセスコントロールリスト (ACL) の概要」を参照してください。

その他のポリシータイプ

AWS では、別のあまり一般的ではないポリシータイプもサポートしています。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の許可を設定できます。

- 許可の境界 – 許可の境界は、ID ベースのポリシーが IAM エンティティ (IAM ユーザーまたはロール) に付与できる許可の上限を設定する高度な機能です。エンティティの許可の境界を設定できます。結果として得られる許可は、エンティティの ID ベースのポリシーとその許可の境界の共通部分で

す。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーは、許可の境界では制限されません。これらのポリシーのいずれかを明示的に拒否した場合、その許可は無効になります。許可の境界の詳細については、IAM ユーザーガイドの「IAM エンティティの許可の境界」を参照してください。

- サービスコントロールポリシー (SCP) – SCP は、AWS Organizations で組織や組織単位 (OU) に最大アクセス許可を指定する JSON ポリシーです。AWS Organizations は、お客様のビジネスが所有する複数の AWS アカウント をグループ化し、一元的に管理するサービスです。組織内のすべての機能を有効にすると、サービス制御ポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP はメンバーアカウントのエンティティ (各 AWS アカウント ルートユーザーなど) に対するアクセス許可を制限します。Organizations と SCP の詳細については、AWS Organizations ユーザーガイドの「 SCP の仕組み」を参照してください。
- セッションポリシー – セッションポリシーは、ロールまたはフェデレーティッドユーザーの一時セッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果として得られるセッションの許可は、ユーザーまたはロールの ID ベースのポリシーとセッションポリシーの共通部分です。また、リソースベースのポリシーから許可が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、その許可は無効になります。詳細については、IAM ユーザーガイドの「セッションポリシー」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される許可を理解するのがさらに複雑になります。複数のポリシータイプが関連するとき、リクエストを許可するかどうかを AWS が決定する方法の詳細については、IAM ユーザーガイドの「ポリシーの評価ロジック」を参照してください。

Amazon VPC で IAM を使用する方法

IAM を使用して Amazon VPC へのアクセスを管理する前に、Amazon VPC で使用できる IAM 機能について理解しておく必要があります。Amazon VPC およびその他の AWS のサービスが IAM と連携する方法の概要を把握するには、IAM ユーザーガイドの「IAM と連携する AWS のサービス」を参照してください。

目次

IAM アイデンティティベースのポリシーでは、許可されるアクションまたは拒否されるアクションを指定できます。一部のアクションでは、アクションを許可または拒否するリソースと条件を指定できます。Amazon VPC は、特定のアクション、リソース、および条件キーをサポートしています。JSON ポリシーで使用するすべての要素については、IAM ユーザーガイドの IAM JSON ポリシーの要素のリファレンスを参照してください。

Actions

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの `Action` 要素は、ポリシー内のアクセスを許可または拒否するために使用できるアクションを記述します。ポリシーアクションの名前は通常、関連する AWS API オペレーションと同じで

す。一致する API オペレーションを持たないアクセス許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

Amazon VPC は、その API 名前空間を Amazon EC2 と共有します。Amazon VPC のポリシーアクションは、アクションの前にプレフィックス `ec2:` を使用します。たとえば、Amazon EC2 `CreateVpc` API オペレーションを使用して VPC を作成するアクセス許可を付与するには、ポリシーに `ec2:CreateVpc` アクションを含めます。ポリシーステートメントには、`Action` または `NotAction` エレメントを含める必要があります。

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

```
"Action": [
      "ec2:action1",
      "ec2:action2"
]
```

ワイルドカード (*) を使用して複数のアクションを指定することができます。たとえば、`Describe` という単語で始まるすべてのアクションを指定するには、以下のアクションを含めます。

```
"Action": "ec2:Describe*"
```

Amazon VPC アクションのリストを表示するには、IAM ユーザーガイドの「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。

Resources

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスするかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

`Resource` JSON ポリシーエレメントは、アクションが適用されるオブジェクトを指定します。ステートメントには、`Resource` または `NotResource` エレメントを含める必要があります。ベストプラクティスとして、リソースは Amazon リソースネーム (ARN) を使用して指定します。これは、リソースレベルのアクセス許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルのアクセス許可をサポートしないアクションの場合は、ワイルドカード (*) を使用して、ステートメントがすべてのリソースに適用されることを示します。

```
"Resource": "*"
```

> **Important**
>
> 現在、すべての Amazon EC2 API アクションがリソースレベルのアクセス許可をサポートしているわけではありません。Amazon EC2 API アクションでリソースレベルのアクセス許可がサポートされない場合、アクションを使用するアクセス許可をユーザーに付与できますが、ポリシーステートメントのリソース要素として * を指定する必要があります。リソース要素の ARN を指定できるアクションを確認するには、「Amazon EC2 で定義されるアクション」を参照してください。

VPC リソースには、次の例に示す ARN があります。

```
arn:${Partition}:ec2:${Region}:${Account}:vpc/${VpcId}
```

ARN の形式の詳細については、「Amazon リソースネーム (ARN)」を参照してください。

たとえば、ステートメントで vpc-1234567890abcdef0 VPC を指定するには、次の例に示す ARN を使用します。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:vpc/vpc-1234567890abcdef0"
```

特定のアカウントに属する特定のリージョン内のすべての VPC を指定するには、ワイルドカード (*) を使用します。

```
"Resource": "arn:aws:ec2:us-east-1:123456789012:vpc/*"
```

リソースの作成など、一部の Amazon VPC アクションは、特定のリソースで実行できません。このような場合は、ワイルドカード (*) を使用する必要があります。

```
"Resource": "*"
```

Amazon EC2 API アクションの多くが複数のリソースと関連します。複数のリソースを単一のステートメントで指定するには、ARN をカンマで区切ります。

```
"Resource": [
      "resource1",
      "resource2"
]
```

Amazon VPC リソースタイプとその ARN のリストを表示するには、IAM ユーザーガイドの「Amazon EC2 で定義されるリソース」を参照してください。

条件キー

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition エレメント (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Conditionエレメントはオプションです。イコールや以下などの条件演算子を使用する条件式を作成して、リクエスト内に値のあるポリシーの条件に一致させることができます。

1 つのステートメントに複数の Condition エレメントを指定する場合、または 1 つの Condition エレメントに複数のキーを指定する場合、AWS が論理 AND 演算を使用してそれらを評価します。単一の条件キーに複数の値を指定する場合、AWS が論理 OR 演算を使用して条件を評価します。ステートメントのアクセス許可が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば、IAM ユーザー名でタグ付けされている場合のみ、リソースにアクセスする IAM ユーザーアクセス許可を付与できます。詳細については、IAM ユーザーガイドの IAM ポリシーエレメント: 変数およびタグを参照してください。

AWS はグローバル条件キーとサービス固有の条件キーをサポートしています。すべての AWS グローバル条件キーを確認するには、IAM ユーザーガイドの「AWS グローバル条件コンテキストキー」を参照してください。

Amazon VPC は独自の条件キーを定義し、一部のグローバル条件キーの使用をサポートしています。すべての AWS グローバル条件キーを確認するには、IAM ユーザーガイドの「AWS グローバル条件コンテキストキー」を参照してください。

すべての Amazon EC2 アクションは、aws:RequestedRegion および ec2:Region 条件キーをサポートします。詳細については、「例: 特定のリージョンへのアクセスの制限」を参照してください。

Amazon VPC 条件キーのリストについては、IAM ユーザーガイドの「Amazon EC2 の条件キー」を参照してください。どのアクションおよびリソースと条件キーを使用できるかについては、「Amazon EC2 で定義されるアクション」を参照してください。

Amazon VPC リソースベースのポリシー

リソースベースのポリシーとは、Amazon VPC リソース上で指定するプリンシパルとしてのどのアクションをどの条件で実行できるかを指定する JSON ポリシードキュメントです。

クロスアカウントアクセスを有効にするには、全体のアカウント、または別のアカウントの IAM エンティティを、リソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる AWS アカウントにある場合は、リソースにアクセスするためのアクセス許可をプリンシパルエンティティにも付与する必要があります。アクセス許可は、アイデンティティベースのポリシーをエンティティにアタッチすることで付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、ID ベースのポリシーをさらに付与する必要はありません。詳細については、IAM ユーザーガイドの「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。

タグに基づいた承認

タグを Amazon VPC リソースにアタッチするか、リクエストでタグを渡すことができます。タグに基づいてアクセスを制御するには、`ec2:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。詳細については、Amazon EC2 ユーザーガイドの「Tagging に関するリソースレベルのアクセス許可」を参照してください。

リソースのタグに基づいてリソースへのアクセスを制限するためのアイデンティティベースのポリシーの例を表示するには、「特定の VPC 内にインスタンスを起動する (p. 182)」を参照してください。

IAM; ロール

IAM ロールは、特定のアクセス許可を持つ AWS アカウント内のエンティティです。

一時認証情報の使用

一時的な認証情報を使用して、フェデレーションでサインイン、IAM ロールを引き受ける、またはクロスアカウントロールを引き受けることができます。一時的なセキュリティ認証情報を取得するには、AssumeRole または GetFederationToken などの AWS STS API オペレーションを呼び出します。

Amazon VPC では、一時認証情報の使用をサポートしています。

サービスにリンクされたロール

サービスにリンクされたロールによって、AWS サービスが他のサービスのリソースにアクセスして自動的にアクションを完了できます。サービスにリンクされたロールは、IAM アカウント内に表示され、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールのアクセス許可を表示できますが、編集することはできません。

トランジットゲートウェイは、サービスにリンクされたロールをサポートします。

サービスロール

この機能では、サービスのロールをユーザーに代わって引き受けることをサービスに許可します。このロールにより、サービスはユーザーに代わって他のサービスのリソースにアクセスし、アクションを実行できます。サービスロールは、IAM アカウントに表示され、サービスによって所有されます。つま

り、IAM 管理者は、このロールのアクセス許可を変更できます。ただし、これを行うことにより、サービスの機能が損なわれる場合があります。

Amazon VPC では、フローログのサービスロールがサポートされています。フローログを作成するときは、フローログサービスへ CloudWatch Logs のアクセスを許可するロールを選択する必要があります。詳細については、「」を参照してくださいCloudWatch Logs へのフローログ発行のための IAM ロール (p. 337)

Amazon VPC ポリシーの例

デフォルトでは、IAM ユーザーおよびロールには、VPC リソースを作成または変更するアクセス許可はありません。また、AWS Management Console や AWS CLI、AWS API を使用してタスクを実行することもできません。IAM 管理者は、ユーザーとロールに必要な、指定されたリソースで特定の API オペレーションを実行するアクセス許可をユーザーとロールに付与する IAM ポリシーを作成する必要があります。続いて、管理者はそれらのアクセス許可が必要な IAM ユーザーまたはグループにそのポリシーをアタッチします。

JSON ポリシードキュメントのこれらの例を使用して、IAM アイデンティティベースのポリシーを作成する方法については、IAM ユーザーガイドの「JSON タブでのポリシーの作成」を参照してください。

目次

ポリシーのベストプラクティス

ID ベースのポリシーは非常に強力です。アカウント内で、Amazon VPC リソースを作成、アクセス、または削除できるかどうかを決定します。これらのアクションを実行すると、 AWS アカウント に追加料金が発生する可能性があります。アイデンティティベースのポリシーを作成または編集するときは、以下のガイドラインと推奨事項に従います。

- AWS 管理ポリシーを使用して開始する – Amazon VPC の使用をすばやく開始するには、AWS 管理ポリシーを使用して、従業員に必要なアクセス許可を付与します。これらのポリシーはアカウントですでに有効になっており、 によって管理および更新されていますAWS 詳細については、IAM ユーザーガイドの「AWS 管理ポリシーを使用したアクセス許可の使用開始」を参照してください。
- 最小権限を付与する – カスタムポリシーを作成するときは、タスクの実行に必要な許可のみを付与します。最小限のアクセス許可から開始し、必要に応じて追加のアクセス許可を付与します。この方法は、寛容なアクセス許可で始め、後でそれらを強化しようとするよりも安全です。詳細については、IAM ユーザーガイドの「最小限の特権を認める」を参照してください。
- 機密性の高い操作に MFA を有効にする – 追加セキュリティとして、機密性の高いリソースまたは API 操作にアクセスするために IAM ユーザーに対して、多要素認証 (MFA) の使用を要求します。詳細については、IAM ユーザーガイドの「AWS での多要素認証 (MFA) の使用」を参照してください。
- 追加のセキュリティとしてポリシー条件を使用する – 実行可能な範囲内で、ID ベースのポリシーでリソースへのアクセスを許可する条件を定義します。例えば、要求が発生しなければならない許容 IP アドレスの範囲を指定するための条件を記述できます。指定された日付または時間範囲内でのみリクエスト

を許可する条件を書くことも、SSL や MFA の使用を要求することもできます。詳細については、IAM ユーザーガイドの「IAM JSON ポリシー要素: 条件」を参照してください。

Amazon VPC コンソールを使用する

Amazon VPC コンソールにアクセスするには、一連の最小限のアクセス許可が必要です。これらのアクセス許可により、AWS アカウントの Amazon VPC リソースの詳細をリストおよび表示できます。最小限必要なアクセス許可よりも制限されたアイデンティティベースのポリシーを作成すると、そのポリシーをアタッチしたエンティティ (IAM ユーザーまたはロール) に対してはコンソールが意図したとおりに機能しません。

次のポリシーは、VPC コンソールでリソースを一覧表示するアクセス許可をユーザーに付与しますが、リソースを作成、更新、削除することはできません。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAddresses",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeClassicLinkInstances",
                "ec2:DescribeClientVpnEndpoints",
                "ec2:DescribeCustomerGateways",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeEgressOnlyInternetGateways",
                "ec2:DescribeFlowLogs",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeManagedPrefixLists",
                "ec2:DescribeMovingAddresses",
                "ec2:DescribeNatGateways",
                "ec2:DescribeNetworkAcls",
                "ec2:DescribeNetworkInterfaceAttribute",
                "ec2:DescribeNetworkInterfacePermissions",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribePrefixLists",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroupReferences",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeStaleSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeTags",
                "ec2:DescribeTrafficMirrorFilters",
                "ec2:DescribeTrafficMirrorSessions",
                "ec2:DescribeTrafficMirrorTargets",
                "ec2:DescribeTransitGateways",
                "ec2:DescribeTransitGatewayVpcAttachments",
                "ec2:DescribeTransitGatewayRouteTables",
                "ec2:DescribeVpcAttribute",
                "ec2:DescribeVpcClassicLink",
                "ec2:DescribeVpcClassicLinkDnsSupport",
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeVpcEndpointConnectionNotifications",
                "ec2:DescribeVpcEndpointConnections",
                "ec2:DescribeVpcEndpointServiceConfigurations",
                "ec2:DescribeVpcEndpointServicePermissions",
                "ec2:DescribeVpcEndpointServices",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeVpcs",
```

```
                "ec2:DescribeVpnConnections",
                "ec2:DescribeVpnGateways",
                "ec2:GetManagedPrefixListAssociations",
                "ec2:GetManagedPrefixListEntries"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、これらのユーザーに対して、実行する必要がある API オペレーションと一致するアクションにのみアクセスを許可します。

パブリックサブネットを持つ VPC を作成する

次の例では、ユーザーが VPC、サブネット、ルートテーブル、およびインターネットゲートウェイを作成できるようにします。ユーザーは、インターネットゲートウェイを VPC にアタッチし、ルートテーブルにルートを作成することもできます。`ec2:ModifyVpcAttribute` アクションにより、ユーザーは、VPC 内で起動される各インスタンスが DNS ホスト名を受け取ることができるように、VPC の DNS ホスト名を有効にできます。

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:CreateVpc",
        "ec2:CreateSubnet",
        "ec2:DescribeAvailabilityZones",
        "ec2:CreateRouteTable",
        "ec2:CreateRoute",
        "ec2:CreateInternetGateway",
        "ec2:AttachInternetGateway",
        "ec2:AssociateRouteTable",
        "ec2:ModifyVpcAttribute"
      ],
      "Resource": "*"
    }
   ]
}
```

前述のポリシーにより、ユーザーは、Amazon VPC コンソールで最初の VPC ウィザード設定オプションを使用して VPC を作成することもできます。VPC ウィザードを表示するには、ユーザーに `ec2:DescribeVpcEndpointServices` を使用するアクセス許可も必要です。これにより、VPC ウィザードの VPC エンドポイントセクションが正しく読み込まれます。

VPC リソースの変更と削除

ユーザーが変更または削除できる VPC リソースを制御することもできます。たとえば、次のポリシーでは、タグ `Purpose=Test` を持つルートテーブルの操作と削除をユーザーに許可します。また、このポリシーでは、ユーザーがタグ `Purpose=Test` を持つインターネットゲートウェイのみを削除できることを指定します。ユーザーは、このタグを持たないルートテーブルまたはインターネットゲートウェイを操作できません。

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

```
            "Action": "ec2:DeleteInternetGateway",
            "Resource": "arn:aws:ec2:*:*:internet-gateway/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Purpose": "Test"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteRouteTable",
                "ec2:CreateRoute",
                "ec2:ReplaceRoute",
                "ec2:DeleteRoute"
            ],
            "Resource": "arn:aws:ec2:*:*:route-table/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/Purpose": "Test"
                }
            }
        }
    ]
}
```

セキュリティグループの管理

次のポリシーでは、すべてのセキュリティグループとセキュリティグループのルールを表示できます。2 番目のステートメントでは、ユーザーがタグ `Stack=test` の付いたセキュリティグループを削除したり、タグ `Stack=test` の付いたセキュリティグループのインバウンドおよびアウトバウンドのルールを管理することを許可します。3 番目のステートメントでは、ユーザーが作成したセキュリティグループにタグ `Stack=Test` を付ける必要があります。4 番目のステートメントは、セキュリティグループの作成時に、タグを作成することをユーザーに許可します。

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeSecurityGroups",
         "ec2:DescribeSecurityGroupRules",
         "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:AuthorizeSecurityGroupIngress",
         "ec2:RevokeSecurityGroupIngress",
         "ec2:UpdateSecurityGroupRuleDescriptionsIngress",
         "ec2:AuthorizeSecurityGroupEgress",
         "ec2:RevokeSecurityGroupEgress",
         "ec2:UpdateSecurityGroupRuleDescriptionsEgress",
         "ec2:ModifySecurityGroupRules",
         "ec2:DeleteSecurityGroup"
      ],
      "Resource": "arn:aws:ec2:*:*:security-group/*",
      "Condition":{
         "StringEquals": {
            "ec2:ResourceTag/Stack": "test"
         }
```

```
      }
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateSecurityGroup"
      ],
      "Resource": "arn:aws:ec2:*:*:security-group/*",
      "Condition": {
         "StringEquals": {
           "aws:RequestTag/Stack": "test"
         },
         "ForAllValues:StringEquals": {
           "aws:TagKeys": ["Stack"]
         }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
         "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:security-group/*",
      "Condition": {
        "StringEquals": {
           "ec2:CreateAction" : "CreateSecurityGroup"
        }
      }
    }
  ]
}
```

インスタンスに関連付けられたセキュリティグループをユーザーが変更できるようにするには、ポリシーに `ec2:ModifyInstanceAttribute` アクションを追加します。

ユーザーがネットワークインターフェイスのセキュリティグループを変更できるようにするには、ポリシーに `ec2:ModifyNetworkInterfaceAttribute` アクションを追加します。

セキュリティグループルールの管理

次のポリシーは、セキュリティグループとセキュリティグループルールの表示、特定の VPC のセキュリティグループのインバウンドおよびアウトバウンドのルールの追加と削除、および指定された VPC のルールの説明を変更するアクセス許可をユーザーに付与します。1 番目のステートメントでは、`ec2:Vpc` 条件キーを使用して、特定の VPC に許可をスコープしています。

2 番目のステートメントは、すべてのセキュリティグループ、セキュリティグループルール、タグについて説明する許可をユーザーに与えます。これにより、ユーザーはセキュリティグループルールを表示して変更できるようになります。

```
{
  "Version": "2012-10-17",
  "Statement":[{
    "Effect":"Allow",
    "Action": [
       "ec2:AuthorizeSecurityGroupIngress",
       "ec2:RevokeSecurityGroupIngress",
       "ec2:UpdateSecurityGroupRuleDescriptionsIngress",
       "ec2:AuthorizeSecurityGroupEgress",
       "ec2:RevokeSecurityGroupEgress",
       "ec2:UpdateSecurityGroupRuleDescriptionsEgress",
       "ec2:ModifySecurityGroupRules"
    ],
```

```
     "Resource": "arn:aws:ec2:region:account:security-group/*",
      "Condition": {
        "ArnEquals": {
          "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-11223344556677889"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
          "ec2:DescribeSecurityGroups",
          "ec2:DescribeSecurityGroupRules",
          "ec2:DescribeTags"
      ],
      "Resource": "*"
    }
  ]
}
```

特定のサブネット内にインスタンスを起動する

以下のポリシーは、特定のサブネット内にインスタンスを起動し、リクエストで特定のセキュリティグループを使用する許可をユーザーに与えます。このポリシーは、subnet-11223344556677889 の ARN および sg-11223344551122334 の ARN を指定することで許可を与えます。ユーザーが別のサブネット内でまたは別のセキュリティグループを使用してインスタンスを起動しようとすると、リクエストは失敗します (ただし、別のポリシーまたは別の定義文で、ユーザーにその許可が与えられている場合を除きます)。

このポリシーは、ネットワークインターフェイスリソースを使用する許可も与えます。サブネット内に起動すると、RunInstances リクエストは、デフォルトでプライマリネットワークインターフェイスを作成するので、ユーザーには、インスタンスを起動するときにこのリソースを作成する許可が必要です。

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region::image/ami-*",
        "arn:aws:ec2:region:account:instance/*",
        "arn:aws:ec2:region:account:subnet/subnet-11223344556677889",
        "arn:aws:ec2:region:account:network-interface/*",
        "arn:aws:ec2:region:account:volume/*",
        "arn:aws:ec2:region:account:key-pair/*",
        "arn:aws:ec2:region:account:security-group/sg-11223344551122334"
      ]
    }
   ]
}
```

特定の VPC 内にインスタンスを起動する

以下のポリシーは、特定の VPC 内の任意のサブネットにインスタンスを起動する許可をユーザーに与えます。このポリシーは、条件キー (ec2:Vpc) をサブネットリソースに適用することで許可を与えます。

また、このポリシーは、タグ「department=dev」のある AMI のみを使用してインスタンスを起動する許可をユーザーに与えます。

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

```
   "Statement": [{
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:region:account:subnet/*",
        "Condition": {
         "StringEquals": {
            "ec2:Vpc": "arn:aws:ec2:region:account:vpc/vpc-11223344556677889"
            }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:region::image/ami-*",
      "Condition": {
         "StringEquals": {
            "ec2:ResourceTag/department": "dev"
            }
      }
   },
   {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
         "arn:aws:ec2:region:account:instance/*",
         "arn:aws:ec2:region:account:volume/*",
         "arn:aws:ec2:region:account:network-interface/*",
         "arn:aws:ec2:region:account:key-pair/*",
         "arn:aws:ec2:region:account:security-group/*"
         ]
      }
   ]
}
```

その他の Amazon VPC ポリシーの例

Amazon VPC に関連するその他の IAM ポリシーの例については、次のドキュメントを参照してください。

- ClassicLink
- マネージドプレフィックスリスト (p. 277)
- トラフィックのミラーリング
- トランジットゲートウェイ
- VPC エンドポイントおよび VPC エンドポイントサービス
- VPC エンドポイントポリシー
- VPC ピアリング接続
- AWS Wavelength

Amazon VPC の ID とアクセスのトラブルシューティング

次の情報は、Amazon VPC と IAM の使用に伴って発生する可能性がある一般的な問題の診断や修復に役立ちます。

問題点

- Amazon VPC でアクションを実行する権限がない (p. 184)
- iam:PassRole を実行する権限がない (p. 184)

Amazon VPC でアクションを実行する権限がない

AWS Management Console から、アクションを実行する権限がないと通知された場合、管理者に問い合わせ、サポートを依頼する必要があります。お客様のユーザー名とパスワードを発行したのが、担当の管理者です。

以下の例のエラーは、mateojackson IAM ユーザーがコンソールを使用して、サブネットの詳細を表示しようとしているが、ec2:DescribeSubnets アクセス許可がない場合に発生します。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 ec2:DescribeSubnets on resource: subnet-id
```

この場合、Mateo は、サブネットにアクセスできるように、ポリシーの更新を管理者に依頼します。

iam:PassRole を実行する権限がない

iam:PassRole アクションを実行する権限がないというエラーが表示された場合、管理者に問い合わせ、サポートを依頼する必要があります。お客様のユーザー名とパスワードを発行したのが、担当の管理者です。Amazon VPC にロールを渡すことができるようにポリシーを更新するよう、管理者に依頼します。

一部の AWS サービスでは、新しいサービスロールまたはサービスにリンクされたロールを作成せずに、既存のロールをサービスに渡すことができます。そのためには、サービスにロールを渡すアクセス許可が必要です。

以下の例のエラーは、marymajor という IAM ユーザーがコンソールを使用して Amazon VPC でアクションを実行しようする場合に発生します。ただし、アクションでは、サービスロールによって付与されたアクセス許可がサービスにある必要があります。メアリーには、ロールをサービスに渡すアクセス許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

この場合、メアリーは担当の管理者に iam:PassRole アクションを実行できるようにポリシーの更新を依頼します。

アクセスキーを表示する場合

IAM ユーザーアクセスキーを作成した後は、いつでもアクセスキー ID を表示できます。ただし、シークレットアクセスキーをもう一度表示することはできません。シークレットアクセスキーを紛失した場合は、新しいキーペアを作成する必要があります。

アクセスキーは、アクセスキー ID (AKIAIOSFODNN7EXAMPLE など) とシークレットアクセスキー (wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY など) の 2 つの部分から構成されます。ユーザー名とパスワードと同様に、リクエストを認証するために、アクセスキー ID とシークレットアクセスキーの両方を使用する必要があります。ユーザー名とパスワードと同様に、アクセスキーをしっかり管理してください。

> Important
>
> 正規ユーザー ID を確認するためであっても、アクセスキーをサードパーティーに提供しないでください。提供すると、第三者がアカウントへの永続的アクセスを取得する場合があります。

アクセスキーペアを作成する場合、アクセスキー ID とシークレットアクセスキーを安全な場所に保存するように求めるプロンプトが表示されます。このシークレットアクセスキーは、作成時にのみ使用できます。シークレットアクセスキーを紛失した場合、新しいアクセスキーを IAM ユーザーに追加する必要があります。最大 2 つのアクセスキーを持つことができます。すでに 2 つある場合は、新しいキーペアを作成する前に、いずれかを削除する必要があります。手順を表示するには、IAM ユーザーガイドのアクセスキーの管理を参照してください。

管理者として Amazon VPC へのアクセスを他のユーザーに許可したい

Amazon VPC へのアクセスを他のユーザーに許可するには、アクセスを必要とする人またはアプリケーションの IAM エンティティ (ユーザーまたはロール) を作成する必要があります。ユーザーは、このエンティティの認証情報を使用して にアクセスしますAWS 次に、Amazon VPC の適切なアクセス許可を付与するポリシーを、そのエンティティにアタッチする必要があります。

すぐに開始するには、IAM ユーザーガイドの IAM が委任した最初のユーザーおよびグループの作成を参照してください。

AWS アカウント以外のユーザーに Amazon VPC リソースへのアクセスを許可したい

他のアカウントのユーザーや組織外のユーザーが、リソースへのアクセスに使用できるロールを作成できます。ロールを引き受けるように信頼されたユーザーを指定することができます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください。

- Amazon VPC がこれらの機能をサポートしているかどうかについては、「Amazon VPC で IAM を使用する方法 (p. 173)」を参照してください。
- 所有している AWS アカウント 全体のリソースへのアクセス権を提供する方法については、IAM ユーザーガイドの「所有している別の AWS アカウント アカウントへのアクセス権を IAM ユーザーに提供」を参照してください。
- サードパーティーの AWS アカウント にリソースへのアクセス権を提供する方法については、IAM ユーザーガイドの「第三者が所有する AWS アカウント アカウントへのアクセス権を付与する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイドの「外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可」を参照してください。
- クロスアカウントアクセスでのロールとリソースベースのポリシーの使用の違いの詳細については、IAM ユーザーガイドの IAM ロールとリソースベースのポリシーとの相違点を参照してください。

Amazon Virtual Private Cloud の AWS 管理ポリシー

ユーザー、グループ、ロールにアクセス権限を追加するには、自分でポリシーを作成するよりも、AWS 管理ポリシーを使用する方が簡単です。チームに必要なアクセス許可のみを提供する IAM カスタマー管理ポリシーを作成するには、時間と専門知識が必要です。すぐに使用を開始するために、AWS 管理ポリシーを使用できます。これらのポリシーは、一般的なユースケースを対象範囲に含めており、AWS アカウントで利用できます。AWS管理ポリシーの詳細については、IAM ユーザーガイドのAWS管理ポリシーを参照してください。

AWSサービスは、AWS管理ポリシーを維持および更新します。AWS 管理ポリシーのアクセス許可を変更することはできません。サービスでは、新しい機能を利用できるようにするために、AWS管理ポリシーに

アクセス許可が追加されることがあります。このタイプの更新は、ポリシーがアタッチされているすべての ID (ユーザー、グループ、ロール) に影響します。新しい機能が立ち上げられた場合や、新しいオペレーションが使用可能になった場合に、各サービスがAWS 管理ポリシーを更新する可能性が最も高くなります。サービスは、AWS 管理ポリシーからアクセス許可を削除しないため、ポリシーの更新によって既存のアクセス許可が破棄されることはありません。

加えてAWS では、複数のサービスにまたがる職務機能のための管理ポリシーもサポートしています。例えば、ReadOnlyAccessAWS管理ポリシーでは、すべての AWS のサービスおよびリソースへの読み取り専用アクセスを許可します。あるサービスで新しい機能を立ち上げる場合は、AWS 追加されたオペレーションとリソースに対し、読み取り専用のアクセス許可を設定します。職務機能ポリシーのリストと説明については、IAM ユーザーガイドの「職務機能の AWS 管理ポリシー」を参照してください。

AWS 管理ポリシー: AmazonVPCFullAccess

`AmazonVPCFullAccess` ポリシーを IAM アイデンティティにアタッチできます。このポリシーは、Amazon VPC への完全なアクセスを可能にする許可を付与します。

このポリシーのアクセス権限を確認するには、AWS Management Console で「AmazonVPCFullAccess」を参照してください。

AWS 管理ポリシー: AmazonVPCReadOnlyAccess

`AmazonVPCReadOnlyAccess` ポリシーを IAM アイデンティティにアタッチできます。このポリシーは、Amazon VPC への読み取り専用アクセスを可能にする許可を付与します。

このポリシーのアクセス権限を確認するには、AWS Management Console で「AmazonVPCReadOnlyAccess」を参照してください。

Amazon VPC による AWS 管理ポリシーの更新

Amazon VPC の AWS 管理ポリシーに対する更新の詳細について、このサービスがこれらの変更の追跡を開始した 2021 年 3 月以降のものを表示します。

変更	説明	日付
the section called “AmazonVPCReadOnlyAccess” (p. 186) – 既存ポリシーへの更新	DescribeSecurityGroupRules アクションを追加しました。これにより、IAM ユーザーまたはロールがセキュリティグループルールを表示できるようになります。	2021 年 8 月 2 日
the section called “AmazonVPCFullAccess” (p. 186) – 既存ポリシーへの更新	DescribeSecurityGroupRules および ModifySecurityGroupRules アクションを追加しました。これにより、IAM ユーザーまたはロールがセキュリティグループルールを表示および変更できるようになります。	2021 年 8 月 2 日
the section called “AmazonVPCFullAccess” (p. 186) – 既存ポリシーへの更新	キャリアゲートウェイ、IPv6 プール、ローカルゲートウェイ、およびローカルゲートウェイルートテーブルに対するアクションが追加されました。	2021 年 6 月 23 日

変更	説明	日付
the section called “AmazonVPCReadOnlyAccess” (p. 136) – 既存ポリシーへの更新	キャリアゲートウェイ、IPv6 プール、ローカルゲートウェイ、およびローカルゲートウェイルートテーブルに対するアクションが追加されました。	2021 年 6 月 23 日

VPC のログとモニタリング

以下の自動化されたモニタリングツールを使用して、VPC のコンポーネントを監視し、問題が発生したときにレポートできます。

- フローログ: フローログは、 のネットワークインターフェイスとの間で行き来する IP トラフィックに関する情報をキャプチャします。VPC、サブネット、または個々のネットワークインターフェイスのフローログを作成できます。フローログデータは、CloudWatch Logs または Amazon S3 に発行され、過度に制限されているか制限のないセキュリティグループとネットワーク ACL ルールを診断するうえで役立ちます。詳細については、「VPC フローログ (p. 325)」を参照してください。
- NAT ゲートウェイのモニタリング: CloudWatch を使用して NAT ゲートウェイをモニタリングすることで、NAT ゲートウェイから情報を収集し、リアルタイムに近い読み取り可能なメトリクスに加工することができます。詳細については、「Amazon CloudWatch を使用した NAT ゲートウェイのモニタリング (p. 245)」を参照してください。

Amazon Virtual Private Cloud での耐障害性

AWS のグローバルインフラストラクチャは AWS リージョンとアベイラビリティーゾーンを中心として構築されます。AWS リージョンには、低レイテンシー、高スループット、そして高冗長性のネットワークで接続されている複数の物理的に独立・隔離されたアベイラビリティーゾーンがあります。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、および拡張性が優れています。

AWS リージョンとアベイラビリティーゾーンの詳細については、「AWS グローバルインフラストラクチャ」を参照してください。

Amazon VPC では、AWS グローバルインフラストラクチャに加えて、データの耐障害性とバックアップのニーズに対応できるように複数の機能を提供しています。

- Amazon VPC から Amazon VPC への接続オプション
- ネットワークから Amazon VPC への接続オプション

Amazon Virtual Private Cloud のコンプライアンス検証

サードパーティーの監査人は、SOC、PCI、FedRAMP、HIPAA など複数の AWS コンプライアンスプログラムの一環として、AWS サービスのセキュリティとコンプライアンスを評価します。

Amazon VPC のサービスや他の AWS のサービスが、特定のコンプライアンスプログラムの対象であるかどうかを確認するには、「コンプライアンスプログラムによる対象範囲内の AWS のサービス」「」を参照してください。一般的な情報については、「AWS コンプライアンスプログラム」を参照してください。

サードパーティーの監査レポートをダウンロードするには、AWS Artifact を使用します。詳細については、AWS Artifactにおけるレポートのダウンロードを参照してください。

AWS サービスを使用する際のお客様のコンプライアンス責任は、データの機密性、企業のコンプライアンス目的、適用法規によって決まります。AWS ではコンプライアンスに役立つ以下のリソースを用意しています。

- セキュリティおよびコンプライアンスのクイックスタートガイド - これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境を AWS でデプロイするための手順を説明します。
- HIPAA のセキュリティとコンプライアンスに関するホワイトペーパーを作成する - このホワイトペーパーは、企業が AWS を使用して HIPAA 準拠のアプリケーションを作成する方法について説明します。

  Note

  すべてのサービスが HIPAA に準拠しているわけではありません。
- AWS コンプライアンスのリソース - このワークブックおよびガイドのコレクションは、ユーザーの業界や地域で使用できるかもしれません。
- AWS Config デベロッパーガイドの「ルールでのリソースの評価」 - AWS Config サービスは、リソース設定が社内のプラクティス、業界のガイドライン、規制にどの程度準拠しているかを評価します。
- AWS Security Hub: この AWS サービスでは、AWS 内のセキュリティ状態を包括的に表示しており、セキュリティ業界の標準およびベストプラクティスへの準拠を確認するのに役立ちます。
- AWS Audit Manager - この AWS サービスでは AWSの使用状況を継続的に監査し、リスクの管理方法、規制や業界標準への準拠を簡素化できます。

Amazon Virtual Private Cloud での設定と脆弱性の分析

設定および IT 管理は、AWS とお客様の間で共有される責任です。詳細については、AWS 責任共有モデルを参照してください。責任共有モデルに加えて、VPC ユーザーは以下の点に留意する必要があります。

- 関連するクライアント側の依存関係を使用して、クライアントアプリケーションにパッチを適用するのはお客様の責任です。
- お客様は、NAT ゲートウェイおよび EC2 インスタンスの侵入テストを検討する必要があります (侵入テスト/ を参照) 。

VPC のセキュリティグループ

セキュリティグループは、インスタンスの仮想ファイアウォールとして機能し、インバウンドトラフィックとアウトバウンドトラフィックをコントロールします。VPC 内でインスタンスを起動した場合、そのインスタンスには最大 5 つのセキュリティグループを割り当てることができます。セキュリティグループは、サブネットレベルでなくインスタンスレベルで動作します。このため、VPC 内のサブネット内のインスタンスごとに異なるセキュリティグループのセットに割り当てることができます。

Amazon EC2 API またはコマンドラインツールを使用してインスタンスを起動するが、セキュリティグループを指定しない場合、そのインスタンスは VPC のデフォルトのセキュリティグループに自動的に割り当てられます。Amazon EC2 コンソールを使用してインスタンスを起動する場合、インスタンスの新しいセキュリティグループを作成するオプションを使用できます。

セキュリティグループごとに、インスタンスへのインバウンドトラフィックをコントロールするルールと、アウトバウンドトラフィックをコントロールする一連のルールを個別に追加します。このセクションでは、VPC のセキュリティグループとそのルールについて、知っておく必要がある基本事項について説明します。

セキュリティの追加レイヤーを VPC に追加するには、セキュリティグループと同様のルールを指定したネットワーク ACL をセットアップできます。セキュリティグループとネットワーク ACL の違いの詳細については、「セキュリティグループとネットワーク ACL を比較する (p. 166)」を参照してください。

目次

セキュリティグループの基本

セキュリティグループの特徴を次に示します。

- 許可ルールを指定できます。拒否ルールは指定できません。
- インバウンドトラフィックとアウトバウンドトラフィックのルールを個別に指定できます。
- セキュリティグループルールを使用すると、プロトコルとポート番号に基づいてトラフィックをフィルタリングできます。
- セキュリティグループはステートフルです。インスタンスからリクエストを送信する場合、そのリクエストのレスポンストラフィックは、インバウンドセキュリティグループのルールにかかわらず、流れることができます。許可されたインバウンドトラフィックに対する応答 (戻りのトラフィック) は、アウトバウンドルールにかかわらずアウト側に対し通過することができます。

  Note

  トラフィックのタイプにより、追跡の方法は異なります。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「接続追跡」を参照してください。
- セキュリティグループを初めて作成するときには、インバウンドルールはありません。したがって、インバウンドルールをセキュリティグループに追加するまで、別のホストからインスタンスに送信されるインバウンドトラフィックは許可されません。
- デフォルトでは、セキュリティグループにはすべてのアウトバウンドトラフィックを許可するアウトバウンドルールが含まれています。ルールを削除し、任意の発信トラフィックのみを許可するアウトバウンドルールを追加できます。セキュリティグループにアウトバウンドルールがない場合、インスタンスから送信されるアウトバウンドトラフィックは許可されません。
- VPC あたりの作成可能なセキュリティグループの数、各セキュリティグループに追加できるルールの数、ネットワークインターフェイスに関連付けることができるセキュリティグループの数にはクォータがあります。詳細については、「Amazon VPC クォータ (p. 362)」を参照してください。
- セキュリティグループに関連付けられたインスタンスの相互通信は、トラフィックを許可するルールを追加するまで許可されません (例外: デフォルトのセキュリティグループについては、このルールがデフォルトで指定されています) 。
- セキュリティグループはネットワークインターフェイスに関連付けられます。インスタンスを起動した後で、インスタンスに関連付けられたセキュリティグループを変更できます。これにより、プライマリネットワークインターフェイス (eth0) に関連付けられたセキュリティグループが変更されます。あらゆるネットワークインターフェイスに関連付けられているセキュリティグループも指定または変更できます。デフォルトでは、ネットワークインターフェイスを作成すると、別のセキュリティグループを指定しない限り、VPC のデフォルトのセキュリティグループに関連付けられます。ネットワークインターフェイスの詳細については、「Elastic network interface」を参照してください。
- セキュリティグループを作成する場合、名前と説明を指定する必要があります。以下のルールが適用されます。
  - 名前と説明の長さは最大 255 文字とすることができます。
  - 名前と説明に使用できる文字は、a ~ z、A ~ Z、0 ~ 9、スペース、._-:/()#,@[]+=&;{}!$* です。

  - 名前に末尾のスペースが含まれている場合は、名前の末尾のスペースを削除します。例えば、名前に「セキュリティグループのテスト 」と入力すると、「セキュリティグループのテスト」として保存されます。
  - セキュリティグループ名は、デフォルトのセキュリティグループを示すため、`sg-` で始めることはできません。
  - セキュリティグループ名は VPC 内で一意である必要があります。
- セキュリティグループは、セキュリティグループの作成時に指定した VPC でのみ使用できます。

VPC のデフォルトセキュリティグループ

VPC ではデフォルトのセキュリティグループが自動的に使用されます。インスタンスの起動時に別のセキュリティグループを指定しない場合、デフォルトのセキュリティグループがインスタンスに関連付けられます。

> **Note**
>
> Amazon EC2 コンソールでインスタンスを起動する場合、インスタンス起動ウィザードによって「launch-wizard-*xx*」セキュリティグループが自動的に定義され、デフォルトのセキュリティグループの代わりにインスタンスに関連付けることができます。

次の表では、デフォルトのセキュリティグループ用のデフォルトルールについて説明します。

Inbound			
送信元	プロトコル	ポート範囲	説明
セキュリティグループ ID (sg-xxxxxxxx)	すべて	すべて	同じセキュリティグループに割り当てられているネットワークインターフェイス (および関連付けられているインスタンス) からのインバウンドトラフィックを許可します。
Outbound			
送信先	プロトコル	ポート範囲	説明
0.0.0.0/0	すべて	すべて	すべての発信 IPv4 トラフィックを許可する
::/0	All	All	Allow all outbound IPv6 traffic. This rule is added by default if you create a VPC with an IPv6 CIDR block or if you associate an IPv6 CIDR block with your existing VPC.

デフォルトのセキュリティグループのルールは変更できます。

デフォルトのセキュリティグループを削除することはできません。デフォルトのセキュリティグループを削除しようとした場合、`Client.CannotDelete: the specified group: "sg-51530134" name: "default" cannot be deleted by a user` エラーが発生します。

セキュリティグループのアウトバウンドルールの設定を修正した場合、VPC と IPv6 ブロックを関連付けたときに IPv6 トラフィック用のルールは自動的に追加されません。

セキュリティグループのルール

セキュリティグループのルールは追加または削除できます (インバウンドまたはアウトバウンドアクセスの許可または取り消しとも呼ばれます)。ルールが適用されるのは、インバウンドトラフィック (受信) またはアウトバウンドトラフィック (送信) のいずれかです。特定の CIDR 範囲、または VPC あるいはピア VPC (VPC ピアリング接続が必要) の別のセキュリティグループにアクセス権を付与できます。

セキュリティグループルールは、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到達することを許可するインバウンドトラフィックを制御します。また、このルールによって、インスタンスから送信されるアウトバウンドトラフィックも制御されます。

セキュリティグループのルールの特徴を次に示します。

- デフォルトで、セキュリティグループはすべてのアウトバウンドトラフィックを許可します。
- セキュリティグループのルールは常にパーミッシブです。アクセスを拒否するルールを作成することはできません。
- セキュリティグループルールを使用すると、プロトコルとポート番号に基づいてトラフィックをフィルタリングできます。
- セキュリティグループはステートフルです。インスタンスからリクエストを送信すると、そのリクエストに対する応答トラフィックは、インバウンドルールにかかわらず、流入できます。つまり、許可されたインバウンドトラフィックに対する応答は、アウトバウンドルールにかかわらず通過することができます。
- ルールの追加と削除は随時行うことができます。変更は、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自動的に適用されます。

  一部のルール変更の影響は、トラフィックの追跡方法によって異なる場合があります。
- 複数のセキュリティグループをインスタンスに関連付けると、各セキュリティグループのルールが効率的に集約され、1 つのルールセットが作成されます。Amazon EC2 はこのルールセットを使用して、アクセスを許可するかを判断します。

  セキュリティグループは、1 つのインスタンスに複数割り当てることができます。そのため、1 つのインスタンスに数百のルールが適用される場合があります。結果として、インスタンスにアクセスするときに問題が発生する可能性があります。そのため、ルールは可能な限り要約することをお勧めします。

ルールごとに、以下の点について指定します。

- [Name (名前)]: セキュリティグループの名前 (`my-security-group` など)。

  名前の最大長は 255 文字です。使用できる文字は、a ~ z、A ~ Z、0 ~ 9、スペース、._-:/()#,@[]+=;{}!$* です。名前の末尾にスペースが含まれている場合は、名前を保存するときにスペースが切り捨てられます。例えば、名前に「セキュリティグループのテスト 」と入力すると、「セキュリティグループのテスト」として保存されます。
- プロトコル: 許可するプロトコル。最も一般的なプロトコルは、6 (TCP)、17 (UDP)、1 (ICMP) です。
- ポートの範囲: TCP、UDP、カスタムプロトコルの場合、許可するポートの範囲。1 つのポート番号 (`22` など)、または一定範囲のポート番号 (`7000-8000` など) を指定できます。
- ICMP タイプおよびコード: ICMP の場合、ICMP タイプおよびコードです。
- 送信元または送信先: トラフィックの送信元 (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)。これらのオプションの 1 つを指定します。
  - 単一の IPv4 アドレス。長さ `/32` のプレフィックスを使用する必要があります (例: `203.0.113.1/32`)。
  - 単一の IPv6 アドレス。長さ `/128` のプレフィックスを使用する必要があります (例: `2001:db8:1234:1a00::123/128`)。

  - CIDR ブロック表記での IPv4 アドレスの範囲 (例: `203.0.113.0/24`)。
  - CIDR ブロック表記での IPv6 アドレスの範囲 (例: `2001:db8:1234:1a00::/64`)。
  - プレフィックスリスト ID (例: `pl-1234abc1234abc123`)。詳細については、「プレフィックスリスト (p. 276)」を参照してください。
  - 別のセキュリティグループ。これにより、指定したセキュリティグループに関連付けられたインスタンスからこのセキュリティグループに関連付けられたインスタンスへのアクセスが許可されます。このオプションを選択しても、送信元のセキュリティグループからこのセキュリティグループにルールが追加されることはありません。以下のセキュリティグループの 1 つを指定できます:
    - 現在のセキュリティグループ。
    - 同じ VPC の異なるセキュリティグループ
    - VPC ピア接続のピア VPC の別のセキュリティグループ。
- (オプション) 説明: 後で分かりやすいように、このルールの説明を追加できます。説明の長さは最大 255 文字とすることができます。使用できる文字は、a ~ z、A ~ Z、0 ~ 9、スペース、._-:/()#,@[]+=;{}!$* です。

セキュリティグループルールを作成する際、AWS により、一意の ID がそのルールに割り当てられます。このルールの ID は、API または CLI を使用してルールを変更または削除する際に使用します。

ルールに送信元または送信先としてセキュリティグループを指定する場合、ルールはセキュリティグループに関連付けられているすべてのインスタンスに影響します。着信トラフィックは、ソースセキュリティグループに関連付けられたインスタンスのプライベート IP アドレスに基づいて許可されます (パブリック IP アドレスまたは Elastic IP アドレスは考慮されません)。セキュリティグループルールでピア VPC のセキュリティグループを参照していて、参照先のセキュリティグループまたは VPC ピア接続を削除すると、ルールは古いとマークされます。詳細については、Amazon VPC Peering ガイドの「古いセキュリティグループルールの操作」を参照してください。

セキュリティグループをルールの送信元として指定すると、指定したプロトコルとポートの送信元セキュリティグループに関連付けられたネットワークインターフェイスからのトラフィックが許可されます。着信トラフィックは、ソースセキュリティグループに関連付けられたネットワークインターフェイスのプライベート IP アドレスに基づいて許可されます (パブリック IP アドレスまたは Elastic IP アドレスは考慮されません)。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。

ファイアウォールを設定するための一部のシステムを使用すると、送信元ポートでフィルタを適用できます。セキュリティグループを使用すると、送信先ポートでのみフィルタを適用できます。

ルールを追加、更新、または削除すると、セキュリティグループに関連付けられたすべてのインスタンスにこの変更が自動的に適用されます。

追加するルールの種類は、多くの場合、セキュリティグループの目的によって異なります。次の表に、ウェブサーバーに関連付けられているセキュリティグループのルールの例を示します。ウェブサーバーはすべての IPv4 および IPv6 アドレスから HTTP および HTTPS トラフィックを受信し、SQL または MySQL トラフィックをデータベースサーバーに送信することができます。

Inbound			
送信元	プロトコル	ポート範囲	説明
0.0.0.0/0	TCP	80	任意の IPv4 アドレスからのインバウンド HTTP アクセスを許可します。

::/0	TCP	80	Allow inbound HTTP access from all IPv6 addresses
0.0.0.0/0	TCP	443	任意の IPv4 アドレスからのインバウンド HTTPS アクセスを許可する
::/0	TCP	443	Allow inbound HTTPS access from all IPv6 addresses
ネットワークのパブリック IPv4 アドレスの範囲	TCP	22	ネットワークの IPv4 アドレスから Linux インスタンスへのインバウンド SSH アクセス (インターネットゲートウェイ経由) を許可する
ネットワークのパブリック IPv4 アドレスの範囲	TCP	3389	ネットワークの IPv4 アドレスから Windows インスタンスへのインバウンド RDP アクセス (インターネットゲートウェイ経由) を許可する
Outbound			
送信先	プロトコル	ポート範囲	説明
Microsoft SQL Server データベースサーバーのセキュリティグループの ID	TCP	1433	アウトバウンド Microsoft SQL Server が指定されたセキュリティグループ内のインスタンスにアクセスするのを許可する
MySQL データベースサーバーのセキュリティグループの ID	TCP	3306	アウトバウンド MySQL が指定されたセキュリティグループ内のインスタンスにアクセスするのを許可する

データベースサーバーには、異なるルールセットが必要です。例えば、インバウンド HTTP および HTTPS トラフィックの代わりに、インバウンドの MySQL または Microsoft SQL Server アクセスを許可するルールを追加できます。ウェブサーバーとデータベースサーバーのセキュリティグループルールの例については、「Security (p. 58)」を参照してください。Amazon RDS DB インスタンスのセキュリティグループの詳細については、Amazon RDS ユーザーガイドの「セキュリティグループによるアクセスの制御」を参照してください。

特定の種類のアクセスに関するセキュリティグループのルールの例については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「セキュリティグループのルールのリファレンス」を参照してください。

古くなったセキュリティグループルール

VPC に別の VPC との VPC ピアリング接続がある場合、セキュリティグループルールで、ピア VPC の別のセキュリティグループを参照できます。これにより、参照されるセキュリティグループに関連付けられ

ているインスタンスと、参照するセキュリティグループに関連付けられているインスタンスが、相互に通信できるようになります。

ピア VPC の所有者が、参照されるセキュリティグループを削除するか、ユーザーまたはピア VPC の所有者が VPC ピアリング接続を削除した場合、セキュリティグループルールは `stale` とマークされます。古くなったセキュリティグループルールは他のセキュリティグループルールと同じ方法で削除できます。

詳細については、Amazon VPC ピア機能ガイドの「古いセキュリティグループの操作」を参照してください。

セキュリティグループの操作

以下のタスクでは、Amazon VPC コンソールを使用してセキュリティグループを操作する方法を示しています。

必要なアクセス許可

- セキュリティグループの管理 (p. 180)
- セキュリティグループルールの管理 (p. 181)

タスク

- デフォルトのセキュリティグループの変更 (p. 194)
- セキュリティグループの作成 (p. 194)
- セキュリティグループの表示 (p. 195)
- セキュリティグループのタグ付け (p. 195)
- セキュリティグループへのルールの追加 (p. 196)
- セキュリティグループルールの更新 (p. 197)
- セキュリティグループルールのタグ付け (p. 197)
- セキュリティグループルールの削除 (p. 198)
- インスタンスのセキュリティグループを変更する (p. 198)
- セキュリティグループを削除する (p. 198)

デフォルトのセキュリティグループの変更

VPC にはデフォルトのセキュリティグループ (p. 190)があります。このグループは削除できませんが、グループのルールは変更できます。その手順は、他のセキュリティグループを変更する手順と同じです。

セキュリティグループの作成

インスタンスのデフォルトのセキュリティグループを使用できますが、独自のグループを作成し、システムにおけるインスタンスの様々な役割を反映させたい場合があります。

デフォルトでは、新しいセキュリティグループには、すべてのトラフィックがインスタンスを出ることを許可するアウトバウンドルールのみが設定されています。任意のインバウンドトラフィックを許可するには、またはアウトバウンドトラフィックを制限するには、ルールを追加する必要があります。

セキュリティグループは、それが対象としている VPC 内でのみ使用が可能です。

セキュリティグループの作成とセキュリティグループルールの管理に必要な許可については、「セキュリティグループの管理 (p. 180)」と「セキュリティグループルールの管理 (p. 181)」を参照してください。

コンソールを使用してセキュリティグループを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Security Groups] を選択します。
3. [セキュリティグループの作成] を選択します。
4. セキュリティグループの名前と説明を入力します。セキュリティグループの作成後に名前と説明を変更することはできません。
5. [VPC] で、VPC を選択します。
6. セキュリティグループルールはここで追加することも、後で追加することもできます。詳細については、「セキュリティグループへのルールの追加 (p. 196)」を参照してください。
7. タグはここで追加することも、後で追加することもできます。タグを追加するには、[新しいタグを追加] をクリックし、タグのキーと値を入力します。
8. [セキュリティグループの作成] を選択します。

コマンドラインを使用してセキュリティグループを作成するには

- create-security-group (AWS CLI)
- New-EC2SecurityGroup (AWS Tools for Windows PowerShell)

セキュリティグループの表示

次のように、セキュリティグループに関する情報を表示できます。

セキュリティグループの表示に必要な許可については、「セキュリティグループの管理 (p. 180)」を参照してください。

コンソールを使用してセキュリティグループを表示するには

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] を選択します。
3. セキュリティグループが一覧表示されます。インバウンドルールやアウトバウンドルールなど、特定のセキュリティグループの詳細を表示するには、セキュリティグループを選択します。

コマンドラインを使用してセキュリティグループを表示するには

- describe-security-groups および describe-security-group-rules (AWS CLI)
- Get-EC2SecurityGroup および Get-EC2SecurityGroupRules (AWS Tools for Windows PowerShell)

リージョン間ですべてのセキュリティグループを表示するには

次の URL で Amazon EC2 グローバルビューコンソールを開きます。https://console.aws.amazon.com/ec2globalview/home

Amazon EC2 グローバルビューの使用方法については、Linux インスタンス用ユーザーガイドの 「リソースの一覧表示およびフィルタリング」を参照してください。

セキュリティグループのタグ付け

目的、所有者、環境などに応じて、タグを整理して識別しやすくするために、リソースにタグを追加します。セキュリティグループにはタグを追加できます。タグキーは、各セキュリティグループで一意である

必要があります。既にルールに関連付けられているキーを持つタグを追加すると、そのタグの値が更新されます。

コンソールを使用してセキュリティグループにタグを付けるには

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] を選択します。
3. セキュリティグループのチェックボックスを選択します。
4. [Actions] (アクション)、[Manage tags] (タグの管理) を選択します。
5. [Manage tags] (タグの管理) ページには、セキュリティグループに割り当てられているタグが表示されます。タグを追加するには、[タグの追加] を選択し、タグのキーと値を入力します。タグを削除するには、削除するタグの横にある [Remove] を選択します。
6. [Save changes] を選択します。

コマンドラインを使用してセキュリティグループのタグ付けを行うには

- create-tags (AWS CLI)
- New-EC2Tag (AWS Tools for Windows PowerShell)

セキュリティグループへのルールの追加

ルールをセキュリティグループに追加すると、セキュリティグループに関連付けられているすべてのインスタンスに新しいルールが自動的に適用されます。

VPC ピアリング接続がある場合は、セキュリティグループルールで、送信元または送信先としてピア VPC からセキュリティグループを参照できます。詳細については、Amazon VPC ピア機能ガイドの「セキュリティグループの更新によるピア VPC セキュリティグループの参照」を参照してください。

セキュリティグループルールの管理に必要な許可については、「セキュリティグループルールの管理 (p. 181)」を参照してください。

コンソールを使用してルールを追加するには

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、[Security Groups] を選択します。
3. セキュリティグループを選択します。
4. [アクション]、[Edit inbound rules (インバウンドルールを編集)] を選択するか、[アクション]、[Edit outbound rules (アウトバウンドルールを編集)] を選択します。
5. 各ルールで、[Add rule] (ルールの追加) を選択し、次の操作を行います。

   a. [タイプ] で、許可するプロトコルのタイプを選択します。

      - TCP または UDP の場合は、許可するポート範囲を入力する必要があります。
      - カスタムの ICMP の場合は、[プロトコル] から ICMP タイプ名を選択し、該当するものがある場合は [ポート範囲] からコード名を選択します。
      - その他のタイプの場合は、プロトコルとポート範囲は自動的に設定されます。

   b. [Source] (送信元) (インバウンドルール) または [Destination] (送信先) (アウトバウンドルール) で、トラフィックを許可するために次のいずれかの操作を行います。

      - [Custom] (カスタム) をクリックし、IP アドレス (CIDR 表記)、CIDR ブロック、別のセキュリティグループ、プレフィックスリストのいずれかを入力します。
      - [Anywhere] (任意の場所) を選択して、任意の IP アドレスからインスタンスへのトラフィックの到達 (インバウンドルール)、またはインスタンスからすべての IP アドレスへのトラフィッ

クの到達を許可します (アウトバウンドルール)。このオプションでは、IPv4 の CIDR ブロック 0.0.0.0/0 が自動的に追加されます。

セキュリティグループが IPv6 が有効な VPC にある場合、このオプションでは ::/0 IPv6 トラフィックのためにルールが自動的に追加されます。

インバウンドルールの場合、このオプションはテスト環境で短時間なら許容できますが、実稼働環境で行うのは安全ではありません。実稼働環境では、特定の IP アドレスまたは特定のアドレス範囲にのみ、インスタンスへのアクセスを許可します。

- ローカルコンピュータのパブリック IPv4 アドレスとの間の受信トラフィック (インバウンドルール) または送信トラフィック (アウトバウンドルール)

c. (オプション) [Description] (説明) では、ルールの簡単な説明を指定できます。

6. [Save Rules (ルールの保存)] を選択します。

コマンドラインを使用してセキュリティグループにルールを追加するには

- authorize-security-group-ingress および authorize-security-group-egress (AWS CLI)
- Grant-EC2SecurityGroupIngress および Grant-EC2SecurityGroupEgress (AWS Tools for Windows PowerShell)

セキュリティグループルールの更新

ルールを更新すると、更新されたルールは、セキュリティグループに関連付けられているすべてのインスタンスに自動的に適用されます。

セキュリティグループルールの管理に必要な許可については、「セキュリティグループルールの管理 (p. 181)」を参照してください。

コンソールを使用して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、[Security Groups] を選択します。
3. セキュリティグループを選択します。
4. [アクション]、[Edit inbound rules (インバウンドルールを編集)] を選択するか、[アクション]、[Edit outbound rules (アウトバウンドルールを編集)] を選択します。
5. 必要に応じてルールを更新します。
6. [Save Rules (ルールの保存)] を選択します。

コマンドラインを使用してセキュリティグループルールの説明を更新するには

- modify-security-group-rules、update-security-group-rule-descriptions-ingress、および update-security-group-rule-descriptions-egress (AWS CLI)
- Update-EC2SecurityGroupRuleIngressDescription および Update-EC2SecurityGroupRuleEgressDescription (AWS Tools for Windows PowerShell)

セキュリティグループルールのタグ付け

目的、所有者、環境などに応じて、タグを整理して識別しやすくするために、リソースにタグを追加します。セキュリティグループルールにタグを追加できます。タグキーは、各ターゲットグループルールで一意である必要があります。既にターゲットグループルールに関連付けられているキーを持つタグを追加すると、そのタグの値が更新されます。

コンソールを使用してルールにタグを付けるには

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] を選択します。
3. セキュリティグループを選択します。
4. [インバウンドルール] または [アウトバウンドルール] タブで、対象となるルールのチェックボックスを選択してから、[タグを管理] をクリックします。
5. [タグの管理] ページには、ルールに割り当てられているすべてのタグが表示されます。タグを追加するには、[タグの追加] を選択し、タグのキーと値を入力します。タグを削除するには、削除するタグの横にある [Remove] を選択します。
6. [Save changes] を選択します。

コマンドラインを使用してルールにタグを付けるには

- create-tags (AWS CLI)
- New-EC2Tag (AWS Tools for Windows PowerShell)

セキュリティグループルールの削除

セキュリティグループからルールを削除すると、その変更内容が自動的にセキュリティグループに関連付けられているインスタンスに適用されます。

コンソールを使用してセキュリティグループルールを削除するには

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、[Security Groups] を選択します。
3. セキュリティグループを選択します。
4. [Actions] (アクション) を選択してから、[Edit inbound rules] (インバウンドのルールの編集) を選択してインバウンドルールを削除するか、[Edit outbound rules] (アウトバウンドのルールの編集) を選択してアウトバウンドルールを削除します。
5. 削除するルールの横にある [Delete] (削除) ボタンを選択します。
6. [Save Rules (ルールの保存)] を選択します。

コマンドラインを使用してセキュリティグループルールを削除するには

- revoke-security-group-ingress および revoke-security-group-egress (AWS CLI)
- Revoke-EC2SecurityGroupIngress および Revoke-EC2SecurityGroupEgress (AWS Tools for Windows PowerShell)

インスタンスのセキュリティグループを変更する

VPC でインスタンスを起動したら、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変更はインスタンスが `running` または `stopped` の状態のときに行えます。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「インスタンスのセキュリティグループの変更」を参照してください。

セキュリティグループを削除する

セキュリティグループは、インスタンスに割り当てられていない場合にのみ削除できます (インスタンスが実行されているかどうかは関係ありません)。実行中のインスタンスまたは停止した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を変更できます。詳細については、インスタンスのセキュリティグ

ループを変更する (p. 198) を参照してください。デフォルトのセキュリティグループを削除することはできません。

コンソールを使用している場合は、複数のセキュリティグループを一度に削除できます。コマンドラインまたは API を使用している場合、一度に削除できるのは 1 つのセキュリティグループのみです。

コンソールを使用してセキュリティグループを削除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Security Groups] を選択します。
3. 1 つ以上のセキュリティグループを選択して、[Actions] (アクション)、[Delete Security Group] (セキュリティグループセキュリティグループの削除) を選択します。
4. 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セキュリティグループを削除するには

- delete-security-group (AWS CLI)
- Remove-EC2SecurityGroup (AWS Tools for Windows PowerShell)

の使用による VPC セキュリティグループの一元管理AWS Firewall Manager

AWS Firewall Manager を使用すると、複数のアカウントおよび複数のリソースを対象にして、VPC セキュリティグループの管理およびメンテナンスタスクを簡略化できます。Firewall Manager を使用すると、1 つの中央管理者アカウントで組織のセキュリティグループを設定および監査できます。Firewall Manager により、ルールと保護が既存のアカウントとリソースに (追加する新しいリソースにも) 自動的に適用されます。Firewall Manager は、組織全体を保護する場合や、中央管理者アカウントで保護する新しいリソースを頻繁に追加する場合に特に便利です。

Firewall Manager を使用すると、次の方法でセキュリティグループを一元管理できます。

- 組織全体で共通のベースラインセキュリティグループを設定する: 共通のセキュリティグループポリシーを使用して、組織全体のアカウントおよびリソースに対するセキュリティグループの関連付けを一元的に制御できます。組織内でポリシーを適用する場所と方法を指定します。
- 組織内の既存のセキュリティグループを監査する: 監査セキュリティグループポリシーを使用して、組織のセキュリティグループで使用中の既存のルールを確認できます。ポリシーの範囲を設定して、すべてのアカウント、特定のアカウント、または組織内でタグ付けされたリソースを監査できます。Firewall Manager により、新しいアカウントとリソースの自動検出と監査が行われます。監査ルールを作成することにより、組織内で許可または禁止するセキュリティグループルールに関するガードレールを設定し、未使用または冗長なセキュリティグループをチェックできます。
- 非準拠のリソースに関するレポートを取得して修復する: ベースラインおよび監査ポリシーについて、非準拠のリソースに関するレポートとアラートを取得できます。自動修復ワークフローを設定しておき、Firewall Manager によって検出された非準拠のリソースを修復することもできます。

Firewall Manager を使用してセキュリティグループを管理する方法の詳細については、AWS WAF 開発者ガイドの以下のトピックを参照してください。

- AWS Firewall Manager の前提条件
- https://docs.aws.amazon.com/waf/latest/developerguide/getting-started-fms-security-group.html Amazon VPC セキュリティグループポリシーの使用を開始するAWS Firewall Manager
- でのセキュリティグループポリシーの仕組みAWS Firewall Manager
- セキュリティグループポリシーのユースケース

ネットワーク ACL

ネットワークアクセスコントロールリスト (ACL) は、1 つ以上のサブネットのインバウンドトラフィックとアウトバウンドトラフィックを制御するファイアウォールとして動作する、VPC 用のセキュリティのオプションレイヤーです。セキュリティの追加レイヤーを VPC に追加するには、セキュリティグループと同様のルールを指定したネットワーク ACL をセットアップできます。セキュリティグループとネットワーク ACL の違いの詳細については、「セキュリティグループとネットワーク ACL を比較する (p. 166)」を参照してください。

目次

ネットワーク ACL の基本

ネットワーク ACL について知っておく必要がある基本的な情報を以下に示します。

- VPC には、変更可能なデフォルトのネットワーク ACL が自動的に設定されます。デフォルトでは、すべてのインバウンドおよびアウトバウンドの IPv4 トラフィックと、IPv6 トラフィック (該当する場合) が許可されます。
- カスタムネットワーク ACL を作成し、サブネットと関連付けることができます。デフォルトでは、各カスタムネットワーク ACL は、ルールを追加するまですべてのインバウンドトラフィックとアウトバウンドトラフィックを拒否します。
- VPC 内の各サブネットにネットワーク ACL を関連付ける必要があります。ネットワーク ACL に明示的にサブネットを関連付けない場合、サブネットはデフォルトのネットワーク ACL に自動的に関連付けられます。
- ネットワーク ACL を複数のサブネットに関連付けることができます。ただし、サブネットは一度に 1 つのネットワーク ACL にのみ関連付けることができます。サブネットとネットワーク ACL を関連付けると、以前の関連付けは削除されます。
- ネットワーク ACL には、ルールの番号付きリストが含まれます。低い番号から順にルールを評価し、ネットワーク ACL に関連付けられたサブネットのインバウンドトラフィックまたはアウトバウンドトラフィックが許可されるかどうかを指定します。ルールに使用できる最も高い番号は 32766 です。まずは増分 (たとえば 10 または 100 の増分) でルールを作成することをお勧めします。こうすると、後で必要になったときに新しいルールを挿入できます。
- ネットワーク ACL には個別のインバウンドルールとアウトバウンドルールがあり、各ルールでトラフィックを許可または拒否できます。
- ネットワーク ACL はステートレスです。許可されているインバウンドトラフィックに対する応答は、アウトバウンドトラフィックのルールに従います (その逆の場合も同様です) 。

VPC あたりのネットワーク ACL の数とネットワーク ACL あたりのルールの数には、クォータ (制限) があります。詳細については、「」を参照してくださいAmazon VPC クォータ (p. 362)

ネットワーク ACL ルール

デフォルトのネットワーク ACL に対してルールの追加または削除を行うことができます。また、VPC に合わせて追加のネットワーク ACL を作成することができます。ネットワーク ACL に対してルールの追加または削除を行うと、変更内容は、その ACL に関連付けられているサブネットに自動的に適用されます。

次に、ネットワーク ACL ルールの一部を示します。

- ルール番号。ルールは、最も低い番号のルールから評価されます。ルールがトラフィックに一致すると、それと相反するより高い数値のルールの有無にかかわらず、すぐに適用されます。
- タイプ。トラフィックのタイプ (SSH など) 。また、すべてのトラフィックまたはカスタム範囲を指定することもできます。
- プロトコル。標準のプロトコル番号を持つ任意のプロトコルを指定できます。詳細については、「プロトコル番号」を参照してください。プロトコルとして ICMP を指定する場合、任意またはすべての ICMP タイプとコードを指定できます。
- ポート範囲。トラフィックのリスニングポートまたはポート範囲。たとえば、HTTP トラフィックの場合は 80 です。
- ソース: [インバウンドルールのみ] トラフィックの送信元 (CIDR 範囲)。
- 送信先 [アウトバウンドルールのみ] トラフィックの送信先 (CIDR 範囲)。
- 許可/拒否。指定されたトラフィックを許可するか拒否するかを指定します。

コマンドラインツールまたは Amazon EC2 API を使用してルールを追加すると、CIDR 範囲は自動的に正規形式に変更されます。たとえば、CIDR 範囲に `100.68.0.18/18` を指定すると、`100.68.0.0/18` の CIDR 範囲を持つルールが作成されます。

デフォルトのネットワーク ACL

デフォルトのネットワーク ACL は、すべてのトラフィックが、関連するサブネットを出入りすることを許可するように設定されます。各ネットワーク ACL にも、ルール番号がアスタリスクのルールが含まれます。このルールによって、パケットが他のいずれの番号のルールとも一致しない場合は、確実に拒否されます。このルールを変更または削除することはできません。

IPv4 のみをサポートする VPC のデフォルトネットワーク ACL の例を以下に示します。

インバウンド					
ルール番号	タイプ	プロトコル	ポート範囲	送信元	許可/拒否
100	All IPv4 traffic	すべて	すべて	0.0.0.0/0	許可
*	All IPv4 traffic	すべて	すべて	0.0.0.0/0	拒否
アウトバウンド					
ルール番号	タイプ	プロトコル	ポート範囲	送信先	許可/拒否
100	All IPv4 traffic	すべて	すべて	0.0.0.0/0	許可
*	All IPv4 traffic	すべて	すべて	0.0.0.0/0	拒否

IPv6 CIDR ブロックを持つ VPC を作成するか、IPv6 CIDR ブロックを既存の VPC と関連付ける場合は、すべての IPv6 トラフィックがサブネット間を流れるようにするルールが自動的に追加されます。また、ルール番号がアスタリスクのルールが追加されます。このルールにより、パケットが他のいずれのルール

とも一致しない場合は、確実に拒否されます。このルールを変更または削除することはできません。IPv4 または IPv6 をサポートする VPC のデフォルトネットワーク ACL の例を以下に示します。

Note

デフォルトのネットワーク ACL のインバウンドルールを変更した場合は、IPv6 ブロックを VPC と関連付けても、インバウンド IPv6 トラフィックを許可するルールが自動的に追加されることはありません。同様に、アウトバウンドルールを変更した場合、アウトバウンド IPv6 を許可するルールが自動的に追加されることはありません。

インバウンド					
ルール番号	タイプ	プロトコル	ポート範囲	送信元	許可/拒否
100	All IPv4 traffic	すべて	すべて	0.0.0.0/0	許可
101	すべての IPv6 トラフィック	すべて	すべて	::/0	許可
*	All traffic	すべて	すべて	0.0.0.0/0	拒否
*	すべての IPv6 トラフィック	すべて	すべて	::/0	拒否
アウトバウンド					
ルール番号	タイプ	プロトコル	ポート範囲	送信先	許可/拒否
100	All traffic	すべて	すべて	0.0.0.0/0	許可
101	すべての IPv6 トラフィック	すべて	すべて	::/0	許可
*	All traffic	すべて	すべて	0.0.0.0/0	拒否
*	すべての IPv6 トラフィック	すべて	すべて	::/0	拒否

カスタムネットワーク ACL

IPv4 のみをサポートする VPC のカスタムネットワーク ACL の例を以下のテーブルに示します。この ACL には、HTTP および HTTPS のインバウンドトラフィック (インバウンドルール 100 および 110) を許可するルールが含まれます。そのインバウンドトラフィックに対する応答を可能にする、対応するアウトバウンドルールがあります (一時ポート 32768 ~ 65535 を対象とするアウトバウンドルール 140)。適切な一時ポートの範囲を選択する方法の詳細については、「一時ポート (p. 212)」を参照してください。

ネットワーク ACL には、SSH および RDP からサブネットに対するトラフィックを許可するインバウンドルールも含まれます。アウトバウンドルール 120 は、サブネットに送信される応答を可能にします。

ネットワーク ACL には、サブネットからの HTTP および HTTPS のアウトバウンドトラフィックを許可するアウトバウンドルール (100 および 110) があります。そのアウトバウンドトラフィックに対する応答を可能にする、対応するインバウンドルールがあります (一時ポート 32768 ~ 65535 を対象とするインバウンドルール 140)。

Note

各ネットワーク ACL には、ルール番号がアスタリスクのデフォルトルールが含まれます。このルールによって、パケットが他のいずれのルールとも一致しない場合は、確実に拒否されます。このルールを変更または削除することはできません。

インバウンド						
ルール番号	タイプ	プロトコル	ポート範囲	送信元	許可/拒否	コメント
100	HTTP	TCP	80	0.0.0.0/0	許可	任意の IPv4 アドレスからのインバウンド HTTP トラフィックを許可します。
110	HTTPS	TCP	443	0.0.0.0/0	許可	任意の IPv4 アドレスからのインバウンド HTTPS トラフィックを許可します。
120	SSH	TCP	22	192.0.2.0/24	許可	(インターネットゲートウェイを介した) ホームネットワークのパブリック IPv4 アドレスの範囲からのインバウンド SSH トラフィックを許可します。
130	RDP	TCP	3389	192.0.2.0/24	許可	(インターネットゲートウェイを介した) ホームネットワークのパブリック IPv4 アドレスの範囲からウェブサーバーに対するインバウンド RDP トラフィックを許可します。
140	Custom TCP	TCP	32768-65535	0.0.0.0/0	許可	(送信元がサブネットであるリクエストに対

						する）インターネットからのインバウンドリターン IPv4 トラフィックを許可します。 この範囲は一例に過ぎません。適切な一時ポートの範囲を選択する方法の詳細については、「一時ポート (p. 212)」を参照してください。
*	All traffic	すべて	すべて	0.0.0.0/0	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv4 トラフィックを拒否します (変更不可)。
アウトバウンド						
ルール番号	タイプ	プロトコル	ポート範囲	送信先	許可/拒否	コメント
100	HTTP	TCP	80	0.0.0.0/0	許可	サブネットからインターネットへのアウトバウンド IPv4 HTTP トラフィックを許可します。
110	HTTPS	TCP	443	0.0.0.0/0	許可	サブネットからインターネットへのアウトバウンド IPv4 HTTPS トラフィックを許可します。

120	SSH	TCP	22	192.0.2.0/24	許可	(インターネットゲートウェイを介した) ホームネットワークのパブリック IPv4 アドレスの範囲からのアウトバウンド SSH トラフィックを許可します。
140	Custom TCP	TCP	32768-65535	0.0.0.0/0	許可	インターネット上のクライアントに対するアウトバウンド IPv4 応答を許可します (例: サブネット内のウェブサーバーを訪問するユーザーに対するウェブページの提供) 。 この範囲は一例に過ぎません。適切な一時ポートの範囲を選択する方法の詳細については、「一時ポート (p. 212)」を参照してください。
*	All traffic	すべて	すべて	0.0.0.0/0	拒否	前のルールでまだ処理されていないすべてのアウトバウンド IPv4 トラフィックを拒否します (変更不可)。

パケットがサブネットに送信されると、サブネットが関連付けられている ACL のインバウンドルールと照合して評価されます (ルールリストの一番上から順に一番下まで評価されます) 。パケットが HTTPS ポート (443) あての場合の評価方法は次のとおりです。パケットは最初に評価されるルール (ルール 100) と一致しません。また、2 番目のルール (110) とは一致します。このルールでは、サブネットに送信されるパケットを許可します。パケットの宛先がポート 139 (NetBIOS) である場合は、いずれのルールとも一致せず、最終的に * ルールによってパケットが拒否されます。

正当に幅広い範囲のポートを開く必要があり、その範囲内の特定のポートは拒否する場合は、拒否ルールを追加します。このとき、テーブル内で、幅広い範囲のポートトラフィックを許可するルールよりも先に拒否ルールを配置します。

ユースケースに応じて、許可 ルールを追加します。たとえば、DNS 解決のためにポート 53 でアウトバウンド TCP および UDP アクセスを許可するルールを追加できます。追加するすべてのルールにおいて、応答トラフィックを許可する該当のインバウンドルールまたはアウトバウンドルールがあることを確認します。

IPv6 CIDR ブロックと関連付けた VPC のカスタムネットワーク ACL の同一例を以下のテーブルに示します。このネットワーク ACL には、すべての IPv6 HTTP および HTTPS トラフィックのルールが含まれます。この場合、IPv4 トラフィックの既存のルールの間に新しいルールが挿入されました。IPv4 ルールの後に、ルールを大きい数のルールとして追加することもできます。IPv4 トラフィックと IPv6 トラフィックは異なります。したがって、IPv4 トラフィックのルールはいずれも IPv6 トラフィックに適用することはできません。

インバウンド						
ルール番号	タイプ	プロトコル	ポート範囲	送信元	許可/拒否	コメント
100	HTTP	TCP	80	0.0.0.0/0	許可	任意の IPv4 アドレスからのインバウンド HTTP トラフィックを許可します。
105	HTTP	TCP	80	::/0	許可	任意の IPv6 アドレスからのインバウンド HTTP トラフィックを許可します。
110	HTTPS	TCP	443	0.0.0.0/0	許可	任意の IPv4 アドレスからのインバウンド HTTPS トラフィックを許可します。
115	HTTPS	TCP	443	::/0	許可	任意の IPv6 アドレスからのインバウンド HTTPS トラフィック

						を許可します。
120	SSH	TCP	22	192.0.2.0/24	許可	（インターネットゲートウェイを介した）ホームネットワークのパブリック IPv4 アドレスの範囲からのインバウンド SSH トラフィックを許可します。
130	RDP	TCP	3389	192.0.2.0/24	許可	（インターネットゲートウェイを介した）ホームネットワークのパブリック IPv4 アドレスの範囲からウェブサーバーに対するインバウンド RDP トラフィックを許可します。

140	Custom TCP	TCP	32768-65535	0.0.0.0/0	許可	（送信元がサブネットであるリクエストに対する）インターネットからのインバウンドリターン IPv4 トラフィックを許可します。 この範囲は一例に過ぎません。適切な一時ポートの範囲を選択する方法の詳細については、「一時ポート (p. 212)」を参照してください。
145	Custom TCP	TCP	32768-65535	::/0	ALLOW	（送信元がサブネットであるリクエストに対する）インターネットからのインバウンドリターン IPv6 トラフィックを許可します。 この範囲は一例に過ぎません。適切な一時ポートの範囲を選択する方法の詳細については、「一時ポート (p. 212)」を参照してください。

*	All traffic	すべて	すべて	0.0.0.0/0	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv4 トラフィックを拒否します (変更不可)。
*	すべてのトラフィック	すべて	すべて	::/0	拒否	前のルールでまだ処理されていないすべてのインバウンド IPv6 トラフィックを拒否します (変更不可)。

アウトバウンド

ルール番号	タイプ	プロトコル	ポート範囲	送信先	許可/拒否	コメント
100	HTTP	TCP	80	0.0.0.0/0	許可	サブネットからインターネットへのアウトバウンド IPv4 HTTP トラフィックを許可します。
105	HTTP	TCP	80	::/0	許可	サブネットからインターネットへのアウトバウンド IPv6 HTTP トラフィックを許可します。
110	HTTPS	TCP	443	0.0.0.0/0	許可	サブネットからインターネットへのアウトバウンド IPv4 HTTPS トラフィックを許可します。

115	HTTPS	TCP	443	::/0	許可	サブネットからインターネットへのアウトバウンド IPv6 HTTPS トラフィックを許可します。
140	Custom TCP	TCP	32768-65535	0.0.0.0/0	許可	インターネット上のクライアントに対するアウトバウンド IPv4 応答を許可します (例: サブネット内のウェブサーバーを訪問するユーザーに対するウェブページの提供) 。 この範囲は一例に過ぎません。適切な一時ポートの範囲を選択する方法の詳細については、「一時ポート (p. 212)」を参照してください。

145	カスタム TCP	TCP	32768-65535	::/0	許可	インターネット上のクライアントに対するアウトバウンド IPv6 応答を許可します (例: サブネット内のウェブサーバーを訪問するユーザーに対するウェブページの提供) 。 この範囲は一例に過ぎません。適切な一時ポートの範囲を選択する方法の詳細については、「一時ポート (p. 212)」を参照してください。
*	All traffic	すべて	すべて	0.0.0.0/0	拒否	前のルールでまだ処理されていないすべてのアウトバウンド IPv4 トラフィックを拒否します (変更不可)。
*	すべてのトラフィック	すべて	すべて	::/0	拒否	前のルールでまだ処理されていないすべてのアウトバウンド IPv6 トラフィックを拒否します (変更不可)。

その他の例については、「VPC ウィザードシナリオの推奨ルール (p. 219)」を参照してください。

カスタムネットワーク ACL およびその他の AWS のサービス

カスタムネットワーク ACL を作成する場合は、他の AWS のサービスを使用して作成したリソースにどのように影響するか注意してください。

Elastic Load Balancing では、バックエンドインスタンスのサブネットに、ソースが `0.0.0.0/0` であるかサブネットの CIDR のいずれかであるすべてのトラフィックに追加した拒否ルールを適用するネットワーク ACL がある場合、ロードバランサーはインスタンスのヘルスチェックを実行できません。ロードバランサーとバックエンドインスタンスに推奨されるネットワーク ACL ルールに関する詳細については、Classic Load Balancer のユーザーガイドの「VPC のロードバランサーのネットワーク ACL」を参照してください。

一時ポート

前のセクションでは、ネットワーク ACL の例に 32768 ~ 65535 という一時ポートの範囲を使用しています。ただし、使用または通信しているクライアントの種類によっては、ネットワーク ACL に別の範囲を使用してもかまいません。

リクエストを開始するクライアントは、一時ポートの範囲を選択します。範囲は、クライアントのオペレーティングシステムによって変わります。

- 多くの Linux カーネル (Amazon Linux カーネルを含む) は、ポート 32768 ~ 61000 を使用します。
- Elastic Load Balancing からのリクエストは、ポート 1024-65535 を使用します。
- Windows Server 2003 を介する Windows オペレーティングシステムは、ポート 1025 ~ 5000 を使用します。
- Windows Server 2008 以降のバージョンでは、ポート 49152 ~ 65535 を使用します。
- NAT ゲートウェイはポート 1024 ~ 65535 を使用します。
- AWS Lambda 関数は、ポート 1024-65535 を使用します。

たとえば、インターネット上の Windows 10 クライアントから、お客様の VPC のウェブサーバーにリクエストが送信される場合、ネットワーク ACL には、ポート 49152 ~ 65535 宛てのトラフィックを可能にするアウトバウンドルールを用意する必要があります。

VPC 内のインスタンスが、リクエストを開始するクライアントの場合、ネットワーク ACL には、インスタンス (Amazon Linux、Windows Server 2008 など) の種類に固有の一時ポートあてのトラフィックを可能にするインバウンドルールを用意する必要があります。

実際に、VPC 内のパブリックに面したインスタンスに対して、トラフィックを開始することができる多様なクライアントを対象にするには、一時ポート 1024 ~ 65535 を開くことができます。ただし、その範囲内で悪意のあるポートのトラフィックを拒否するルールを ACL を追加することもできます。このとき、テーブル内で、幅広い範囲の一時ポートを開く許可ルールよりも先に拒否ルールを配置します。

パス MTU 検出

2 つのデバイス間のパス MTU を判断するために、パス MTU 検出が使用されます。パス MTU は、送信側ホストと受信側ホスト間のパスでサポートされている最大のパケットサイズです。

IPv4 の場合、ホストがパスに沿って送信するパケットが、受信側ホストの MTU、あるいはデバイスの MTU よりも大きな場合、受信側ホストまたはデバイスはそのパケットをドロップし、次のような ICMP メッセージ `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set` (タイプ 3、コード 4) を返します。このメッセージは送信側ホストに対し、ペイロードを複数の小さなパケットに分割し再送信することを指示します。

IPv6 プロトコルは、ネットワークのフラグメンテーションをサポートしていません。ホストがパスに沿って送信するパケットが、受信側ホストの MTU、あるいはデバイスの MTU よりも大きな場合、受信側ホストまたはデバイスはそのパケットをドロップし、次のような ICMP メッセージ `ICMPv6 Packet Too Big (PTB)` (タイプ 2) を返します。このメッセージは送信側ホストに対し、ペイロードを複数の小さなパケットに分割し再送信することを指示します。

サブネット内のホスト間の最大送信単位 (MTU) が異なる場合、またはインスタンスがインターネット経由でピアと通信する場合、インバウンドとアウトバウンドの両方に、以下のネットワーク ACL ルールを追加する必要があります。これにより、パス MTU 検出が正しく機能し、パケット損失を防ぐことができます。タイプに [Custom ICMP Rule] を選択し、ポート範囲 (タイプ 3、コード 4) に [送信先に到達できません]、[fragmentation required, and DF flag set (フラグメンテーションが必要、および DF フラグを設定)] を選択します。トレースルートを使用する場合は、次のルールも追加します。[カスタム ICMP ルール] (タイプ)、[時間超過]、[TTL 伝送期限切れ] (ポート範囲: タイプ 11、コード 0) を選択します。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「EC2 インスタンスのネットワーク最大送信単位 (MTU)」を参照してください。

ネットワーク ACL の動作

以下のタスクでは、Amazon VPC コンソールを使用してネットワーク ACL を操作する方法を示しています。

タスク

- ネットワーク ACL の関連付けの確認 (p. 213)
- ネットワーク ACL の作成 (p. 214)
- ルールの追加と削除 (p. 214)
- サブネットとネットワーク ACL の関連付け (p. 215)
- ネットワーク ACL とサブネットの関連付けの解除 (p. 215)
- サブネットのネットワーク ACL の変更 (p. 215)
- ネットワーク ACL を削除する (p. 216)
- API とコマンドの概要 (p. 216)

ネットワーク ACL の関連付けの確認

Amazon VPC コンソールを使用して、サブネットに関連付けられているネットワーク ACL を確認することができます。ネットワーク ACL を複数のサブネットに関連付けて、ネットワーク ACL に関連付けられているサブネットを確認することもできます。

サブネットと関連付けられているネットワーク ACL を確認するには

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 [Subnets] を選択し、サブネットを選択します。

   サブネットに関連付けられているネットワーク ACL は、ネットワーク ACL のルールと共に [Network ACL] タブに表示されます。

ネットワーク ACL に関連付けられたサブネットを決定するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインの [Network ACLs] を選択します。[Associated With] 列には、各ネットワーク ACL に関連付けられているサブネットの数が表示されます。
3. ネットワーク ACL を選択します。
4. 詳細ペインで [Subnet Associations (サブネットの関連付け)] を選択して、ネットワーク ACL に関連付けられているサブネットを表示します。

ネットワーク ACL の作成

VPC のカスタムネットワーク ACL を作成できます。デフォルトでは、作成するネットワーク ACL により、ルールを追加するまですべてのインバウンドおよびアウトバウンドトラフィックがブロックされ、明示的に関連付けるまではサブネットと関連付けられません。

ネットワーク ACL を作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインの [Network ACLs] を選択します。
3. [Create Network ACL] を選択します。
4. [Create Network ACL (ネットワーク ACL の作成)] ダイアログボックスで、オプションでネットワーク ACL に名前を付けて、[VPC] リストから VPC の ID を選択します。続いて、[Yes, Create (はい、作成します)] を選択します。

ルールの追加と削除

ACL のルールの追加または削除を行うと、その ACL に関連付けられたすべてのサブネットに変更が反映されます。サブネット内のインスタンスを終了して再起動する必要はありません。変更は短期間で有効になります。

> Important
>
> ルールを同時に追加したり削除したりする場合は、十分に注意してください。ネットワーク ACL ルールは、VPC に出入りできるネットワークトラフィックのタイプを定義します。インバウンドルールまたはアウトバウンドルールを削除し、Amazon VPC クォータ (p. 362) で許可されている数より多くのエントリを追加した場合、削除対象として選択されたエントリは削除されますが、新しいエントリは追加されません。これにより、予期しない接続の問題が発生し、VPC とのアクセスが意図せず妨げられる可能性があります。

Amazon EC2 API またはコマンドラインツールを使用している場合は、ルールを変更できません。ルールの追加と削除のみを行うことができます。Amazon VPC コンソールを使用している場合は、既存のルールのエントリを変更できます。コンソールは既存のルールを削除し、新しいルールを追加します。ACL のルールの順序を変更する必要がある場合は、新しいルール番号を指定した新しいルールを追加してから、元のルールを削除します。

ルールをネットワーク ACL に追加するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインの [Network ACLs] を選択します。
3. 詳細ペインで、追加する必要があるルールの種類に応じて、[Inbound Rules] タブまたは [Outbound Rules] タブを選択し、[Edit] を選択します。
4. [Rule #] にルール番号 (100 など) を入力します。ネットワーク ACL にすでに使用されているルール番号は使用できません。ルールは、最も低い番号から順に処理されます。

   ルール番号は、連続番号 (101、102、103 など) を使用せずに、間を空けておくことをお勧めします (100、200、300 など)。こうすることで、既存のルールに番号を振り直さなくても、新しいルールを簡単に追加できるようになります。
5. [Type] リストからルールを選択します。たとえば、HTTP のルールを追加するには、[HTTP] を選択します。すべての TCP トラフィックを許可するルールを追加するには、[All TCP] を選択します。これらのオプションの一部 (HTTP など) については、ポートが自動入力されます。表示されていないプロトコルを使用するには、[Custom Protocol Rule] を選択します。
6. (オプション) カスタムプロトコルルールを作成する場合は、[Protocol] リストからプロトコルの番号または名前を選択します。詳細については、「プロトコル番号の IANA リスト」を参照してください。
7. (オプション) 選択したプロトコルにポート番号が必要な場合、ポート番号またはハイフンで区切ったポート番号の範囲 (49152-65535 など) を入力します。

8. インバウンドルールかアウトバウンドルールかに応じて、[Source] または [Destination] フィールドに、ルールを適用する CIDR の範囲を入力します。
9. [Allow/Deny] リストから、指定したトラフィックを許可するには [ALLOW]、指定したトラフィックを拒否するには [DENY] を選択します。
10. (オプション) 別のルールを追加するには、[Add another rule] を選択し、必要に応じてステップ 4 ~ 9 を繰り返します。
11. 完了したら、[Save] を選択します。

ネットワーク ACL からルールを削除するには

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 [Network ACLs] を選択してから、ネットワーク ACL を選択します。
3. 詳細ペインで、[Inbound Rules] タブまたは [Outbound Rules] タブを選択してから、[Edit] を選択します。削除するルールの [Remove] を選択し、[Save] を選択します。

サブネットとネットワーク ACL の関連付け

ネットワーク ACL のルールを特定のサブネットに適用するには、サブネットをネットワーク ACL と関連付ける必要があります。ネットワーク ACL を複数のサブネットに関連付けることができます。ただし、サブネットに関連付けることができるネットワーク ACL は 1 つだけです。特定の ACL に関連付けられていないサブネットは、デフォルトでデフォルトのネットワーク ACL と関連付けられます。

サブネットをネットワーク ACL と関連付けるには

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 [Network ACLs] を選択してから、ネットワーク ACL を選択します。
3. 詳細ペインの [Subnet Associations] タブで、[Edit] を選択します。ネットワーク ACL に関連付けるサブネットの [Associate] チェックボックスをオンにしてから、[Save] を選択します。

ネットワーク ACL とサブネットの関連付けの解除

サブネットからカスタムネットワーク ACL の関連付けを解除できます。サブネットがカスタムネットワーク ACL から関連付けが解除されると、そのサブネットはデフォルトのネットワーク ACL に自動的に関連付けられます。

サブネットとネットワーク ACL の関連付けを解除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Network ACLs] を選択してから、ネットワーク ACL を選択します。
3. 詳細ペインの [Subnet Associations] タブを選択します。
4. [Edit] を選択して、サブネットの [Associate] チェックボックスをオフにします。[Save] を選択します。

サブネットのネットワーク ACL の変更

サブネットに関連付けられているネットワーク ACL を変更できます。例えば、サブネットを作成すると、初期状態で、そのサブネットにはデフォルトのネットワーク ACL が関連付けられます。このサブネットには、作成したカスタムネットワーク ACL を関連付けることができます。

サブネットのネットワーク ACL を変更した後、サブネット内のインスタンスを終了して再起動する必要はありません。変更は短期間で有効になります。

サブネットのネットワーク ACL の関連付けを変更するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Subnets] を選択し、サブネットを選択します。
3. [Network ACL] タブを選択し、[Edit] を選択します。
4. [Change to (変更する)] リストからサブネットを関連付けるネットワーク ACL を選択して、[Save (保存)] を選択します。

ネットワーク ACL を削除する

ネットワーク ACL に関連付けられているサブネットがない場合にのみ、そのネットワーク ACL を削除できます。デフォルトのネットワーク ACL は削除できません。

ネットワーク ACL を削除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインの [Network ACLs] を選択します。
3. ネットワーク ACL を選択し、[Delete] を選択します。
4. 確認ダイアログボックスで、[Yes, Delete] を選択します。

API とコマンドの概要

このページで説明しているタスクは、コマンドラインまたは API を使用して実行できます。コマンドラインインターフェイスの詳細および利用できる API の一覧については、「Amazon VPC にアクセスする (p. 1)」を参照してください。

VPC のネットワーク ACL を作成する

- create-network-acl (AWS CLI)
- New-EC2NetworkAcl (AWS Tools for Windows PowerShell)

1 つまたは複数のネットワーク ACL について説明する

- describe-network-acls (AWS CLI)
- Get-EC2NetworkAcl (AWS Tools for Windows PowerShell)

ルールをネットワーク ACL に追加する

- create-network-acl-entry (AWS CLI)
- New-EC2NetworkAclEntry (AWS Tools for Windows PowerShell)

ネットワーク ACL からルールを削除する

- delete-network-acl-entry (AWS CLI)
- Remove-EC2NetworkAclEntry (AWS Tools for Windows PowerShell)

ネットワーク ACL の既存のルールを置換する

- replace-network-acl-entry (AWS CLI)

- Set-EC2NetworkAclEntry (AWS Tools for Windows PowerShell)

ネットワーク ACL の関連付けを置換する

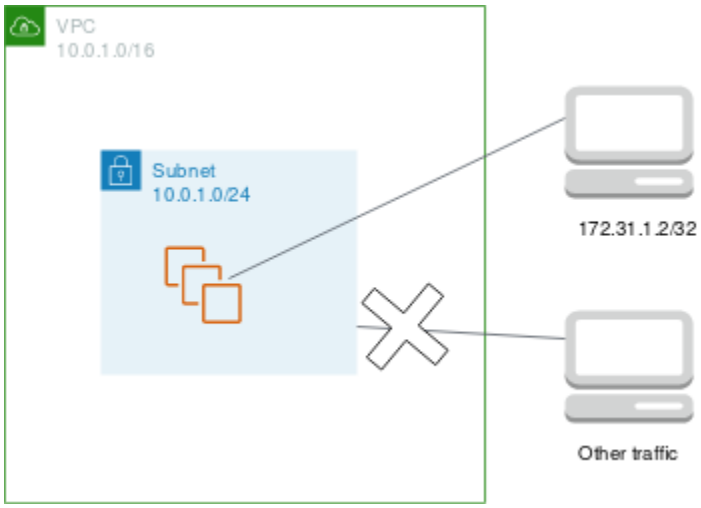
- replace-network-acl-association (AWS CLI)
- Set-EC2NetworkAclAssociation (AWS Tools for Windows PowerShell)

ネットワーク ACL を削除する

- delete-network-acl (AWS CLI)
- Remove-EC2NetworkAcl (AWS Tools for Windows PowerShell)

例: サブネットのインスタンスへのアクセス制御

この例では、サブネットのインスタンスは相互に通信でき、信頼されたリモートコンピュータからアクセス可能です。リモートコンピュータは、ローカルネットワーク内のコンピュータであるか、別のサブネットまたは VPC 内のインスタンスである可能性があります。これを使用して、インスタンスに接続し、管理タスクを実行します。セキュリティグループルールとネットワーク ACL ルールでは、リモートコンピュータの IP アドレス (172.31.1.2/32) からのアクセスを許可します。インターネットまたは他のネットワークからのその他のトラフィックはすべて拒否されます。このシナリオでは、インスタンスのセキュリティグループまたはセキュリティグループルールを変更し、防衛のバックアップレイヤーとしてネットワーク ACL を持つことができます。



インスタンスに関連付けるセキュリティグループの例を次に示します。セキュリティグループはステートフルです。したがって、インバウンドトラフィックへの応答を許可するルールは必要ありません。

インバウンドルール

プロトコルタイプ	プロトコル	ポート範囲	送信元	コメント
すべてのトラフィック	すべて	すべて	sg-1234567890abcdef0	このセキュリティグループに関連付けられたすべてのインスタンスは相互に通信できます。

SSH	TCP	22	172.31.1.2/32	リモートコンピュータからのインバウンド SSH アクセスを許可します。
アウトバウンドルール				
プロトコルタイプ	プロトコル	ポート範囲	送信先	コメント
すべてのトラフィック	すべて	すべて	sg-1234567890abcdef0	このセキュリティグループに関連付けられたすべてのインスタンスは相互に通信できます。

次に、インスタンスのサブネットに関連付けるネットワーク ACL の例を示します。ネットワーク ACL ルールは、サブネット内のすべてのインスタンスに適用されます。ネットワーク ACL はステートレスです。したがって、インバウンドトラフィックへの応答を許可するルールが必要です。

インバウンドルール						
ルール番号	タイプ	プロトコル	ポート範囲	送信元	許可/拒否	コメント
100	SSH	TCP	22	172.31.1.2/32	許可	リモートコンピュータからのインバウンドトラフィックを許可します。
*	すべてのトラフィック	すべて	すべて	0.0.0.0/0	拒否	他のすべてのインバウンドトラフィックを拒否します。
アウトバウンドルール						
ルール番号	タイプ	プロトコル	ポート範囲	送信先	許可/拒否	コメント
100	カスタム TCP	TCP	1024-65535	172.31.1.2/32	許可	リモートコンピュータに対するアウトバウンド応答を許可します。
*	すべてのトラフィック	すべて	すべて	0.0.0.0/0	拒否	他のすべてのアウトバウンドトラフィックを拒否します。

誤ってセキュリティグループルールを過度に制限の低いものにした場合、この例ではネットワーク ACL ルールは指定した IP アドレスからのアクセスのみを許可し続けます。たとえば、次のセキュリティグループには、任意の IP アドレスからのインバウンド SSH アクセスを許可するルールが含まれています。ただし、ネットワーク ACL を使用するサブネット内のインスタンスに、このセキュリティグループを関連付けると、ネットワーク ACL ルールによってサブネットへの他のインバウンドトラフィックが拒否されるため、そのインスタンスにアクセスできるのは、サブネット内およびリモートコンピュータ内の他のインスタンスのみです。

インバウンドルール				
タイプ	プロトコル	ポート範囲	送信元	コメント
すべてのトラフィック	すべて	すべて	sg-1234567890abcdef0	このセキュリティグループに関連付けられたすべてのインスタンスは相互に通信できます。
SSH	TCP	22	0.0.0.0/0	すべての IP アドレスからの SSH アクセスを許可します。
アウトバウンドルール				
タイプ	プロトコル	ポート範囲	送信先	コメント
すべてのトラフィック	すべて	すべて	0.0.0.0/0	すべてのアウトバウンドトラフィックを許可します。

VPC ウィザードシナリオの推奨ルール

Amazon VPC コンソールで VPC ウィザードを使用して、Amazon VPC の一般的なシナリオを実装できます。ドキュメントの説明どおりにこのシナリオを実装した場合、デフォルトのネットワークアクセスコントロールリスト (ACL) を使用することになります。この場合、すべてのインバウンドトラフィックとアウトバウンドトラフィックが許可されます。セキュリティを強化する必要がある場合は、ネットワーク ACL を作成し、ルールを追加できます。詳細については、以下のいずれかを参照してください。

- the section called “1 つのパブリックサブネットを持つ VPC に対する推奨ネットワーク ACL ルール” (p. 27)
- the section called “パブリックサブネットとプライベートサブネット (NAT) を持つ VPC の推奨ネットワーク ACL ルール” (p. 41)
- the section called “パブリックサブネットとプライベートサブネットおよび AWS Site-to-Site VPN アクセスを持つ VPC の推奨ネットワーク ACL ルール” (p. 62)
- the section called “プライベートサブネットのみと AWS Site-to-Site VPN アクセスを持つ VPC の推奨ネットワーク ACL ルール” (p. 76)

VPC のセキュリティのベストプラクティス

以下のベストプラクティスは一般的なガイドラインであり、完全なセキュリティソリューションに相当するものではありません。これらのベストプラクティスはお客様の環境に適切ではないか、十分ではない場合があるため、これらは処方箋ではなく、有用な考慮事項と見なしてください。

一般的なベストプラクティスは以下のとおりです。

- 高可用性を実現するために、複数のアベイラビリティーゾーンデプロイを使用します。
- セキュリティグループとネットワーク ACL を使用します。詳細については、「VPC のセキュリティグループ (p. 188)」および「ネットワーク ACL (p. 200)」を参照してください。
- IAM ポリシーを使用してアクセスを制御します。
- Amazon CloudWatch を VPC コンポーネントと VPN 接続のモニタリングに使用します。
- フローログを使用して、VPC 内のネットワークインターフェイス間で送受信される IP トラフィックに関する情報をキャプチャします。詳細については、「VPC フローログ (p. 325)」を参照してください。

その他のリソース

- ID フェデレーション、IAM ユーザー、IAM ロールを使用して、AWS リソースおよび API へのアクセスを管理します。AWS アクセス認証情報の作成、配布、ローテーション、および取り消しを行うための認証情報管理のポリシーおよび手順を確立します。詳細については、IAM ユーザーガイドの「IAM ベストプラクティス」を参照してください。
- VPC セキュリティに関するよくある質問の回答については、「Amazon VPC のよくある質問」を参照してください。

VPC のネットワーキングコンポーネント

VPC のネットワーキングを設定するには、次のコンポーネントを使用します。

コンポーネント

- インターネットゲートウェイ (p. 221)
- Egress-Only インターネットゲートウェイ (p. 228)
- キャリアゲートウェイ (p. 231)
- VPC の NAT デバイス (p. 236)
- VPC の DHCP オプションセット (p. 266)
- VPC の DNS サポート (p. 271)
- プレフィックスリスト (p. 276)

インターネットゲートウェイ

インターネットゲートウェイは、VPC とインターネットとの間の通信を可能にする VPC コンポーネントであり、冗長性と高い可用性を備えており、水平スケーリングが可能です。

インターネットゲートウェイは 2 つの目的を果たします。1 つは、インターネットでルーティング可能なトラフィックの送信先を VPC のルートテーブルに追加することです。もう 1 つは、パブリック IPv4 アドレスが割り当てられているインスタンスに対してネットワークアドレス変換 (NAT) を行うことです。詳細については、「」を参照してくださいインターネットアクセスを有効にする (p. 221)

インターネットゲートウェイは、IPv4 トラフィックおよび IPv6 トラフィックをサポートしています。ネットワークトラフィックに可用性のリスクや帯域幅の制約が発生することはありません。アカウントでインターネットゲートウェイを設定しても、追加料金は発生しません。

インターネットアクセスを有効にする

VPC のサブネット内のインスタンスでインターネットのアクセスを有効にするには、以下を実行する必要があります。

- インターネットゲートウェイを作成して VPC にアタッチします。
- インターネットバウンドトラフィックをインターネットゲートウェイに転送するルートを、サブネットのルートテーブルに追加します。
- サブネットのインスタンスに、グローバルに一意な IP アドレス (パブリック IPv4 アドレス、Elastic IP アドレス、IPv6 アドレス) が割り当てられていることを確認します。
- ネットワークアクセスコントロールリストとセキュリティグループルールがインスタンス間で関連するトラフィックを許可していることを確認します。

パブリックサブネットおよびプライベートサブネット

サブネットに関連付けられているルートテーブルにインターネットゲートウェイへのルートがある場合、そのサブネットは「パブリックサブネット」と呼ばれます。インターネットゲートウェイへのルートを持たないルートテーブルに関連付けられているサブネットは、「プライベートサブネット」と呼ばれます。

パブリックサブネットのルートテーブルでは、インターネットゲートウェイのルートに、ルートテーブルに明示的に知られていないすべての送信先 (`0.0.0.0/0` の場合は IPv4、`::/0` の場合は IPv6) を指定

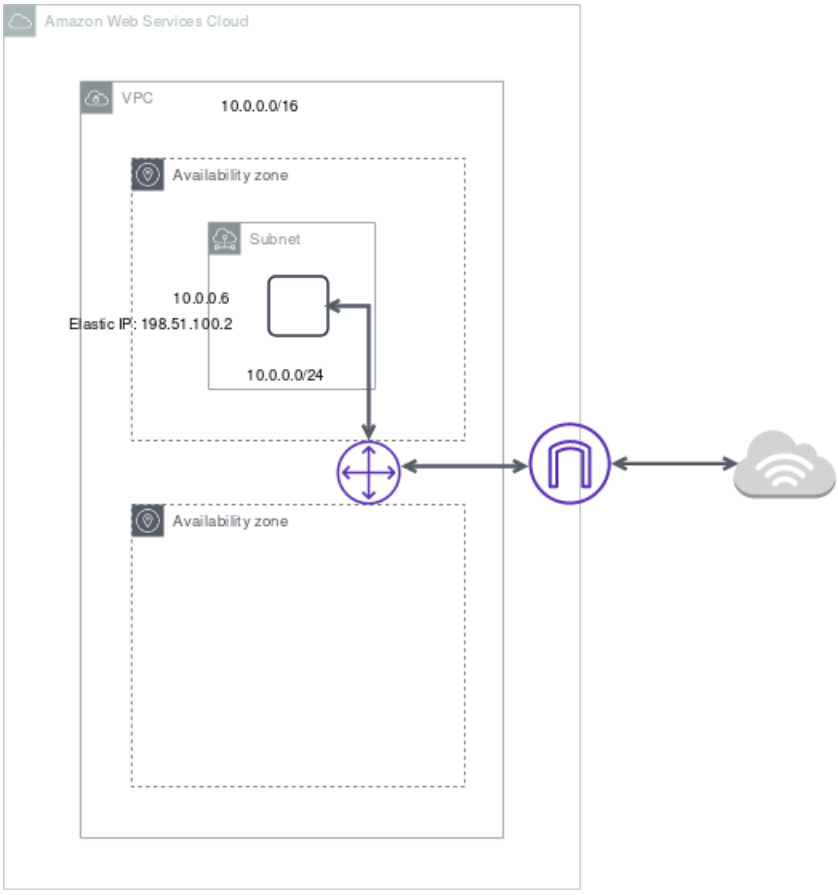
することができます。または、より狭い範囲の IP アドレスにルートを絞り込むこともできます。例えば、AWS 外部にある会社のパブリックエンドポイントのパブリック IPv4 アドレスや、VPC 外部にある他の Amazon EC2 インスタンスの elastic IP アドレスなどです。

IP アドレスおよび NAT

IPv4 でインターネット通信できるようにするには、パブリック IPv4 アドレス、またはインスタンスのプライベート IPv4 アドレスに関連付けられる Elastic IP アドレスが必要です。インスタンスは、VPC とサブネット内で定義されたプライベート（内部）IP アドレス空間のみを認識します。インターネットゲートウェイはインスタンスに代わって 1 対 1 の NAT を論理的に行います。そのため、トラフィックが VPC サブネットから出てインターネットへ向かうとき、返信アドレスフィールドは、インスタンスのプライベート IP アドレスではなくパブリック IPv4 アドレスまたは Elastic IP アドレスに設定されます。逆に、インスタンスのパブリック IPv4 アドレスまたは Elastic IP アドレス宛てのトラフィックは、その送信先アドレスがインスタンスのプライベート IPv4 アドレスに変換されてから、VPC に配信されます。

IPv6 のインターネット経由の通信を有効にするには、VPC およびサブネットは IPv6 CIDR ブロックと関連付け、インスタンスはサブネットの範囲の IPv6 アドレスに割り当てる必要があります。IPv6 アドレスは、グローバルに一意であるため、デフォルトではパブリックアドレスになっています。

次の図では、VPC のサブネット 1 はパブリックサブネットです。サブネット 1 は、カスタムルートテーブルと関連付けられており、インターネット経由の IPv4 トラフィックはすべてインターネットゲートウェイにポイントされます。このインスタンスには、インターネットとの通信を有効にする Elastic IP アドレスが割り当てられています。



パブリック IP アドレスを割り当てずにインスタンスにインターネットアクセスを提供するには、代わりに NAT デバイスを使用できます。NAT デバイスを使用すると、プライベートサブネットのインスタンスはインターネットに接続できますが、インターネット上のホストがインスタンスとの接続を開始できなくなります。詳細については、「」を参照してくださいVPC の NAT デバイス (p. 236)

デフォルトとデフォルト以外の VPC へのインターネットアクセス

次の表では、IPv4 または IPv6 経由でインターネットアクセスに必要なコンポーネントが VPC に自動的に付与されるかどうかについて示します。

コンポーネント	デフォルト VPC	デフォルトではない VPC
インターネットゲートウェイ	はい	最初または 2 番目のオプションを使用して VPC を作成した場合は自動的に付与されます。それ以外の場合は、インターネットゲートウェイを手動で作成してアタッチする必要があります。
IPv4 トラフィックのインターネットゲートウェイ (0.0.0.0/0) にルーティングするルートテーブル。	はい	最初または 2 番目のオプションを使用して VPC を作成した場合は自動的に付与されます。それ以外の場合は、手動でルートテーブルを作成し、ルーティングテーブルを追加する必要があります。
IPv6 トラフィックのインターネットゲートウェイ (::/0) にルーティングするルートテーブル。	いいえ	VPC ウィザードで、最初または 2 番目のオプションを使用して VPC を作成した場合や、IPv6 を VPC CIDR ブロックを関連付けるオプションを指定した場合は自動的に付与されます。それ以外の場合は、手動でルートテーブルを作成し、ルーティングテーブルを追加する必要があります。
サブネットに起動されるインスタンスに自動的に割り当てられたパブリック IPv4 アドレス。	Yes (デフォルトサブネット)	No (デフォルト以外のサブネット)
サブネットに起動されるインスタンスに自動的に割り当てられた IPv6 アドレス。	いいえ (デフォルトサブネット)	No (デフォルト以外のサブネット)

デフォルト VPC の詳細については、「デフォルト VPC とデフォルトサブネット (p. 156)」を参照してください。VPC ウィザードを使用して、インターネットゲートウェイを使う VPC を作成する方法の詳細については、「1 つのパブリックサブネットを持つ VPC (p. 20)」または「パブリックサブネットとプライベートサブネットを持つ VPC (NAT) (p. 31)」を参照してください。

VPC 内の IP アドレス、インスタンスにパブリック IPv4 または IPv6 アドレスを割り当てる方法を制御する方法の詳細は、「VPC の IP アドレス指定 (p. 124)」を参照してください。

新しいサブネットを VPC に追加するとき、サブネットに必要なルーティングとセキュリティを設定する必要があります。

インターネットゲートウェイを VPC に追加する

ここでは、パブリックサブネットを手動で作成し、インターネットアクセスをサポートするために VPC にインターネットゲートウェイをアタッチする方法について説明します。

タスク

サブネットの作成

サブネットを VPC に追加するには

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、[サブネット]、[サブネットの作成] の順に選択します。
3. 必要に応じて、サブネットの詳細を指定します。
   - [Name tag]: 必要に応じてサブネットの名前を入力します。これにより、`Name` というキーと指定した値を含むタグが作成されます。
   - [VPC]: サブネットを作成する VPC を選択します。
   - [Availability Zone] (アベイラビリティーゾーン): 必要に応じて、サブネットが存在するアベイラビリティーゾーンまたはローカルゾーンを選択するか、またはデフォルトの [No Preference] (指定なし) のままにして、AWS がアベイラビリティーゾーンを選択するようにします。

     ローカルゾーンをサポートするリージョンの詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「利用できるリージョン」を参照してください。
   - [IPv4 CIDR ブロック]: サブネットの IPv4 CIDR ブロックを指定します。例: `10.0.1.0/24`。詳細については、「」を参照してくださいIPv4 用の VPC とサブネットのサイズ設定 (p. 109)
   - [IPv6 CIDR ブロック]: (オプション) IPv6 CIDR ブロックを VPC に関連付けている場合は、[Specify a custom IPv6 CIDR] を選択します。16 進法でキーペア値を選択するか、デフォルト値のままにします。
4. [Create] を選択します。

サブネットの詳細については、「VPC とサブネット (p. 106)」を参照してください。

インターネットゲートウェイの作成とアタッチ

インターネットゲートウェイを作成した後で、それを VPC にアタッチします。

インターネットゲートウェイを作成して VPC にアタッチするには

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 [インターネットゲートウェイ] を選択してから、[インターネットゲートウェイの作成] を選択します。
3. オプションで、インターネットゲートウェイに名前を付けます。
4. オプションで、タグを追加または削除します。

   [タグの追加] [タグの追加] を選択して、以下を実行します。
   - [キー] にはキー名を入力します。

- [値] にキー値を入力します。

[タグを削除] タグのキーと値の右側にある [削除] を選択します。

5. [インターネットゲートウェイの作成] を選択します。
6. 作成したインターネットゲートウェイを選択して、[アクション]、[VPC にアタッチ] を選択します。
7. リストから VPC を選択し、[インターネットゲートウェイのアタッチ] を選択します。

カスタムルートテーブルを作成する

サブネットを作成すると、VPC のメインルートテーブルと自動的に関連付けられます。デフォルトでは、メインルートテーブルにインターネットゲートウェイへのルートは含まれません。次の手順では、VPC の外部あてのトラフィックをインターネットゲートウェイに送信するルートを含むカスタムルートテーブルを作成してから、それをサブネットに関連付けます。

カスタムルートテーブルを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Route Tables] (ルートテーブル) を選択して、[Create route table] (ルートテーブルの作成) を選択します。
3. [Create route table] (ルートテーブルの作成) ダイアログボックスで、必要に応じてルートテーブルに名前を指定し、VPC を選んでから、[Create route table] (ルートテーブルの作成) を選択します。
4. 作成したカスタムルートテーブルを選択します。詳細ペインには、ルート、関連付け、ルートのプロパゲーションを操作するタブが表示されます。
5. [Routes] (ルート) タブで、[Edit routes] (ルートの編集)、[Add route] (ルートの追加) の順に選択し、必要に応じて以下のルートを追加します。完了したら、[Save changes] (変更を保存) を選択します。
   - IPv4 トラフィックの場合、[送信先] ボックスで `0.0.0.0/0` を指定し、[ターゲット] リストでインターネットゲートウェイ ID を選択します。
   - IPv6 トラフィックの場合、[送信先] ボックスで `::/0` を指定し、[ターゲット] リストでインターネットゲートウェイ ID を選択します。
6. [Subnet associations] (サブネットの関連付け) タブで [Edit subnet associations] (サブネットの関連付けの編集)を選択し、サブネットのチェックボックスをオンにして、[Save associations] (関連付けを保存) を選択します。

詳細については、「」を参照してくださいVPC のルートテーブル (p. 294)

インターネットアクセス用セキュリティグループの作成

デフォルトでは、VPC セキュリティグループは、すべてのアウトバウンドトラフィックを許可します。新しいセキュリティグループを作成し、インターネットからのインバウンドトラフィックを許可するルールを追加できます。その後、セキュリティグループをパブリックサブネットのインスタンスに関連付けることができます。

セキュリティグループを作成し、インスタンスに関連付けるには

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] を選択して、[Create Security Group] を選択します。
3. [Create Security Group] ダイアログボックスに、セキュリティグループの名前と説明を入力します。[VPC] リストで VPC の ID を選択し、[Yes, Create] を選択します。
4. セキュリティグループを選択します。詳細ペインには、セキュリティグループの詳細と、インバウンドルールとアウトバンドルールを操作するタブが表示されます。

5. [Inbound Rules] タブで、[Edit] を選択します。[Add Rule] を選択し、必要な情報を入力します。たとえば、[Type] (タイプ) リストから [HTTP] または [HTTPS] を選択し、IPv4 トラフィックの場合は `0.0.0.0/0`、IPv6 トラフィックの場合は `::/0` を [Source] (送信元) に入力します。以上が完了したら、[Save] を選択します。
6. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
7. ナビゲーションペインで、[インスタンス] を選択します。
8. インスタンスを選択し、[Actions] を選択して、続いて [Networking] を選択し、次に [Change Security Groups] を選択します。
9. [Change Security Groups] ダイアログボックスで、現在選択しているセキュリティグループのチェックボックスをオフにし、新しいセキュリティグループを選択します。[Assign Security Groups] を選択します。

詳細については、「」を参照してくださいVPC のセキュリティグループ (p. 188)

Elastic IP アドレスのインスタンスへの割り当て

IPv4 経由でインターネットからインスタンスに到達できるようにするには、サブネットでインスタンスを起動した後に、そのインスタンスに Elastic IP アドレスを割り当てる必要があります。

> Note
>
> 起動中にパブリック IPv4 アドレスをインスタンスに割り当てた場合、インスタンスはインターネットから到達可能であるため、Elastic IP アドレスを割り当てる必要はありません。インスタンスの IP アドレスの割り当ての詳細については、「VPC の IP アドレス指定 (p. 124)」を参照してください。

コンソールを使用して、Elastic IP アドレスを配分し、インスタンスに割り当てるには

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 [Elastic IP] を選択します。
3. [Allocate new address] を選択します。
4. [Allocate] を選択します。

   > Note
   >
   > アカウントが EC2-Classic をサポートしている場合には、まず [VPC] を選択します。
5. リストで Elastic IP アドレスを選び、[Actions] を選択してから [Associate address] を選択します。
6. [Instance] または [Network interface] を選択してから、インスタンスまたはネットワークインターフェイス ID を選択します。Elastic IP アドレスを関連付けるプライベート IP アドレスを選択してから、[Associate] を選択します。

詳細については、「」を参照してくださいElastic IP アドレス (p. 288)

VPC からのインターネットゲートウェイのデタッチ

デフォルトではない VPC 内に起動するインスタンスでインターネットアクセスが不要になった場合には、VPC からインターネットゲートウェイをデタッチできます。VPC に関連付けられたパブリック IP アドレスまたは Elastic IP アドレスを持つリソースがある場合、インターネットゲートウェイをデタッチすることはできません。

インターネットゲートウェイをデタッチするには

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 [Elastic IPs] を選択し、Elastic IP アドレスを選択します。

3. [Actions]、[Disassociate address] の順に選択します。[Disassociate address] を選択します。
4. ナビゲーションペインで、[Internet Gateways] を選択します。
5. インターネットゲートウェイを選択し、[アクション]、[VPC からデタッチ] を選択します。
6. [VPC からデタッチ] ダイアログボックスで、[インターネットゲートウェイのデタッチ] を選択します。

インターネットゲートウェイを削除する

インターネットゲートウェイが不要になった場合には、それを削除することができます。VPC にアタッチされているインターネットゲートウェイを削除することはできません。

インターネットゲートウェイを削除するには

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、[Internet Gateways] を選択します。
3. インターネットゲートウェイを選択し、[アクション]、[インターネットゲートウェイの削除] の順に選択します。
4. [インターネットゲートウェイの削除] ダイアログボックスで、「`delete`」と入力し、[インターネットゲートウェイの削除] を選択します。

API とコマンドの概要

このページで説明しているタスクは、コマンドラインまたは API を使用して実行できます。コマンドラインインターフェイスの詳細および利用できる API アクションの一覧については、「Amazon VPC にアクセスする (p. 1)」を参照してください。

インターネットゲートウェイを作成する

- create-internet-gateway (AWS CLI)
- New-EC2InternetGateway (AWS Tools for Windows PowerShell)

インターネットゲートウェイを VPC にアタッチする

- attach-internet-gateway (AWS CLI)
- Add-EC2InternetGateway (AWS Tools for Windows PowerShell)

インターネットゲートウェイについて説明する

- describe-internet-gateways (AWS CLI)
- Get-EC2InternetGateway (AWS Tools for Windows PowerShell)

VPC からインターネットゲートウェイをデタッチする

- detach-internet-gateway (AWS CLI)
- Dismount-EC2InternetGateway (AWS Tools for Windows PowerShell)

インターネットゲートウェイを削除する

- delete-internet-gateway (AWS CLI)
- Remove-EC2InternetGateway (AWS Tools for Windows PowerShell)

Egress-Only インターネットゲートウェイ

Egress-Only インターネットゲートウェイは水平にスケールされ、冗長で、高度な可用性を持つ VPC コンポーネントで、IPv6 経由での VPC からインターネットへの送信を可能にし、インスタンスとの IPv6 接続が開始されるのを防ぎます。

> Note
>
> Egress-Only インターネットゲートウェイは、IPv6 トラフィックでのみ使用されます。IPv4 経由での送信専用のインターネット通信を可能にするには、代わりに NAT ゲートウェイを使用します。詳細については、「」を参照してくださいNAT ゲートウェイ (p. 237)

目次

Egress-Only インターネットゲートウェイの基本

パブリックサブネットのインスタンスは、パブリック IPv4 または IPv6 アドレスがある場合、インターネットゲートウェイを介してインターネットに接続できます。同様に、インターネット上のリソースはパブリック IPv4 アドレスまたは IPv6 アドレスを使用してインスタンスへの接続を開始します。たとえば、インスタンスに接続する場合はローカルコンピュータを使用します。

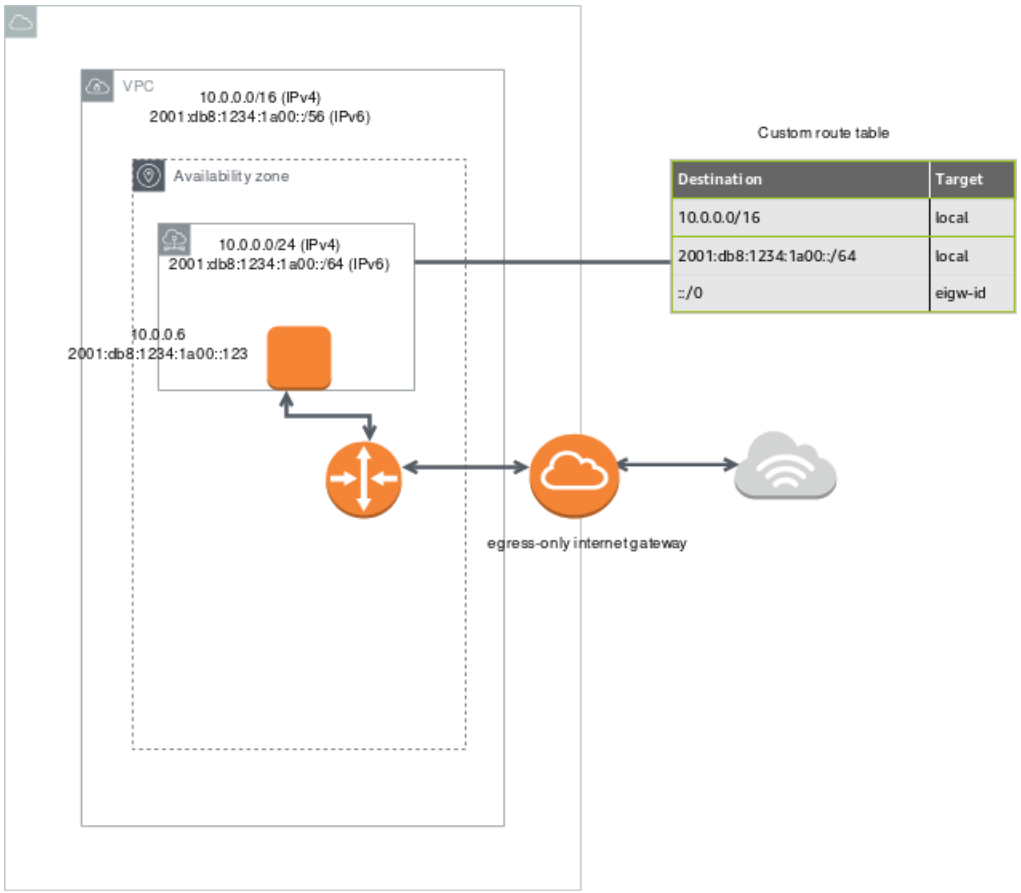
IPv6 アドレスはグローバルに一意であるため、デフォルトではパブリックアドレスになっています。インスタンスにインターネットにアクセスさせる場合で、インターネット上のリソースにインスタンスとの通信を開始させないようにする場合は、Egress-Only インターネットゲートウェイを使用できます。これを行うには、Egress-Only インターネットゲートウェイを VPC で作成し、次にすべての IPv6 トラフィック (`::/0`) または特定の IPv6 アドレスの範囲をポイントするルートテーブルに、Egress-Only インターネットゲートウェイへのルートを追加します。ルートテーブルに関連付けられるサブネットの IPv6 トラフィックは、Egress-Only インターネットゲートウェイにルーティングされます。

Egress-Only インターネットゲートウェイはステートフルです。サブネットのインスタンスからインターネットや他の AWS のサービスに転送し、インスタンスに応答を戻します。

Egress-Only インターネットゲートウェイには、次のプロパティがあります:

- Egress-Only インターネットゲートウェイとセキュリティグループを関連付けることはできません。セキュリティグループは、プライベートサブネットのインスタンスに対して使用し、それらのインスタンスに出入りするトラフィックを管理できます。
- ネットワーク ACL を使用して、Egress-Only インターネットゲートウェイがサブネットとの間でルーティングするトラフィックを制御できます。

以下の図では、VPC に IPv6 CIDR ブロックがあり、VPC のサブネットに IPv6 CIDR ブロックがあります。カスタムルートテーブルはサブネット 1 に関連付けられており、すべてのインターネット宛て IPv6 トラフィック (`::/0`) を VPC の Egress-Only インターネットゲートウェイにルーティングします。



Egress-Only インターネットゲートウェイの操作

以下のタスクでは、プライベートサブネット用の Egress-Only (アウトバウンド) インターネットゲートウェイを作成する方法とサブネットのルーティングを設定する方法について説明します。

タスク

- Egress-Only インターネットゲートウェイを作成する (p. 229)
- Egress-Only インターネットゲートウェイを表示する (p. 230)
- カスタムルートテーブルを作成する (p. 230)
- Egress-Only インターネットゲートウェイを削除する (p. 231)

Egress-Only インターネットゲートウェイを作成する

Amazon VPC コンソールを使用して、VPC 用の Egress-Only インターネットゲートウェイを作成できます。

Egress-Only インターネットゲートウェイを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Egress Only Internet Gateways] を選択します。
3. [Create Egress Only Internet Gateway] を選択します。
4. (オプション) タグを追加または削除します。

[タグの追加] [新しいタグの追加] を選択して、以下を実行します。

- [キー] にはキー名を入力します。
- [値] にキー値を入力します。

[タグを削除] タグのキーと値の右側にある [削除] を選択します。

5. Egress-Only インターネットゲートウェイを作成する VPC を選択します。
6. [Create] を選択します。

Egress-Only インターネットゲートウェイを表示する

Amazon VPC コンソールで、Egress-Only インターネットゲートウェイの情報を表示できます。

Egress-Only インターネットゲートウェイの情報を表示するには

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、[Egress Only Internet Gateways] を選択します。
3. Egress-Only インターネットゲートウェイを選択して、詳細ペインに情報を表示します。

カスタムルートテーブルを作成する

トラフィックを VPC 外の Egress-Only インターネットゲートウェイに送信するには、カスタムルートテーブルを作成して、Egress-Only インターネットゲートウェイへのルートを追加し、それをサブネットに関連付けます。

カスタムルートテーブルを作成してルートを Egress-Only インターネットゲートウェイに追加するには

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) を選択して、[Create Route Table] (ルートテーブルの作成) を選択します。
3. [Create route table] (ルートテーブルの作成) ダイアログボックスで、必要に応じてルートテーブルに名前を指定し、VPC を選んでから、[Create route table] (ルートテーブルの作成) を選択します。
4. 作成したカスタムルートテーブルを選択します。詳細ペインには、ルート、関連付け、ルートのプロパゲーションを操作するタブが表示されます。
5. [Routes] (ルート) タブで [Edit routes] (ルートの編集) を選択し、[Destination] (送信先) ボックスに [`::/0`] を指定します。次に、[Target] (ターゲット) リストで Egress-Only インターネットゲートウェイ ID を選択し、[Save changes] (変更を保存) を選択します。
6. [Subnet associations] (サブネットの関連付け) タブで [Edit subnet associations] (サブネットの関連付けの編集) を選択し、サブネットのチェックボックスをオンにします。[Save] を選択します。

または、サブネットに関連付けられた既存のルーティングテーブルにルートを追加できます。既存のルートテーブルを選択して、上記のステップ 5 と 6 に従って存在をルーティングし、Egress-Only インターネットゲートウェイへのルートを追加します。

ルートテーブルの詳細については、「VPC のルートテーブル (p. 294)」を参照してください。

Egress-Only インターネットゲートウェイを削除する

Egress-Only インターネットゲートウェイが不要になった場合には、それを削除することができます。削除された Egress-Only インターネットゲートウェイをポイントするルートテーブルのルートは、手動で削除するかルートを更新するまで、`blackhole` ステータスのままになります。

Egress-Only インターネットゲートウェイを削除するには

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、[Egress Only インターネットゲートウェイ] を選択して、Egress Only インターネットゲートウェイを選択します。
3. [削除] を選択します。
4. 確認ダイアログボックスで [Delete Egress Only Internet Gateway] を選択します。

API と CLI の概要

このページで説明しているタスクは、コマンドラインまたは API を使用して実行できます。コマンドラインインターフェイスの詳細および利用できる API アクションの一覧については、「Amazon VPC にアクセスする (p. 1)」を参照してください。

Egress-Only インターネットゲートウェイを作成する

- create-egress-only-internet-gateway (AWS CLI)
- New-EC2EgressOnlyInternetGateway (AWS Tools for Windows PowerShell)

Egress-Only インターネットゲートウェイを記述する

- describe-egress-only-internet-gateways (AWS CLI)
- Get-EC2EgressOnlyInternetGatewayList (AWS Tools for Windows PowerShell)

Egress-Only インターネットゲートウェイを削除する

- delete-egress-only-internet-gateway (AWS CLI)
- Remove-EC2EgressOnlyInternetGateway (AWS Tools for Windows PowerShell)

キャリアゲートウェイ

キャリアゲートウェイには 2 つの目的があります。特定の場所にあるキャリアネットワークからのインバウンドトラフィックを許可し、キャリアネットワークおよびインターネットへのアウトバウンドトラフィックを許可します。キャリアゲートウェイを経由した、インターネットから Wavelength Zone へのインバウンド接続設定はありません。

キャリアゲートウェイは IPv4 トラフィックをサポートします。

キャリアゲートウェイは、Wavelength Zone にサブネットを含む VPC でのみ使用できます。キャリアゲートウェイは、通信キャリアネットワーク上の Wavelength Zone と通信事業者、およびデバイス間の接続を提供します。キャリアゲートウェイは、ネットワーク境界グループに割り当てられているプールからキャリア IP アドレスに対して、Wavelength インスタンスの IP アドレスの NAT を実行します。キャリアゲートウェイの NAT 機能は、リージョンでのインターネットゲートウェイの機能に似ています。

通信事業者ネットワークへのアクセスの有効化

Wavelength サブネットのインスタンスに対して、通信事業者ネットワークとの間のアクセスを有効にするには、以下を実行する必要があります。

- VPC を作成します。
- キャリアゲートウェイを作成し、VPC にアタッチします。キャリアゲートウェイを作成するときに、オプションでキャリアゲートウェイにルーティングするサブネットを選択できます。このオプションを選択すると、ルートテーブルやネットワーク ACL など、キャリアゲートウェイに関連するリソースが自動的に作成されます。このオプションを選択しない場合は、次のタスクを実行する必要があります。
  - トラフィックをキャリアゲートウェイにルーティングするサブネットを選択します。
  - サブネットルートテーブルに、トラフィックをキャリアゲートウェイに転送するルートがあることを確認します。
  - サブネット内のインスタンスに、グローバルに一意のキャリア IP アドレスがあることを確認します。
  - ネットワークアクセスコントロールリストとセキュリティグループルールがインスタンス間で関連するトラフィックを許可していることを確認します。

キャリアゲートウェイの操作

以下のセクションでは、キャリアネットワーク (携帯電話など) からのインバウンドトラフィックをサポートし、キャリアネットワークとインターネットへのアウトバウンドトラフィックをサポートする VPC のキャリアゲートウェイを手動で作成する方法について説明します。

タスク

- VPC を作成する (p. 232)
- キャリアゲートウェイの作成 (p. 233)
- 通信事業者ネットワークにアクセスするためのセキュリティグループを作成する (p. 234)
- キャリア IP アドレスを割り当て、Wavelength Zone サブネットのインスタンスに関連付ける (p. 235)
- キャリアゲートウェイの詳細の表示 (p. 235)
- キャリアゲートウェイタグの管理 (p. 236)
- キャリアゲートウェイの削除 (p. 236)

VPC を作成する

空の Wavelength VPC は、次のように作成できます。

Limitation

パブリックにルーティング可能な IPv4 アドレスの範囲を指定できます。ただし、VPC 内のパブリックにルーティング可能な CIDR ブロックからのインターネットへの直接アクセスはサポートしていません。Windows インスタンスは `224.0.0.0` から `255.255.255.255` (クラス D とクラス E の IP アドレス範囲) の VPC では正しく起動できません。

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 [VPC] を選択し、[Create Subnet] を選択します。
3. 次の操作を行い、[Create] を選択します。
   - [Name tag]: オプションで、VPC の名前を指定できます。これにより、`Name` というキーと指定した値を含むタグが作成されます。

- [IPv4 CIDR ブロック]: VPC 用の IPv4 CIDR ブロックを指定します。RFC 1918 で規定されているプライベート (パブリックにルーティングできない) IP アドレス範囲から CIDR ブロックを指定することをお勧めします。たとえば、`10.0.0.0/16` や `192.168.0.0/16` から指定します。

AWS CLI を使用して VPC を作成するには

create-vpc コマンドを使用します。

キャリアゲートウェイの作成

VPC を作成したら、キャリアゲートウェイを作成し、トラフィックをキャリアゲートウェイにルーティングするサブネットを選択します。

Wavelength Zone にオプトインしていない場合、Amazon Virtual Private Cloud Console はオプトインするよう求めます。詳細については、「the section called “ゾーンの管理” (p. 236)」を参照してください。

サブネットからキャリアゲートウェイにトラフィックを自動的にルーティングすることを選択すると、次のリソースが作成されます。

- キャリアゲートウェイ
- サブネット。オプションで、[キー] の値として `Name` を持たないすべてのキャリアゲートウェイタグをサブネットに割り当てることができます。
- 次のリソースを持つネットワーク ACL:
  - Wavelength Zone 内のサブネットに関連付けられたサブネット
  - すべてのトラフィックに対するデフォルトのインバウンドルールとアウトバウンドルール。
- 次のリソースを持つルートテーブル:
  - すべてのローカルトラフィックのルート
  - ローカルではないすべてのトラフィックをキャリアゲートウェイにルーティングするルート
  - サブネットとの関連付け

キャリアゲートウェイを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Carrier Gateways (キャリアゲートウェイ)] を選択し、[Create carrier gateway (キャリアゲートウェイの作成)]を選択します。
3. オプション: [名前] に、キャリアゲートウェイの名前を入力します。
4. [VPC] で、VPC を選択します。
5. [Route subnet traffic to carrier gateway (サブネットトラフィックをキャリアゲートウェイにルーティングする)] を選択し、[Subnets to route (ルーティングするサブネット)] で次の操作を行います。

   a. [Existing subnets in Wavelength Zone (Wavelength Zone の既存のサブネット)] で、キャリアゲートウェイにルーティングする各サブネットのチェックボックスをオンにします。

   b. Wavelength Zone にサブネットを作成するには、[Add new subnet (新しいサブネットの追加)] を選択し、次の情報を指定して、[Add new subnet (新しいサブネットの追加)] を選択します。

      - [Name tag]: 必要に応じてサブネットの名前を入力します。これにより、`Name` というキーと指定した値を含むタグが作成されます。
      - [VPC]: VPC を選択します。
      - [アベイラビリティーゾーン]: Wavelength Zone を選択します。
      - [IPv4 CIDR ブロック]: サブネットの IPv4 CIDR ブロックを指定します。例: `10.0.1.0/24`。
      - サブネットにキャリアゲートウェイタグを適用するには、[Apply same tags from this carrier gateway (このキャリアゲートウェイから同じタグを適用する)] を選択します。

6. (オプション) キャリアゲートウェイにタグを追加するには、[タグの追加] を選択し、次の操作を行います。
   - [キー] にはキー名を入力します。
   - [値] にキー値を入力します。
7. [キャリアゲートウェイの作成] を選択します。

AWS CLI を使用してキャリアゲートウェイを作成するには

1. create-carrier-gateway コマンドを使用します。
2. 次のリソースを持つ VPC ルートテーブルを追加します:
   - すべての VPC ローカルトラフィックのルート
   - ローカルではないすべてのトラフィックをキャリアゲートウェイにルーティングするルート
   - Wavelength Zone 内のサブネットとの関連付け

   詳細については、「」を参照してくださいthe section called “Wavelength ゾーンキャリアゲートウェイへのルーティング” (p. 305)

通信事業者ネットワークにアクセスするためのセキュリティグループを作成する

デフォルトでは、VPC セキュリティグループは、すべてのアウトバウンドトラフィックを許可します。新しいセキュリティグループを作成し、通信事業者からのインバウンドトラフィックを許可するルールを追加できます。次に、セキュリティグループをサブネット内のインスタンスに関連付けます。

新しいセキュリティグループを作成し、インスタンスに関連付けるには

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] を選択して、[Create Security Group] を選択します。
3. セキュリティグループを作成するには、[セキュリティグループの作成] を選択し、次の情報を指定して [作成] を選択します。
   - [セキュリティグループ名]: サブネットの名前を入力します。
   - [説明]: セキュリティグループの説明を入力します。
   - [VPC]: VPC を選択します。
4. セキュリティグループを選択します。詳細ペインには、セキュリティグループの詳細と、インバウンドルールとアウトバンドルールを操作するタブが表示されます。
5. [Inbound Rules] タブで、[Edit] を選択します。[Add Rule] を選択し、必要な情報を入力します。たとえば、[Type] (タイプ) リストから [HTTP] または [HTTPS] を選択し、IPv4 トラフィックの場合は `0.0.0.0/0`、IPv6 トラフィックの場合は `::/0` を [Source] (送信元) に入力します。[Save] を選択します。
6. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
7. ナビゲーションペインで、[インスタンス] を選択します。
8. インスタンスを選択し、[アクション] を選択して、続いて [ネットワーク] を選択し、次に [Change Security Groups (セキュリティグループの変更)] を選択します。
9. 現在選択されているセキュリティグループのチェックボックスをオフにし、新しいセキュリティグループのチェックボックスをオンにします。[Assign Security Groups] を選択します。

AWS CLI を使用してセキュリティグループを作成するには

create-security-group コマンドを使用します。

キャリア IP アドレスを割り当て、Wavelength Zone サブネットのインスタンスに関連付ける

Amazon EC2 コンソールを使用してインスタンスを起動した場合、または AWS CLI で `associate-carrier-ip-address` オプションを使用しなかった場合、キャリア IP アドレスを割り当ててインスタンスに関連付ける必要があります。

AWS CLI を使用してキャリア IP アドレスを割り当てて関連付けるには

1. 次のように、allocate-address コマンドを使用します。

```
aws ec2 allocate-address --region us-east-1 --domain vpc --network-border-group us-
east-1-wl1-bos-wlz-1
```

出力例を次に示します。

```
{
    "AllocationId": "eipalloc-05807b62acEXAMPLE",
    "PublicIpv4Pool": "amazon",
    "NetworkBorderGroup": "us-east-1-wl1-bos-wlz-1",
    "Domain": "vpc",
    "CarrierIp": "155.146.10.111"

}
```

2. 次のように、 associate-address コマンドを使用して、キャリア IP アドレスを EC2 インスタンスに関連付けます。

```
aws ec2 associate-address --allocation-id eipalloc-05807b62acEXAMPLE --network-
interface-id eni-1a2b3c4d
```

出力例を次に示します。

```
{
    "AssociationId": "eipassoc-02463d08ceEXAMPLE",
}
```

キャリアゲートウェイの詳細の表示

状態やタグなど、キャリアゲートウェイに関する情報を表示できます。

キャリアゲートウェイの詳細を表示するには

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、[Carrier Gateways (キャリアゲートウェイ)] を選択します。
3. キャリアゲートウェイを選択し、[アクション]、[詳細の表示] の順に選択します。

AWS CLI を使用して、キャリアゲートウェイの詳細を表示するには

describe-carrier-gateways コマンドを使用します。

キャリアゲートウェイタグの管理

タグは、キャリアゲートウェイの識別に役立ちます。タグを追加または削除できます。

キャリアゲートウェイタグを管理するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Carrier Gateways (キャリアゲートウェイ)] を選択します。
3. キャリアゲートウェイを選択し、[アクション]、[タグの管理] の順に選択します。
4. タグを追加するには、[タグの追加] を選択して、以下を実行します。
   - [キー] にはキー名を入力します。
   - [値] にキー値を入力します。
5. タグを削除するには、タグのキーと値の右側にある [削除] を選択します。
6. [Save] を選択します。

AWS CLI を使用してキャリアゲートウェイタグを管理するには

- タグを追加するには、create-tag コマンドを使用します。
- タグを削除するには、delete-tags コマンドを使用します。

キャリアゲートウェイの削除

不要になったキャリアゲートウェイは削除できます。

> **Important**
>
> キャリアゲートウェイを [ターゲット] とするルートを削除しない場合、ルートはブラックホールルートになります。

キャリアゲートウェイを削除するには

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、[Carrier Gateways (キャリアゲートウェイ)] を選択します。
3. キャリアゲートウェイを選択し、[アクション]、[キャリアゲートウェイの削除] の順に選択します。
4. [キャリアゲートウェイの削除] ダイアログボックスで、「Delete」と入力し、[削除] を選択します。

AWS CLI を使用してキャリアゲートウェイを削除するには

delete-carrier-gateway コマンドを使用します。

ゾーンの管理

リソースまたはサービスの Wavelength Zone を指定する前に、ゾーンにオプトインする必要があります。

Wavelength Zone の使用をオプトインする前に、アクセスをリクエストする必要があります。AWS Wavelength: 詳細については、「」を参照してください。

VPC の NAT デバイス

NAT デバイスを使用して、プライベートサブネット内のインスタンスによるインターネット、他の VPC、またはオンプレミスのネットワークへの接続を許可できます。これらのインスタンスは VPC 外のサービスと通信できますが、未承諾の接続リクエストを受信することはできません。

NAT デバイスは、インスタンスの送信元 IPv4 アドレスを NAT デバイスのアドレスに置き換えます。インスタンスに応答トラフィックを送信するとき、NAT デバイスはアドレスを元の送信元 IPv4 アドレスに変換します。

AWS が提供する NAT ゲートウェイ と呼ばれるマネージド NAT デバイスを使用したり、NAT インスタンス と呼ばれる EC2 インスタンスに独自の NAT デバイスを作成したりすることができます。NAT ゲートウェイを使用すると、可用性と帯域幅が向上し、管理にかかる負担が軽減されるため、NAT ゲートウェイの使用をお勧めします。

Considerations

- NAT デバイスは IPv6 トラフィックには対応していないため、Egress-Only インターネットゲートウェイを使用します。詳細については、「」を参照してくださいEgress-Only インターネットゲートウェイ (p. 228)
- このドキュメントでは、一般的な IT 用語として NAT を使用していますが、NAT デバイスの実際の役割はアドレス変換とポートアドレス変換 (PAT) の両方を兼ねます。

Contents

NAT ゲートウェイ

NAT ゲートウェイは、ネットワークアドレス変換 (NAT) サービスです。NAT ゲートウェイを使用すると、プライベートサブネット内のインスタンスは VPC 外のサービスに接続できますが、外部サービスはそれらのインスタンスとの接続を開始できません。

NAT ゲートウェイを作成するときは、次のいずれかの接続タイプを指定します。

- Public (パブリック) - (デフォルト) プライベートサブネットのインスタンスは、パブリック NAT ゲートウェイを介してインターネットに接続できますが、インターネットから未承諾のインバウンド接続を受信することはできません。パブリックサブネット内にパブリック NAT ゲートウェイを作成し、作成時に Elastic IP アドレスを NAT ゲートウェイに関連付ける必要があります。NAT ゲートウェイへのトラフィックは、VPC のインターネットゲートウェイにルーティングします。パブリック NAT ゲートウェイを使用して、他の VPC やオンプレミスのネットワークに接続することもできます。この場合、NAT ゲートウェイからのトラフィックを Transit Gateway または仮想プライベートゲートウェイ経由でルーティングします。
- Private (プライベート) - プライベートサブネットのインスタンスは、プライベート NAT ゲートウェイを介して他の VPC またはオンプレミスのネットワークに接続できます。この場合、NAT ゲートウェイからのトラフィックを Transit Gateway または仮想プライベートゲートウェイ経由でルーティングできます。elastic IP アドレスをプライベート NAT ゲートウェイに関連付けることはできません。プライベート NAT ゲートウェイを使用して VPC にインターネットゲートウェイをアタッチできますが、プライベート NAT ゲートウェイからインターネットゲートウェイにトラフィックをルーティングすると、インターネットゲートウェイによってトラフィックがドロップされます。

NAT ゲートウェイは、インスタンスの送信元 IP アドレスを NAT ゲートウェイの IP アドレスに置き換えます。パブリック NAT ゲートウェイの場合、これは NAT ゲートウェイの Elastic IP アドレス です。プライベート NAT ゲートウェイの場合、NAT ゲートウェイのプライベート IP アドレスです。インスタンスに応答トラフィックを送信するとき、NAT デバイスはアドレスを元の送信元 IPv4 アドレスに変換します。

Pricing

NAT ゲートウェイをプロビジョニングすると、NAT ゲートウェイが使用可能な時間と、そのゲートウェイが処理するデータ 1 GB ごとに課金されます。詳細については、「Amazon VPC の料金」を参照してください。

次の戦略は、NAT ゲートウェイのデータ転送料金を削減するのに役立ちます。

- AWS リソースがアベイラビリティーゾーン間で大量のトラフィックを送受信する場合は、リソースが NAT ゲートウェイと同じアベイラビリティーゾーンにあることを確認するか、リソースと同じアベイラビリティーゾーンに NAT ゲートウェイを作成してください。
- NAT ゲートウェイを経由するトラフィックのほとんどが、インターフェイスエンドポイントまたはゲートウェイエンドポイントをサポートする AWS サービスへのものである場合、これらのサービスのためにインターフェイスエンドポイントまたはゲートウェイエンドポイントの作成を検討してください。コスト削減の可能性については、「 AWS PrivateLink 料金」を参照してください。

目次

NAT ゲートウェイの基本

各 NAT ゲートウェイは、アベイラビリティーゾーン別に作成され、各ゾーンで冗長性を持たせて実装されます。各アベイラビリティーゾーンに作成できる NAT ゲートウェイの数にはクォータがあります。詳細については、「」を参照してくださいAmazon VPC クォータ (p. 362)

複数のアベイラビリティーゾーンにリソースがあって、1 つの NAT ゲートウェイを共有している場合、その NAT ゲートウェイが属するアベイラビリティーゾーンがダウンすると、その他のアベイラビリティーゾーンのリソースはインターネットにアクセスできなくなります。アベイラビリティーゾーンに依存しないアーキテクチャを作成するには、アベイラビリティーゾーン別に NAT ゲートウェイを作成し、同じアベイラビリティーゾーンに属する NAT ゲートウェイをリソースで使用するようにルーティングを設定します。

NAT ゲートウェイには、次の特性と規則が適用されます。

- NAT ゲートウェイは、プロトコルとして TCP、UDP、ICMP をサポートします。
- NAT ゲートウェイは IPv6 トラフィックでサポートされていないため、送信専用 (Egress-Only) インターネットゲートウェイを使用します。詳細については、「」を参照してくださいEgress-Only インターネットゲートウェイ (p. 228)
- NAT ゲートウェイは 5 Gbps の帯域幅をサポートし、45 Gbps まで自動的に拡張します。これ以上の帯域幅が必要な場合は、リソースを分割して複数のサブネットに配置し、サブネットごとに NAT ゲートウェイを作成できます。
- NAT ゲートウェイは 1 秒あたり 100 万パケットを処理でき、自動的に 1 秒あたり 400 万パケットまで拡張できます。この制限を超えると、NAT ゲートウェイはパケットをドロップします。パケット損失を防ぐには、リソースを分割して複数のサブネットに配置し、サブネットごとに個別の NAT ゲートウェイを作成します。
- NAT ゲートウェイは送信先別に最大 55,000 の同時接続をサポートできます。この制限は、単一の送信先に 1 秒あたり約 900 の接続 (1 分あたり約 55,000 の接続) を作成する場合にも適用されます。送

信先 IP アドレス、送信先ポート、またはプロトコル (TCP/UDP/ICMP) が変更された場合は、追加の 55,000 の接続を作成できます。55,000 を超える接続の場合は、ポートの割り当てエラーによる接続エラーの可能性が高くなります。これらのエラーは、NAT ゲートウェイの `ErrorPortAllocation` CloudWatch メトリクスを表示することでモニタリングできます。詳細については、「」を参照してくださいAmazon CloudWatch を使用した NAT ゲートウェイのモニタリング (p. 245)

- 1 つの elastic IP アドレスを 1 つのパブリック NAT ゲートウェイに関連付けることができます。作成後に NAT ゲートウェイから Elastic IP アドレスの関連付けを解除することはできません。NAT ゲートウェイで別の Elastic IP アドレスを使用するには、新しい NAT ゲートウェイを作成してそのアドレスを関連付け、ルートテーブルを更新します。既存の NAT インスタンスが不要になった場合は、それを削除します。
- プライベート NAT ゲートウェイは、設定されているサブネットから使用可能なプライベート IP アドレスを受け取ります。このプライベート IP アドレスはデタッチできません。また、別のプライベート IP アドレスをアタッチすることもできません。
- NAT ゲートウェイにセキュリティグループを関連付けることはできません。セキュリティグループをインスタンスに関連付けて、インバウンドトラフィックとアウトバウンドトラフィックをコントロールできます。
- NAT ゲートウェイのサブネットに出入りするトラフィックを管理するには、ネットワーク ACL を使用できます。NAT ゲートウェイはポート 1024 ~ 65535 を使用します。詳細については、「」を参照してくださいネットワーク ACL (p. 200)
- NAT ゲートウェイはネットワークインターフェイスを受信し、このネットワークインターフェイスにサブネットの IP アドレス範囲からプライベート IP アドレスが自動的に割り当てられます。NAT ゲートウェイのネットワークインターフェイスは Amazon EC2 コンソールで参照できます。詳細については、「ネットワークインターフェイスに関する詳細の表示」を参照してください。このネットワークインターフェイスの属性を変更することはできません。
- NAT ゲートウェイは、VPC に関連付けられている ClassicLink 接続からはアクセスできません。
- VPC ピア接続、Site-to-Site VPN 接続、または を経由して NAT ゲートウェイにトラフィックをルーティングすることはできませんAWS Direct Connect NAT ゲートウェイは、これらの接続の他方の側にあるリソースからは使用できません。

NAT ゲートウェイの使用を制御する

デフォルトでは、IAM ユーザーには NAT ゲートウェイを使用するためのアクセス許可がありません。NAT ゲートウェイを作成、説明、削除するアクセス許可をユーザーに付与するための IAM ユーザーポリシーを作成できます。詳細については、「」を参照してくださいAmazon VPC の Identity and Access Management (p. 169)

NAT ゲートウェイの使用

Amazon VPC コンソールを使用して、NAT ゲートウェイを作成および管理できます。Amazon VPC ウィザードを使用して、パブリックサブネット、プライベートサブネット、NAT ゲートウェイを使用する VPC を作成することもできます。詳細については、「」を参照してくださいパブリックサブネットとプライベートサブネットを持つ VPC (NAT) (p. 31)

タスク

- NAT ゲートウェイの作成 (p. 239)
- NAT ゲートウェイのタグ付け (p. 240)
- NAT ゲートウェイの削除 (p. 240)

NAT ゲートウェイの作成

NAT ゲートウェイを作成するには、名前 (オプション)、サブネット、および接続タイプ (オプション) を入力します。パブリック NAT ゲートウェイでは、使用可能な Elastic IP アドレスを指定する必要がありま

す。プライベート NAT ゲートウェイは、サブネットからランダムに選択されたプライマリプライベート IP アドレスを受け取ります。プライマリプライベート IP アドレスをデタッチしたり、セカンダリプライベート IP アドレスを追加したりすることはできません。

NAT ゲートウェイを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [NAT ゲートウェイ] を選択します。
3. [NAT ゲートウェイの作成] を選択し、次の操作を行います。
   a. （任意）NAT ゲートウェイの名前を指定します。これにより、キーが **`Name`**、値は指定した名前であるタグが作成されます。
   b. NAT ゲートウェイを作成する先のサブネットを選択します。
   c. [Connectivity type] (接続タイプ) で、プライベート NAT ゲートウェイを作成する場合は [Private] (プライベート)、パブリック NAT ゲートウェイを作成する場合は [Public] (パブリック) (デフォルト) を選択します。
   d. (パブリック NAT ゲートウェイのみ) elastic IP allocation ID (elastic IP の割り当て ID) では、NAT ゲートウェイに関連付ける elastic IP アドレスを選択します。
   e. （オプション）タグごとに、[Add new tag] を選択し、キーの名前と値を入力します。
   f. Create a NAT Gateway (NAT ゲートウェイの作成) を選択します。
4. NAT ゲートウェイの初期ステータスは `Pending` です 。ステータスが `Available` に変わると、NAT ゲートウェイを使用できるようになります。NAT ゲートウェイへのルートをプライベートサブネットのルートテーブルに追加し、NAT ゲートウェイのルートテーブルにルートを追加します。

   NAT ゲートウェイの状態が `Failed` である場合は、作成時にエラーが発生しています。詳細については、「」を参照してくださいNAT ゲートウェイの作成に失敗する (p. 250)

NAT ゲートウェイのタグ付け

NAT ゲートウェイを識別したり、組織のニーズに応じて分類するのに役立つように、NAT ゲートウェイにタグを付けることができます。タグの使用の詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの、「Amazon EC2 リソースのタグ付け」を参照してください。

コスト割り当てタグは、NAT ゲートウェイでサポートされます。そのため、タグを使用して AWS 請求書を整理し、自分のコスト構造を反映することもできます。詳細については、AWS Billing and Cost Management ユーザーガイド の「コスト配分タグの使用」を参照してください。タグによるコスト配分レポートの設定の詳細については、「AWS アカウント請求について」の「毎月のコスト配分レポート」に関する記事を参照してください。

NAT ゲートウェイの削除

不要になった NAT ゲートウェイは削除できます。NAT ゲートウェイを削除すると、そのエントリは Amazon VPC コンソールに 1 時間ほど表示され続けますが、その後自動的に削除されます。このエントリを手動で削除することはできません。

NAT ゲートウェイを削除すると、Elastic IP アドレスとの関連付けは解除されますが、アドレスはアカウントから解放されません。NAT ゲートウェイを削除する場合、NAT ゲートウェイのルートを削除または更新するまで、ルートの状態は `blackhole` になります。

NAT ゲートウェイを削除するには

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 [NAT ゲートウェイ] を選択します。
3. NAT ゲートウェイのラジオボタンを選択し、[アクション]、[NAT ゲートウェイの削除] の順に選択します。

4. 確認を求められたら、「**delete**」と入力し、[削除] を選択します。
5. NAT ゲートウェイに関連付けられた Elastic IP アドレスが不要になった場合は、そのアドレスを解放することをお勧めします。詳細については、「」を参照してくださいElastic IP アドレスを解放する (p. 292)

NAT ゲートウェイシナリオ

次に、パブリック NAT ゲートウェイおよびプライベート NAT ゲートウェイのユースケースの例を示します。

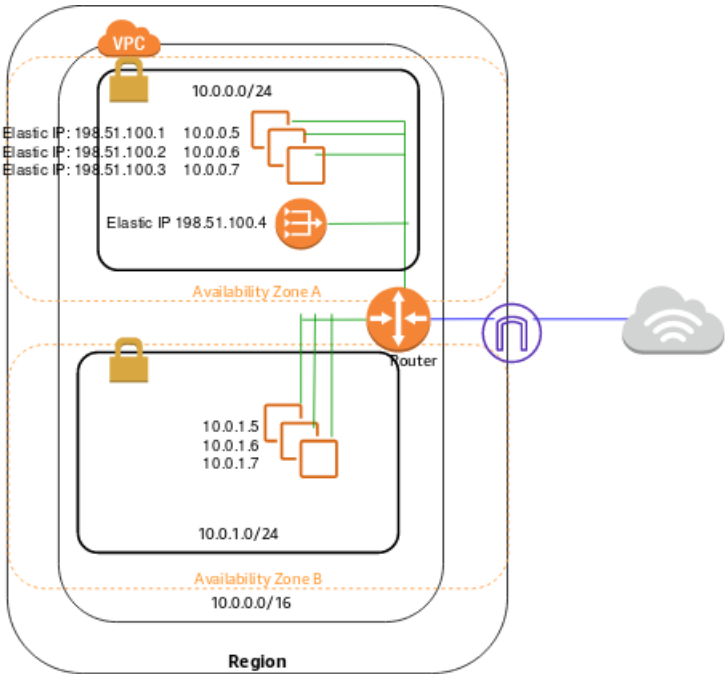
シナリオ

- シナリオ: プライベートサブネットからインターネットにアクセスする (p. 241)
- シナリオ: 許可リストに含まれる IP アドレスからのネットワークへのアクセスを許可する (p. 244)

シナリオ: プライベートサブネットからインターネットにアクセスする

パブリック NAT ゲートウェイを使用して、プライベートサブネット内のインスタンスによるインターネットへのアウトバウンドトラフィックの送信を可能にすることはできますが、インターネットからインスタンスに対する接続の確立はできません。

このユースケースのアーキテクチャを以下に図で示します。アベイラビリティーゾーン A のパブリックサブネットには NAT ゲートウェイが含まれます。アベイラビリティーゾーン B のプライベートサブネットには、インスタンスが含まれます。ルーターは、プライベートサブネットのインスタンスから NAT ゲートウェイにインターネットバウンドトラフィックを送信します。NAT ゲートウェイは、NAT ゲートウェイの elastic IP アドレスを送信元 IP アドレスとして使用し、インターネットゲートウェイにトラフィックを送信します。



以下は、アベイラビリティーゾーン A のパブリックサブネットに関連付けられているルートテーブルです。最初のエントリは、VPC 内のローカルルーティングのデフォルトエントリです。このエントリにより、VPC 内のインスタンスは相互に通信できるようになります。2 番目のエントリは、他のすべてのサブネットトラフィックをインターネットゲートウェイに送信します。これにより、NAT ゲートウェイはインターネットにアクセスできます。

送信先	ターゲット
10.0.0.0/16	ローカル
0.0.0.0/0	internet-gateway-id

以下は、アベイラビリティーゾーン B のプライベートサブネットに関連付けられているルートテーブルです。最初のエントリは、VPC 内のローカルルーティングのデフォルトエントリです。このエントリにより、VPC 内のインスタンスは相互に通信できるようになります。2 番目のエントリは、他のすべてのサブネットトラフィック (インターネットバウンドトラフィックなど) を NAT ゲートウェイに送信します。

送信先	ターゲット
10.0.0.0/16	ローカル
0.0.0.0/0	nat-gateway-id

パブリック NAT ゲートウェイのテスト

NAT ゲートウェイを作成してルートテーブルを更新したら、プライベートサブネットのインスタンスからインターネット上のリモートアドレスに対して ping を送信し、インスタンスがインターネットに接続できることをテストします。これを行う方法の例については、「インターネット接続をテストする (p. 242)」を参照してください。

インターネットに接続できる場合は、さらに以下のように、インターネットトラフィックが NAT ゲートウェイを介してルーティングされているかどうかをテストできます。

- プライベートサブネットのインスタンスからのトラフィックのルートを追跡します。これを行うには、プライベートサブネットの Linux インスタンスから `traceroute` コマンドを実行します。出力で、NAT ゲートウェイのプライベート IP アドレスがホップのいずれか (通常は最初のホップ) に表示されます。
- プライベートサブネットのインスタンスから接続すると、送信元 IP アドレスが表示されるようなサードパーティのウェブサイトやツールを使用します。送信元 IP アドレスとして NAT ゲートウェイの elastic IP アドレスが表示される必要があります。

これらのテストが失敗した場合は、NAT ゲートウェイのトラブルシューティング (p. 250) を参照してください。

インターネット接続をテストする

次の例は、プライベートサブネットのインスタンスからインターネットに接続できるかどうかをテストする方法を示しています。

1. パブリックサブネットのインスタンスを起動します (これを踏み台ホストとして使用します)。詳細については、「」を参照してくださいサブネット内にインスタンスを起動する (p. 120) 起動ウィザードで、Amazon Linux AMI を選択し、インスタンスにパブリック IP アドレスを割り当てます。セキュリティグループルールで、ローカルネットワークの IP アドレス範囲からのインバウンド SSH トラフィック、およびプライベートサブネットの IP アドレス範囲へのアウトバウンド SSH トラフィックが許可されていることを確認します (このテストでは、インバウンドおよびアウトバウンド SSH トラフィックの両方に `0.0.0.0/0` を使用することもできます)。
2. プライベートサブネットのインスタンスを起動します。起動ウィザードで、Amazon Linux AMI を選択します。インスタンスにパブリック IP アドレスを割り当てないでください。パブリックサブネットで起動したインスタンスの IP アドレスからのインバウンド SSH トラフィックとすべてのアウトバウンド ICMP トラフィックが、セキュリティグループのルールで許可されていることを確認します。パブリックサブネットのインスタンスの起動に使用したのと同じキーペアを選択する必要があります。

3. ローカルコンピュータの SSH エージェント転送を設定し、パブリックサブネットの踏み台ホストに接続します。詳細については、「Linux または macOS の SSH エージェント転送を設定するには (p. 243)」または「Windows (PuTTY) 用に SSH エージェント転送を設定するには (p. 243)」を参照してください。
4. 踏み台ホストからプライベートサブネットのインスタンスに接続し、プライベートサブネットのインスタンスからインターネット接続をテストします。詳細については、「」を参照してくださいインターネット接続をテストするには (p. 243)

Linux または macOS の SSH エージェント転送を設定するには

1. ローカルマシンから、認証エージェントにプライベートキーを追加します。

   Linux の場合は、次のコマンドを使用します。

   ```
   ssh-add -c mykeypair.pem
   ```

   macOS の場合は、次のコマンドを使用します。

   ```
   ssh-add -K mykeypair.pem
   ```

2. `-A` オプションを使用してパブリックサブネットのインスタンスに接続して SSH エージェント転送を有効にし、インスタンスのパブリックアドレスを使用します。次に例を示します。

   ```
   ssh -A ec2-user@54.0.0.123
   ```

Windows (PuTTY) 用に SSH エージェント転送を設定するには

1. 既にインストールされていない場合は、PuTTY のダウンロードページから Pageant をダウンロードしてインストールします。
2. プライベートキーを .ppk 形式に変換します。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「PuTTYgen を使用したプライベートキーの交換」を参照してください。
3. Pageant を起動し、タスクバーの Pageant アイコン (非表示の場合があります) を右クリックして、[Add Key] を選択します。作成した .ppk ファイルを選択し、必要に応じてパスフレーズを入力して、[Open (開く)] を選択します。
4. PuTTY セッションを開始し、パブリック IP アドレスを使用してパブリックサブネットのインスタンスに接続します。詳細については、「Linux インスタンスへの接続」を参照してください。[Auth] カテゴリで、必ず [Allow agent forwarding] オプションを選択し、[Private key file for authentication] ボックスは空のままにします。

インターネット接続をテストするには

1. パブリックサブネットのインスタンスから、プライベート IP アドレスを使用して、プライベートサブネットのインスタンスに接続します。次に例を示します。

   ```
   ssh ec2-user@10.0.1.123
   ```

2. プライベートインスタンスから、ICMP が有効なウェブサイトに対して `ping` コマンドを実行して、インターネットに接続できることをテストします。

   ```
   ping ietf.org
   ```

   ```
   PING ietf.org (4.31.198.44) 56(84) bytes of data.
   ```

```
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=1 ttl=47 time=86.0 ms
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=2 ttl=47 time=75.6 ms
...
```

ping コマンドをキャンセルするには、Ctrl + C を押します。ping コマンドが失敗した場合は、「インスタンスがインターネットにアクセスできない (p. 253)」を参照してください。

3. (オプション) 必要がなくなった場合は、インスタンスを終了します。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「インスタンスの終了」を参照してください。

シナリオ: 許可リストに含まれる IP アドレスからのネットワークへのアクセスを許可する

オンプレミスネットワークへのアクセスが許可されている IP アドレス範囲から各インスタンスに個別の IP アドレスを割り当てる代わりに、許可されている IP アドレス範囲で VPC 内にサブネットを作成し、サブネット内にプライベート NAT ゲートウェイを作成し、VPC からのオンプレミスネットワーク宛てのトラフィックを NAT ゲートウェイ経由でルーティングできます。

NAT インスタンスから NAT ゲートウェイに移行する

現在 NATインスタンスを使用している場合は、NAT ゲートウェイに置き換えることをお勧めします。NAT インスタンスと同じサブネットに NAT ゲートウェイを作成し、ルートテーブルを NAT インスタンスを指す既存のルートから NAT ゲートウェイを指すルートに置き換えることができます。現在 NAT インスタンスで使用している同じ Elastic IP アドレスを NAT ゲートウェイで使用するには、まず NAT インスタンスに関連付けられている Elastic IP アドレスを解除し、そのアドレスを ゲートウェイの作成時に NAT ゲートウェイに関連付けます。

NAT インスタンスから NAT ゲートウェイにルーティングを変更したり、NAT インスタンスに関連付けられている Elastic IP アドレスを解除したりすると、現在の接続は切断されるため、再接続する必要があります。重要なタスク (または NAT インスタンスを介してその他のタスク) が実行中でないことを確認してください。

API と CLI の概要

このページで説明しているタスクは、コマンドラインまたは API を使用して実行できます。コマンドラインインターフェイスの詳細と利用可能な API オペレーションの一覧については、「Amazon VPC にアクセスする (p. 1)」を参照してください。

NAT ゲートウェイの作成

- create-nat-gateway (AWS CLI)
- New-EC2NatGateway (AWS Tools for Windows PowerShell)
- CreateNatGateway (Amazon EC2 クエリ API)

NAT ゲートウェイの説明

- describe-nat-gateways (AWS CLI)
- Get-EC2NatGateway (AWS Tools for Windows PowerShell)
- DescribeNatGateways (Amazon EC2 クエリ API)

NAT ゲートウェイのタグ付け

- create-tags (AWS CLI)
- New-EC2Tag (AWS Tools for Windows PowerShell)
- CreateTags (Amazon EC2 クエリ API)

NAT ゲートウェイの削除

- delete-nat-gateway (AWS CLI)
- Remove-EC2NatGateway (AWS Tools for Windows PowerShell)
- DeleteNatGateway (Amazon EC2 クエリ API)

Amazon CloudWatch を使用した NAT ゲートウェイのモニタリング

CloudWatch を使用して NAT ゲートウェイを監視することで、NAT ゲートウェイから情報を収集し、リアルタイムに近い読み取り可能なメトリクスに加工することができます。この情報を使用して、NAT ゲートウェイの監視とトラブルシューティングを行うことができます。NAT ゲートウェイメトリクスデータは 1 分間隔で提供され、統計は 15 か月間記録されます。

Amazon CloudWatch の詳細については、Amazon CloudWatch ユーザーガイドを参照してください。料金の詳細については、「Amazon CloudWatch の料金」を参照してください。

NAT ゲートウェイのメトリクスおよびディメンション

NAT ゲートウェイでは、次のメトリクスを使用できます。

メトリクス	説明
`ActiveConnectionCount`	NAT ゲートウェイ経由の同時アクティブ TCP 接続の合計数。 値が 0 の場合は、NAT ゲートウェイ経由のアクティブな接続がないことを示します。 単位: カウント 統計: 最も有用な統計は `Max` です。
`BytesInFromDestination`	NAT ゲートウェイによって受信された送信先からのバイト数。 `BytesOutToSource` の値が `BytesInFromDestination` の値より少ない場合、NAT ゲートウェイの処理中、またはトラフィックが NAT ゲートウェイによりアクティブにブロックされている間に、データ損失が発生する可能性があります。 単位: バイト 統計: 最も有用な統計は `Sum` です。
`BytesInFromSource`	VPC 内のクライアントから NAT ゲートウェイによって受信されたバイト数。 `BytesOutToDestination` の値が `BytesInFromSource` の値よりも小さい場合、NAT ゲートウェイの処理中にデータ損失が発生する可能性があります。 単位: バイト

メトリクス	説明
	統計: 最も有用な統計は Sum です。
BytesOutToDestination	NAT ゲートウェイ経由で送信先に送信されたバイト数。 値が 0 より大きい場合は、NAT ゲートウェイの背後にあるクライアントからインターネットへのトラフィックがあることを示します。BytesOutToDestination の値が BytesInFromSource の値よりも小さい場合、NAT ゲートウェイの処理中にデータ損失が発生する可能性があります。 単位: バイト 統計: 最も有用な統計は Sum です。
BytesOutToSource	VPC 内の NAT ゲートウェイ経由でクライアントに送信されたバイト数。 値が 0 より大きい場合は、インターネットから NAT ゲートウェイの背後にあるクライアントへのトラフィックがあることを示します。BytesOutToSource の値が BytesInFromDestination の値より少ない場合、NAT ゲートウェイの処理中、またはトラフィックが NAT ゲートウェイによりアクティブにブロックされている間に、データ損失が発生する可能性があります。 単位: バイト 統計: 最も有用な統計は Sum です。
ConnectionAttemptCount	NAT ゲートウェイ経由で行われた接続試行の回数。 ConnectionEstablishedCount の値が ConnectionAttemptCount の値よりも小さい場合は、NAT ゲートウェイの背後にあるクライアントが応答のない新しい接続を確立しようとしたことを示します。 単位: 個 統計: 最も有用な統計は Sum です。
ConnectionEstablishedCount	NAT ゲートウェイ経由で確立された接続の数。 ConnectionEstablishedCount の値が ConnectionAttemptCount の値よりも小さい場合は、NAT ゲートウェイの背後にあるクライアントが応答のない新しい接続を確立しようとしたことを示します。 単位: 個 統計: 最も有用な統計は Sum です。

メトリクス	説明
ErrorPortAllocation	NAT ゲートウェイが送信元ポートを割り当てられなかった回数。 値が 0 より大きい場合は、NAT ゲートウェイ経由の同時接続数が多すぎることを示します。 単位: カウント 統計: 最も有用な統計は Sum です。
IdleTimeoutCount	アクティブな状態からアイドル状態に移行した接続の数。適切に閉じられなかった場合や、直前の 350 秒間にアクティビティがなかった場合、アクティブな接続はアイドル状態に移行します。 値が 0 より大きい場合は、アイドル状態に移行した接続があることを示します。IdleTimeoutCount の値が増加する場合は、NAT ゲートウェイの背後にあるクライアントが無効な接続を再使用している可能性があります。 単位: 個 統計: 最も有用な統計は Sum です。
PacketsDropCount	NAT ゲートウェイによって破棄されたパケットの数。 値が 0 より大きい場合は、NAT ゲートウェイで進行中の一時的な問題を示している可能性があります。この値が NAT ゲートウェイ上の総トラフィックの 0.01% を超える場合は、[AWS service health dashboard] をオンにします。 単位: 個 統計: 最も有用な統計は Sum です。
PacketsInFromDestination	NAT ゲートウェイによって受信された送信先からのパケット数。 PacketsOutToSource の値が PacketsInFromDestination の値より少ない場合、NAT ゲートウェイの処理中、またはトラフィックが NAT ゲートウェイによりアクティブにブロックされている間に、データ損失が発生する可能性があります。 単位: 個 統計: 最も有用な統計は Sum です。

メトリクス	説明
PacketsInFromSource	VPC 内のクライアントから NAT ゲートウェイによって受信されたパケット数。 PacketsOutToDestination の値が PacketsInFromSource の値よりも小さい場合、NAT ゲートウェイの処理中にデータ損失が発生する可能性があります。 単位: 個 統計: 最も有用な統計は Sum です。
PacketsOutToDestination	NAT ゲートウェイ経由で送信先に送信されたパケット数。 値が 0 より大きい場合は、NAT ゲートウェイの背後にあるクライアントからインターネットへのトラフィックがあることを示します。PacketsOutToDestination の値が PacketsInFromSource の値よりも小さい場合、NAT ゲートウェイの処理中にデータ損失が発生する可能性があります。 単位: 個 統計: 最も有用な統計は Sum です。
PacketsOutToSource	VPC 内の NAT ゲートウェイ経由でクライアントに送信されたパケット数。 値が 0 より大きい場合は、インターネットから NAT ゲートウェイの背後にあるクライアントへのトラフィックがあることを示します。PacketsOutToSource の値が PacketsInFromDestination の値より少ない場合、NAT ゲートウェイの処理中、またはトラフィックが NAT ゲートウェイによりアクティブにブロックされている間に、データ損失が発生する可能性があります。 単位: 個 統計: 最も有用な統計は Sum です。

メトリクスデータをフィルタリングするために以下のディメンションを使用します。

ディメンション	説明
NatGatewayId	NAT ゲートウェイ ID でメトリクスデータをフィルタリングします。

NAT ゲートウェイ CloudWatch メトリクスの表示

NAT ゲートウェイのメトリクスは 1 分間隔で CloudWatch に送信されます。メトリクスはまずサービスの名前空間ごとにグループ化され、次に各名前空間内の可能なディメンションの組み合わせごとにグループ化されます。以下のように、NAT ゲートウェイのメトリクスを表示できます。

CloudWatch コンソールを使用してメトリクスを表示するには

1. CloudWatch コンソール (https://console.aws.amazon.com/cloudwatch/) を開きます。
2. ナビゲーションペインで、[Metrics]、[All metrics] を選択します。
3. [NATGateway] メトリクスの名前空間を選択します。
4. メトリクスディメンションを選択します。

を使ってメトリクスを表示するにはAWS CLI

コマンドプロンプトで次のコマンドを使用して、NAT ゲートウェイサービスで利用可能なメトリクスを一覧表示します。

```
aws cloudwatch list-metrics --namespace "AWS/NATGateway"
```

NAT ゲートウェイをモニタリングする CloudWatch のアラームの作成

アラームの状態が変わったときに Amazon SNS メッセージを送信する Amazon CloudWatch のアラームを作成することができます。1 つのアラームで、指定した期間中、1 つのメトリクスを監視します。このアラームは、複数の期間にわたる一定のしきい値とメトリクスの値の関係性に基づき、Amazon SNS トピックに通知を送信します。

例えば、NAT ゲートウェイを出入りするトラフィックの量を監視するアラームを作成できます。次のアラームは、VPC 内のクライアントから NAT ゲートウェイ経由でインターネットに送信されるアウトバウンドトラフィックの量を監視します。15 分間でバイト数が 5,000,000 スレッドに達したときに通知を送信します。

NAT ゲートウェイ経由のアウトバウンドトラフィックのアラームを作成するには

1. CloudWatch コンソール (https://console.aws.amazon.com/cloudwatch/) を開きます。
2. ナビゲーションペインで、[アラーム]、[すべてのアラーム] の順に選択します。
3. [アラームの作成] を選択します。
4. [メトリクスの選択] を選択します。
5. [NATGateway] メトリクス名前空間を選択し、メトリクスディメンションを選択します。メトリクスを表示したら、NAT ゲートウェイに関して [BytesOutToDestination] メトリクスの横にあるチェックボックスをオンにし、その後 [Select metric] を選択します。
6. アラームを以下のように設定して、[Next] (次へ) をクリックします。
   - [統計] で、[合計] を選択します。
   - [Period] で、[15 minutes] を選択します。
   - [Whenever] で、[Greater/Equal] を選択し、しきい値は「5000000」と入力します。
7. [Notification] で、既存の SNS トピックを選択するか、[Create new topic] を選択して新しいトピックを作成します。[Next (次へ)] を選択します。
8. 次のページで、アラームの名前と説明を入力し、[次へ] を選択します。
9. アラームの設定が終わったら、[Create alarm] を選択します。

その他の例として、ポート割り当てをモニタリングし、3 つの連続する 5 分の間で値がゼロより大きい場合に、通知を送信するアラームを作成できます。

ポート割り当てエラーを監視するアラームを作成するには

1. CloudWatch コンソール (https://console.aws.amazon.com/cloudwatch/) を開きます。
2. ナビゲーションペインで、[アラーム]、[すべてのアラーム] の順に選択します。
3. [アラームの作成] を選択します。
4. [メトリクスの選択] を選択します。
5. [NATGateway] メトリクス名前空間を選択し、メトリクスディメンションを選択します。メトリクスを表示したら、NAT ゲートウェイに関して [ErrorPortAllocation] メトリクスの横にあるチェックボックスをオンにし、その後 [Select metric] を選択します。
6. アラームを以下のように設定して、[Next] (次へ) をクリックします。
   - [統計] で、[Maximum] を選択します。
   - [Period] で、[5 minutes] を選択します。
   - [Whenever] で、[Greater] を選択し、しきい値は「0」と入力します。
   - [追加設定]、[Datapoints to alarm] で、「3」と入力します。
7. [Notification] で、既存の SNS トピックを選択するか、[Create new topic] を選択して新しいトピックを作成します。[Next (次へ)] を選択します。
8. 次のページで、アラームの名前と説明を入力し、[次へ] を選択します。
9. アラームの設定が終わったら、[Create alarm] を選択します。

詳細については、『Amazon CloudWatch ユーザーガイド』の「Amazon CloudWatch アラームの使用」を参照してください。

NAT ゲートウェイのトラブルシューティング

以下のトピックでは、NAT ゲートウェイの作成時や使用時によく発生する可能性のある問題のトラブルシューティングについて説明します。

問題点

NAT ゲートウェイの作成に失敗する

Problem

NAT ゲートウェイを作成すると、`Failed` 状態になります。

> **Note**
>
> 障害が発生した NAT ゲートウェイは、通常約 1 時間後に自動的に削除されます。

Cause

NAT ゲートウェイの作成時にエラーが発生しました。返った状態メッセージは、エラーの理由を表します。

Solution

エラーメッセージを表示するには、Amazon VPC コンソールを開き、[NAT ゲートウェイ] を選択します。NAT ゲートウェイのラジオボタンを選択し、[Details] タブで Stateメッセージ を見つけます。

次の表は、Amazon VPC コンソールに示される失敗の考えられる原因のリストです。示された修復手順のいずれかを適用したら、NAT ゲートウェイの作成を再度試すことができます。

表示されるエラー	原因	ソリューション
この NAT ゲートウェイを作成するための十分な空きアドレスがサブネットにありません	指定したサブネットに空きプライベート IP アドレスがありません。NAT ゲートウェイには、サブネットの範囲からプライベート IP アドレスが割り当てられた一つのネットワークインターフェイスが必要です。	Amazon VPC コンソールの [サブネット] ページに移動して、サブネットで使用可能な IP アドレスの数を確認します。[利用可能な IP] は、サブネットの詳細ペインで表示できます。サブネットで空き IP アドレスを作成するには、使用されていないネットワークインターフェイスを終了するか、必要でないインスタンスを削除することができます。
ネットワーク vpc-xxxxxxxx にインターネットゲートウェイがアタッチされていません	NAT ゲートウェイは、インターネットゲートウェイがアタッチされた VPC で作成する必要があります。	インターネットゲートウェイを作成して VPC にアタッチします。詳細については、「」を参照してくださいインターネットゲートウェイの作成とアタッチ (p. 224)
この NAT ゲートウェイに Elastic IP アドレス eipalloc-xxxxxxxx を関連付けられませんでした	指定した Elastic IP アドレスが存在しないか、見つかりませんでした。	Elastic IP アドレスの割り当て ID を調べて正しく入力されていることを確認します。NAT ゲートウェイを作成しているのと同じ AWS リージョンにある elastic IP アドレスを指定していることを確認します。
Elastic IP アドレス eipalloc-xxxxxxxx はすでに関連付けられています	指定した Elastic IP アドレスが別のリソースにすでに関連付けられていて、NAT ゲートウェイに関連付けることはできません。	Elastic IP アドレスに関連付けられているリソースを確認します。Amazon VPC コンソールの [Elastic IP] ページに移動し、インスタンス ID またはネットワークインターフェイス ID に指定された値を表示します。特定のリソースの Elastic IP アドレスが必要ない場合は、その関連付けを解除できます。また、アカウントに新しい Elastic IP アドレスを割り当てることもできます。詳細については、「」を参照してくださいElastic IP アドレスの操作 (p. 289)

表示されるエラー	原因	ソリューション
この NAT ゲートウェイで作成され、内部で使用されているネットワークインターフェイス eni-xxxxxxxx が無効な状態です。もう一度試してください。	NAT ゲートウェイのネットワークインターフェイスの作成中または使用中に問題が発生しました。	このエラーを解決できません。NAT ゲートウェイを作成し直してください。

NAT ゲートウェイクォータ

NAT ゲートウェイを作成しようとすると、次のエラーが表示されます。

```
Performing this operation would exceed the limit of 5 NAT gateways
```

Cause

そのアベイラビリティーゾーンの NAT ゲートウェイの数のクォータに到達しました。

Solution

アカウントでこの NAT ゲートウェイクォータに達した場合は、次のいずれかの操作を実行できます。

- Service Quotas コンソールを使用して、 アベイラビリティーゾーンのクォータごとに NAT ゲートウェイの増加を要求します。
- NAT ゲートウェイの状態を確認します。ステータスが `Pending`、`Available`、`Deleting` のゲートウェイはクォータに含まれます。最近 NAT ゲートウェイを削除した場合は、ステータスが `Deleting` から `Deleted` に変わるまで数分待ちます。NAT ゲートウェイを作成し直します。
- 特定のアベイラビリティーゾーンの NAT ゲートウェイが不要な場合は、まだクォータに達していないアベイラビリティーゾーンで NAT ゲートウェイを作成してみます。

詳細については、「」を参照してくださいAmazon VPC クォータ (p. 362)

Elastic IP アドレスのクォータ

Problem

パブリック NAT ゲートウェイに Elastic IP アドレスを割り当てようとすると、次のエラーが発生します。

```
The maximum number of addresses has been reached.
```

Cause

そのリージョンのアカウントの Elastic IP アドレスの数のクォータに到達している。

Solution

Elastic IP アドレスのクォータに達した場合は、別のリソースに関連付けられている Elastic IP アドレスを解除することができます。または、Service Quotas コンソールを使用して Elastic IPS クォータの増加をリクエストすることもできます 。

アベイラビリティーゾーンがサポートされていない

Problem

NAT ゲートウェイを作成しようとすると、`NotAvailableInZone` エラーが表示されます。

Cause

制約のあるアベイラビリティーゾーン (当社による拡張に制限があるゾーン) で NAT ゲートウェイを作成しようとしている可能性があります。

Solution

これらのアベイラビリティーゾーンでは NAT ゲートウェイはサポートされていません。別のアベイラビリティーゾーンで NAT ゲートウェイを作成し、それを制約のあるゾーンのプライベートサブネットで使用できます。リソースを制約のないアベイラビリティーゾーンに移動し、リソースと NAT ゲートウェイのアベイラビリティーゾーンを同じにすることができます。

NAT ゲートウェイが表示されなくなりました

Problem

作成した NAT ゲートウェイは、Amazon VPC コンソールに表示されなくなりました。

Cause

NAT ゲートウェイの作成中にエラーが発生し、作成に失敗した可能性があります。状態が `Failed` の NAT ゲートウェイは Amazon VPC コンソールに約 1 時間表示されます。1 時間後、自動的に削除されます。

Solution

「NAT ゲートウェイの作成に失敗する (p. 250)」の情報を確認し、新しい NAT ゲートウェイを作成してみてください。

NAT ゲートウェイが ping コマンドに応答しない

Problem

NAT ゲートウェイの Elastic IP アドレスまたはプライベート IP アドレスに、インターネット (家庭用コンピュータなど) や VPC のインスタンスから ping を送信しても、応答がありません。

Cause

NAT ゲートウェイは、プライベートサブネットのインスタンスからインターネットへのトラフィックのみを渡します。

Solution

NAT ゲートウェイが動作していることをテストするには、「パブリック NAT ゲートウェイのテスト (p. 242)」を参照してください。

インスタンスがインターネットにアクセスできない

Problem

NAT ゲートウェイを作成し、手順に従ってテストしましたが、`ping` コマンドが失敗するか、プライベートサブネットのインスタンスがインターネットにアクセスできません。

Causes

この問題の原因として、次のいずれかが考えられます。

- NAT ゲートウェイでトラフィックを処理する準備が整っていません。

- ルートテーブルが正しく構成されていません。
- セキュリティグループまたはネットワーク ACL がインバウンドトラフィックまたはアウトバウンドトラフィックをブロックしています。
- サポートされていないプロトコルを使用しています。

Solution

次の情報を確認します。

- NAT ゲートウェイの状態が `Available` であることを確認します。Amazon VPC コンソールで、[NAT ゲートウェイ] に移動し、詳細ペインの状態情報を参照してください。NAT ゲートウェイの状態が failed である場合は、作成時にエラーが発生した可能性があります。詳細については、「」を参照してくださいNAT ゲートウェイの作成に失敗する (p. 250)
- ルートテーブルが正しく設定されていることを確認します。
  - NAT ゲートウェイはパブリックサブネット内にあって、インターネットトラフィックがインターネットゲートウェイにルーティングされるようにルートテーブルが設定されている必要があります。
  - インスタンスはプライベートサブネット内にあって、インターネットトラフィックが NAT ゲートウェイにルーティングされるようにルートテーブルが設定されている必要があります。
  - インターネットトラフィックの全体または一部を NAT ゲートウェイの代わりに別のデバイスにルーティングするようなエントリがルートテーブルに含まれていないことを確認します。
- プライベートインスタンスのセキュリティグループルールにより、アウトバウンドインターネットトラフィックが許可されていることを確認します。`ping` コマンドを使用するには、ルールにより、アウトバウンド ICMP トラフィックも許可されている必要があります。

  NAT ゲートウェイ自体は、アウトバウンドリクエストと、アウントバウンドリクエストに応じて受信されるトラフィックのすべてを許可します (つまり、ステートフルです)。
- パブリックサブネットとプライベートサブネットに関連付けられているネットワーク ACL に、インバウンドまたはアウトバウンドのインターネットトラフィックをブロックするルールが含まれていないことを確認します。`ping` コマンドを使用するには、ルールにより、インバウンドおよびアウトバウンドの ICMP トラフィックも許可されている必要があります。

  ネットワーク ACL やセキュリティグループのルールによって削除された接続の診断には、フローログを役立てることができます。詳細については、「」を参照してくださいVPC フローログ (p. 325)
- `ping` コマンドは、必ず ICMP が有効になっているホストに対して実行してください。ICMP が有効になっていない場合、応答パケットを受け取ることはできません。これをテストするには、自分のコンピュータのコマンドラインターミナルから同じ `ping` コマンドを実行します。
- インスタンスから他のリソース (プライベートサブネットの他のインスタンスなど) に ping を実行できることを確認します (セキュリティグループルールにより、これが許可されている場合)。
- 接続に TCP、UDP、または ICMP プロトコルのみが使用されていることを確認します。

送信先への TCP 接続が失敗する

Problem

プライベートサブネットのインスタンスから NAT ゲートウェイを介した特定の送信先への TCP 接続の一部は成功しますが、一部は失敗またはタイムアウトします。

Causes

この問題の原因として、次のいずれかが考えられます。

- 送信先エンドポイントがフラグメント化された TCP パケットで応答しています。NAT ゲートウェイは、TCP または ICMP の IP フラグメンテーションをサポートしません。詳細については、「」を参照してくださいNAT ゲートウェイと NAT インスタンスの比較 (p. 264)

- この tcp_tw_recycle オプションは、NAT デバイスの背後から複数の接続がある場合に問題を引き起こすことが知られているリモートサーバーで有効になっています。

Solutions

次の手順を実行して、接続しようとしているエンドポイントがフラグメント化された TCP パケットで応答しているかどうかを確認します。

1. パブリック IP アドレスを持つパブリックサブネットのインスタンスを使用して、特定のエンドポイントからフラグメンテーションを引き起こすのに十分な大きさの応答をトリガーします。
2. エンドポイントがフラグメント化したパケットを送信していることを確認するため、tcpdump ユーティリティを使用します。

   Important

   これらのチェックを実行するには、パブリックサブネットのインスタンスを使用する必要があります。元の接続が失敗したインスタンス、または NAT ゲートウェイまたは NAT インスタンスの背後にあるプライベートサブネットのインスタンスは使用できません。

   大きな ICMP パケットを送信、または受信する診断ツールによって、パケット損失を報告します。例えば、この ping -s 10000 example.com コマンドは NAT ゲートウェイの背後では機能しません。
3. エンドポイントがフラグメント化された TCP パケットを送信している場合、NAT ゲートウェイの代わりに NAT インスタンスを使用できます。

リモートサーバーにアクセスできる場合は、次の手順を実行して、tcp_tw_recycle オプションが有効になっているかどうかを確認できます。

1. サーバーから、以下のコマンドを実行します。

```
cat /proc/sys/net/ipv4/tcp_tw_recycle
```

   出力が 1 の場合、tcp_tw_recycle オプションは有効になっています。
2. tcp_tw_recycle が有効になっている場合は、無効にすることをお勧めします。接続を再使用する必要がある場合は、安全な tcp_tw_reuse を使用することをお勧めします。

リモートサーバーにアクセスできない場合は、プライベートサブネットのインスタンスで tcp_timestamps オプションを一時的に無効にしてテストできます。次に、リモートサーバーに再度接続します。接続が成功した場合、リモートサーバーで tcp_tw_recycle が有効になっているため、以前のエラーが原因であると考えられます。可能であれば、リモートサーバーの所有者に連絡して、このオプションが有効になっているかどうかを確認し、無効にするようにリクエストします。

Traceroute の出力に NAT ゲートウェイのプライベート IP アドレスが表示されない

Problem

インスタンスからインターネットにアクセスできるが、traceroute コマンドを実行すると、出力に NAT ゲートウェイのプライベート IP アドレスが表示されません。

Cause

インスタンスは、インターネットゲートウェイなどの別のゲートウェイを使用してインターネットにアクセスしています。

Solution

インスタンスがあるサブネットのルートテーブルで、次の情報を確認します。

- インターネットトラフィックを NAT ゲートウェイに送信するルートがあることを確認します。
- インターネットトラフィックを他の機器 (仮想プライベートゲートウェイやインターネットゲートウェイなど) に送信するためのより具体的なルートがないことを確認します。

インターネット接続が 350 秒後に中断される

Problem

インスタンスはインターネットにアクセスできますが、350 秒後に接続が切断されます。

Cause

NAT ゲートウェイを使用する接続が 350 秒以上アイドル状態のままになっていると、その接続はタイムアウトします。

接続がタイムアウトになると、NAT ゲートウェイは、NAT ゲートウェイの背後で接続を継続しようとするリソースすべてに RST パケットを返します (FIN パケットは送信しません)。

Solution

接続が中断されないように、接続を介して追加のトラフィックを開始することができます。または、インスタンスで、350 秒未満の値で TCP キープアライブを有効にできます。

IPsec 接続を確立できない

Problem

送信先への IPsec 接続を確立できません。

Cause

NAT ゲートウェイは現在 IPsec プロトコルをサポートしていません。

Solution

NAT トラバーサル (NAT-T) を使用して、IPsec トラフィックを UDP にカプセル化することはできます。これは NAT ゲートウェイでサポートされているプロトコルです。NAT-T および IPsec 設定をテストして、IPsec トラフィックが欠落しないことを検証してください。

追加の接続を開始できない

Problem

NAT ゲートウェイを介した送信先への既存の接続がありますが、それ以上接続を追加で確立することはできません。

Cause

単一の NAT ゲートウェイの同時接続数が上限に達した可能性があります。詳細については、「」を参照してくださいNAT ゲートウェイの基本 (p. 238) プライベートサブネットのインスタンスで多数の接続が作成されると、この上限に達する場合があります。

Solution

次のいずれかを行ってください。

- アベイラビリティーゾーンごとに NAT ゲートウェイを作成し、各ゾーンにクライアントを分散してください。

- パブリックサブネットで追加の NAT ゲートウェイを作成し、クライアントを複数のプライベートサブネットに分散して、それぞれに別の NAT ゲートウェイへのルートを設定します。
- 送信先に対してクライアントが作成できる接続の数を制限します。
- CloudWatch の `IdleTimeoutCount` (p. 245) メトリクスを使用して、アイドル状態の接続の増加を監視します。アイドル状態の接続を閉じてキャパシティーを解放します。

NAT インスタンス

Important

NAT AMI は、2020 年 12 月 31 日に標準サポートが終了した Amazon Linux の最新バージョン 2018.03 に基づいて構築されています。詳細については、ブログ投稿「Amazon Linux AMI のサポート終了」を参照してください。この AMI は、重要なセキュリティ更新だけを受け取ります (定期的な更新はありません)。
既存の NAT AMI を使用する場合は、AWS が NATゲートウェイに移行 (p. 244)することを推奨します。NAT ゲートウェイでは、可用性と帯域幅に優れ、運用管理の手間を軽減できます。NAT インスタンスがユースケースに合致している場合は、独自の NAT AMI を作成できます。詳細については、「NAT ゲートウェイと NAT インスタンスの比較 (p. 264)」を参照してください。

ネットワークアドレス変換を提供する独自の AMI を作成し、AMI を使用して NAT インスタンスとして EC2 インスタンスを起動できます。パブリックサブネットで NAT インスタンスを起動して、プライベートサブネットのインスタンスがインターネットまたは他の AWS サービスへのアウトバウンド IPv4 トラフィックを開始できるようにしますが、インスタンスがインターネットで開始したインバウンドトラフィックを受信しないようにすることができます。

Limitations

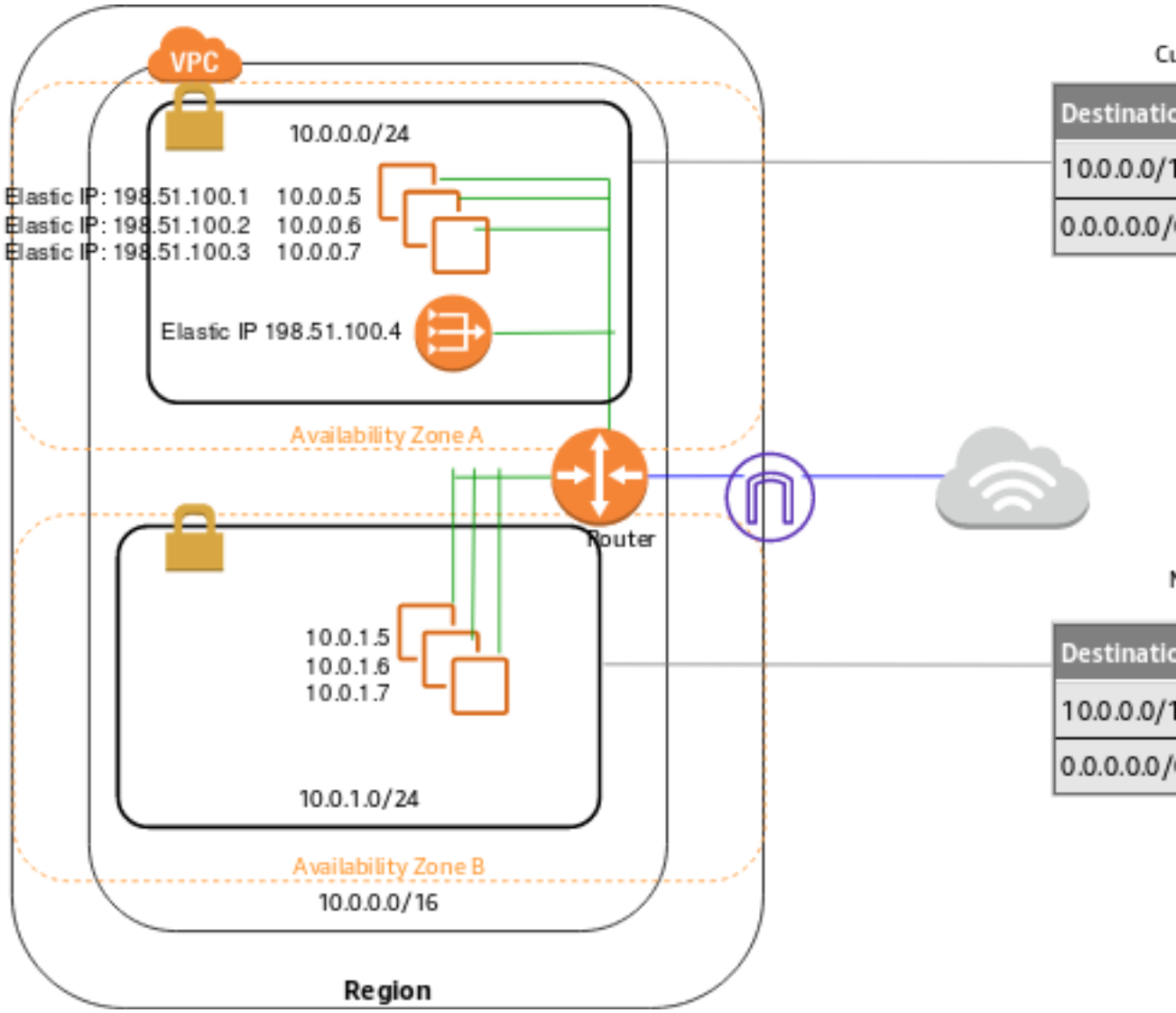
- NAT インスタンスのクォータは、リージョンのインスタンスのクォータによって異なります。キーペアの詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Amazon EC2 のサービスクォータ」を参照してください。
- NAT では IPv6 トラフィックがサポートされていません。Egress-Only インターネットゲートウェイを使用してください。詳細については、「Egress-Only インターネットゲートウェイ (p. 228)」を参照してください。

目次

NAT インスタンスの基本

次の図は、NAT インスタンスの基本を示しています。メインルートテーブルはプライベートサブネットと関連付けられ、プライベートサブネットのインスタンスからパブリックサブネット内の NAT インスタンスにトラフィックを送信します。次に、NAT インスタンスは、そのトラフィックを VPC のインターネットゲートウェイに送信します。トラフィックは NAT インスタンスの Elastic IP アドレスによってもたらされます。NAT インスタンスは応答用に大きなポート番号を指定します。応答が戻ってきた場合、NAT インスタンスはそれをプライベートサブネット内のインスタンスに、応答用のポート番号に基づいて送信します。

プライベートサブネット内のインスタンスからのインターネットトラフィックは NAT インスタンスにルーティングされ、NAT インスタンスはインターネットと通信します。したがって、NAT インスタンスはインターネットにアクセスできる必要があります。また、パブリックサブネット (インターネットゲートウェイへのルートを持つルートテーブルを持つサブネット) に存在し、パブリック IP アドレスまたは Elastic IP アドレスを持っている必要があります。



NAT インスタンスをセットアップする

以下の手順を使用して VPC と NAT インスタンスをセットアップします。

Requirement

開始する前に、NAT インスタンスで実行するように設定する AMI を作成します。NAT を設定する特定のコマンドは、使用するオペレーティングシステムによって異なります。例えば、Amazon Linux 2 では、次のコマンドを使用します。

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo /sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo service iptables save
```

NAT インスタンスをセットアップするには

1. 2 つのサブネットを持つ VPC を作成します。
   a. VPC を作成する (「VPC を作成する (p. 116)」を参照)
   b. 2 つのサブネットを作成する (「サブネットの作成 (p. 224)」を参照)
   c. インターネットゲートウェイを VPC にアタッチする (「インターネットゲートウェイの作成とアタッチ (p. 224)」を参照)
   d. VPC の外部を送信先とするトラフィックをインターネットゲートウェイに送信するカスタムルートテーブルを作成し、1 つのサブネットに関連付けて、パブリックサブネットにする (「カスタムルートテーブルを作成する (p. 225)」を参照)
2. NATSG セキュリティグループを作成します (「NATSG セキュリティグループの作成 (p. 260)」を参照)。このセキュリティグループは、NAT インスタンスの起動時に指定します。
3. NAT インスタンスとして実行されるように設定された AMI からパブリックサブネット内にインスタンスを起動します。
   a. Amazon EC2 コンソールを開きます。
   b. ダッシュボードで、[インスタンスの作成] ボタンを選択し、次のようにウィザードを実行します。
      i. [Choose an Amazon Machine Image (AMI)] (Amazon マシンイメージ (AMI)) ページで、フィルターを [Owned by me] (自己所有) に設定し、AMI を選択します。
      ii. [インスタンスタイプの選択] ページでインスタンスタイプを選択し、[次の手順: インスタンスの詳細の設定] を選択します。
      iii. [インスタンスの詳細の設定] ページで、[ネットワーク] のリストから作成した VPC を選択し、[サブネット] のリストからパブリックサブネットを選択します。
      iv. (オプション) [パブリック IP] チェックボックスをオンにして、NAT インスタンスがパブリック IP アドレスを受け取るように指定します。ここでパブリック IP アドレスを割り当てない場合、Elastic IP アドレスを割り当てて、インスタンスの起動後にそのアドレスをインスタンスに割り当てることができます。起動時のパブリック IP の割り当ての詳細については、「インスタンス起動時のパブリック IPv4 アドレスの割り当て (p. 129)」を参照してください。[次の手順: ストレージの追加] を選択します。
      v. インスタンスにストレージを追加することができます。次のページでは、タグを追加できます。完了したら、[次の手順: セキュリティグループの設定] を選択します。
      vi. [セキュリティグループの設定] ページで、[既存のセキュリティグループを選択する] オプションを選択し、作成した NATSG セキュリティグループを選択します。[Review and Launch] を選択します。
      vii. 選択した設定を確認します。必要な変更を行い、[Launch] を選択し、キーペアを選択してインスタンスを起動します。
4. NAT インスタンスの `SrcDestCheck` 属性を無効にします (「送信元/送信先チェックを無効にする (p. 261)」を参照)。
5. 起動時に NAT インスタンスにパブリック IP アドレスを割り当てていない場合 (手順 3) 、Elastic IP アドレスをそのインスタンスに関連付ける必要があります。
   a. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
   b. ナビゲーションペインで [Elastic IPs] を選択し、[Allocate new address] を選択します。
   c. [Allocate] を選択します。
   d. リストから Elastic IP アドレスを選択し、[Actions]、[Associate address] を選択します。
   e. ネットワークインターフェイスリソースを選択し、NAT インスタンスのネットワークインターフェイスを選択します。[Private IP] リストから Elastic IP アドレスを関連付けるアドレスを選択して、[Associate] を選択します。
6. メインルートテーブルを更新して、NAT インスタンスにトラフィックを送信します。詳細については、「」を参照してくださいメインルートテーブルの更新 (p. 262)

コマンドラインを使用した NAT インスタンスの起動

サブネット内に NAT インスタンスを起動するには、次のいずれかのコマンドを使用します。詳細については、「」を参照してくださいAmazon VPC にアクセスする (p. 1) NAT インスタンスとして実行するように設定した AMI の AMI ID を使用できます。Amazon Linux 2 での AMI の作成方法の詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Amazon EBS-backed AMI の作成」をご参照ください。

- run-instances (AWS CLI)
- New-EC2Instance (AWS Tools for Windows PowerShell)

NATSG セキュリティグループの作成

次の表に示すように NATSG セキュリティグループを定義し、NAT インスタンスを有効にして、インターネットに接続されたトラフィックを、プライベートサブネットのインスタンスから受け取ります。また、SSH トラフィックをネットワークから受け取ります。また、NAT インスタンスは、インターネットにトラフィックを送信することもできます。これにより、プライベートサブネットのインスタンスがソフトウェア更新を取得できます。

推奨されるルールを以下に示します。

インバウンド			
送信元	プロトコル	ポート範囲	コメント
########## CIDR	TCP	80	Allow inbound HTTP traffic from servers in the private subnet
########## CIDR	TCP	443	Allow inbound HTTPS traffic from servers in the private subnet
############ IP ## #####	TCP	22	Allow inbound SSH access to the NAT instance from your network (over the internet gateway)
アウトバウンド			
送信先	プロトコル	ポート範囲	コメント
0.0.0.0/0	TCP	80	Allow outbound HTTP access to the internet
0.0.0.0/0	TCP	443	Allow outbound HTTPS access to the internet

NATSG セキュリティグループを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Security Groups] を選択して、[Create Security Group] を選択します。
3. [Create Security Group] ダイアログボックスで、セキュリティグループ名として `NATSG` を指定し、説明を入力します。[VPC] リストで VPC の ID を選択し、[Yes, Create] を選択します。
4. 先ほど作成した NATSG セキュリティグループを選択します。詳細ペインに、セキュリティグループの詳細と、インバウンドルールおよびアウトバウンドルールを操作するためのタブが表示されます。

5. 次に示すように、[Inbound Rules] タブを使用して、インバウンドトラフィックのルールを追加します。
   a. [Edit] を選択します。
   b. [Add another rule] を選択し、[Type] リストから [HTTP] を選択します。[Source] フィールドで、プライベートサブネットの IP アドレス範囲を指定します。
   c. [Add another rule] を選択し、[Type] リストから [HTTPS] を選択します。[Source] フィールドで、プライベートサブネットの IP アドレス範囲を指定します。
   d. [Add another rule] を選択し、[Type] リストから [SSH] を選択します。[Source] フィールドで、ネットワークのパブリック IP アドレス範囲を指定します。
   e. [Save] を選択します。
6. 次に示すように、[Outbound Rules] タブを使用して、アウトバウンドトラフィックのルールを追加します。
   a. [Edit] を選択します。
   b. [Add another rule] を選択し、[Type] リストから [HTTP] を選択します。[Destination] フィールドで、[0.0.0.0/0] と指定します。
   c. [Add another rule] を選択し、[Type] リストから [HTTPS] を選択します。[Destination] フィールドで、[0.0.0.0/0] と指定します。
   d. [Save] を選択します。

詳細については、「」を参照してくださいVPC のセキュリティグループ (p. 188)

送信元/送信先チェックを無効にする

EC2 インスタンスは、送信元/送信先チェックをデフォルトで実行します。つまり、そのインスタンスは、そのインスタンスが送受信する任意のトラフィックの送信元または送信先である必要があります。しかし、NAT インスタンスは、送信元または送信先がそのインスタンスでないときにも、トラフィックを送受信できなければなりません。したがって、NAT インスタンスでは送信元/送信先チェックを無効にする必要があります。

実行中または停止している NAT インスタンスの `SrcDestCheck` 属性は、コンソールまたはコマンドラインを使用して無効にできます。

コンソールを使用して、送信元/送信先チェックを無効にするには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. NAT インスタンスを選択し、[Actions] (アクション)、[Networking] (ネットワーキング)、[Change source/destination check] (送信元/送信先チェックの変更) を選択します。
4. 送信元/送信先のチェックが停止していることを確認します。それ以外の場合は、[Stop] (停止) を選択します。
5. [Save] を選択します。
6. NAT インスタンスにセカンダリネットワークインターフェイスがある場合は、[Networking] (ネットワーキング) のタブで [Network interfaces] (ネットワークインタフェイス) から選択します。インターフェイス ID を選択して、ネットワークインタフェイスのページに移動します。[Actions] (アクション)、[Change source/dest. check] (送信元/送信先の変更チェック) の順に選択し、[Enable] (有効化) をクリアし、[Save] (保存) を選択します。

コマンドラインを使用して送信元/送信先チェックを無効にするには

次のいずれかのコマンドを使用できます。詳細については、「」を参照してくださいAmazon VPC にアクセスする (p. 1)

- modify-instance-attribute (AWS CLI)
- Edit-EC2InstanceAttribute (AWS Tools for Windows PowerShell)

メインルートテーブルの更新

VPC 内のプライベートサブネットはカスタムルートテーブルに関連付けられないため、メインルートテーブルを使用します。デフォルトでは、メインルートテーブルによって、VPC 内のインスタンスはお互いに通信することができます。その他のすべてのサブネットトラフィックを NAT インスタンスに送信するルートを追加する必要があります。

メインルートテーブルを更新するには

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、[Route Tables] を選択します。
3. VPC のメインルートテーブルを選択します ([Main] 列に [Yes)] と表示されます)。詳細ペインには、ルート、関連付け、ルートのプロパゲーションを操作するタブが表示されます。
4. [Routes] (ルート) タブで、次の手順を実行します。
   a. [Edit routes] (ルートの編集) タブを選択し、[Add route] (ルートの追加) を選択します。
   b. [Destination] (送信先) には「`0.0.0.0/0`」を指定し、[Target] (ターゲット) には NAT インスタンスのインスタンス ID を指定します。
   c. [Save changes] (変更を保存) をクリックします。
5. [Subnet Associations] (サブネットの関連付け) タブで、 [Edit subnet associations] (サブネットの関連付けの編集) を選択します。プライベートサブネットのチェックボックスをオンにして、[Save associations] (関連付けの保存) を選択します。

詳細については、「VPC のルートテーブル (p. 294)」を参照してください。

NAT インスタンスの設定のテスト

NAT インスタンスを起動して上記の設定手順を完了したら、テストを実行し、NAT インスタンスを踏み台サーバーとして使うことで、NAT インスタンス経由でプライベートサブネットのインスタンスからインターネットにアクセスできるかどうかを確認できます。これを行うには、インバウンドおよびアウトバウンドの ICMP トラフィックとアウトバウンドの SSH トラフィックを許可するように NATSG のセキュリティグループルールを更新し、プライベートサブネットでインスタンスを起動します。次に、プライベートサブネットのインスタンスにアクセスするように SSH エージェント転送を設定し、インスタンスに接続して、インターネット接続をテストします。

NAT インスタンスのセキュリティグループを更新するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[Security Groups] を選択します。
3. NAT インスタンスに関連付けられている NATSG セキュリティグループのチェックボックスをオンにします。
4. [Inbound rules] (インバウンドルール) タブで、[Edit inbound rules] (インバウンドルールの編集) を選択します。
5. [Add rule] を選択します。[Type] (タイプ) で [All ICMP - IPv4] (すべての ICMP - IPv4) を選択します。[Source] (送信元) で [Custom] (カスタム) を選択し、プライベートサブネットの IP アドレス範囲を入力します (例: `10.0.1.0/24`)。[Save Rules (ルールの保存)] を選択します。
6. [Outbound rules] (アウトバウンドルール) タブで [Edit outbound rules] (アウトバウンドルールの編集) を選択します。

7. [Add rule] を選択します。[Type] (タイプ) で [SSH] を選択します。[Destination] (送信先) で [Custom] (カスタム) を選択し、プライベートサブネットの IP アドレス範囲を入力します (例: `10.0.1.0/24`)。
8. [Add rule] を選択します。[Type] (タイプ) で [All ICMP - IPv4] (すべての ICMP - IPv4) を選択します。送信先として、Anywhere - IPv4 を選択します。[Save Rules (ルールの保存)] を選択します。

プライベートサブネット内にインスタンスを起動するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. プライベートサブネット内にインスタンスを起動します。詳細については、「」を参照してくださいサブネット内にインスタンスを起動する (p. 120) 起動ウィザードの次のオプションを必ず設定し、[Launch] を選択します。
   - [Choose an Amazon Machine Image (AMI)] ページで、[Quick Start] カテゴリから [Amazon Linux AMI] を選択します。
   - [Configure Instance Details] ページで、[Subnet] リストからプライベートサブネットを選択します。このときに、インスタンスにパブリック IP アドレスを割り当てないでください。
   - [Configure Security Group] ページで、NAT インスタンスのプライベート IP アドレスからの SSH アクセス、またはパブリックサブネットの IP アドレス範囲からの SSH アクセスを許可するインバウンドルールがセキュリティグループに含まれていて、アウトバウンド ICMP トラフィックを許可するアウトバウンドルールがあることを確認します。
   - [Select an existing key pair or create a new key pair] ダイアログボックスで、NAT インスタンスの起動に使用したのと同じキーペアを選択します。

Linux または OS X の SSH エージェント転送を設定するには

1. ローカルマシンから、認証エージェントにプライベートキーを追加します。

   Linux の場合、次のコマンドを使用します。

   ```
   ssh-add -c mykeypair.pem
   ```

   OS X の場合、次のコマンドを使用します。

   ```
   ssh-add -K mykeypair.pem
   ```
2. `-A` オプションを使用して NAT インスタンスに接続し、SSH エージェント転送を有効にします。その例を次に示します。

   ```
   ssh -A ec2-user@54.0.0.123
   ```

Windows (PuTTY) 用に SSH エージェント転送を設定するには

1. 既にインストールされていない場合は、PuTTY のダウンロードページから Pageant をダウンロードしてインストールします。
2. プライベートキーを .ppk 形式に変換します。詳細については、「PuTTYgen を使用してプライベートキーを変換する」を参照してください。
3. Pageant を起動し、タスクバーの Pageant アイコン (非表示の場合があります) を右クリックして、[Add Key] を選択します。作成した .ppk ファイルを選択し、必要に応じてパスフレーズを入力して、[Open] を選択します。
4. PuTTY セッションを開始して NAT インスタンスに接続します。[Auth] カテゴリで、必ず [Allow agent forwarding] オプションを選択し、[Private key file for authentication] フィールドは空のままにします。

インターネット接続をテストするには

1. ICMP が有効なウェブサイトに対して ping コマンドを実行して、NAT インスタンスがインターネットと通信できることをテストします。次に例を示します。

```
ping ietf.org
```

```
PING ietf.org (4.31.198.44) 56(84) bytes of data.
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=1 ttl=48 time=74.9 ms
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=2 ttl=48 time=75.1 ms
...
```

   ping コマンドをキャンセルするには、Ctrl + C を押します。
2. NAT インスタンスから、プライベート IP アドレスを使用してプライベートサブネットのインスタンスに接続します。例:

```
ssh ec2-user@10.0.1.123
```

3. プライベートインスタンスから ping コマンドを実行して、インターネットに接続できることをテストします。

```
ping ietf.org
```

```
PING ietf.org (4.31.198.44) 56(84) bytes of data.
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=1 ttl=47 time=86.0 ms
64 bytes from mail.ietf.org (4.31.198.44): icmp_seq=2 ttl=47 time=75.6 ms
...
```

   ping コマンドをキャンセルするには、Ctrl + C を押します。

   ping コマンドが失敗した場合、次の情報を確認します。

   - NAT インスタンスのセキュリティグループルールで、プライベートサブネットからのインバウンド ICMP トラフィックを許可していることを確認します。許可していない場合、NAT インスタンスはプライベートインスタンスから ping コマンドを受け取ることができません。
   - ルートテーブルが正しく設定されていることを確認します。詳細については、「」を参照してくださいメインルートテーブルの更新 (p. 262)
   - NAT インスタンスの送信元/送信先チェックを無効にしたことを確認します。詳細については、「」を参照してください送信元/送信先チェックを無効にする (p. 261)
   - ICMP を有効にしたウェブサイトに対して ping を実行していることを確認します。そうでない場合、応答パケットを受け取ることはできません。これをテストするには、自分のコンピュータのコマンドラインターミナルから同じ ping コマンドを実行します。
4. （オプション）必要がなくなった場合は、プライベートインスタンスを終了します。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「インスタンスの終了」を参照してください。

NAT ゲートウェイと NAT インスタンスの比較

以下は、NAT ゲートウェイと NAT インスタンスの相違点の概要です。NAT ゲートウェイを使用すると、可用性と帯域幅が向上し、管理にかかる負担が軽減されるため、NAT ゲートウェイの使用をお勧めします。

属性	NAT ゲートウェイ	NAT インスタンス
現在利用できるリージョン	高可用性。各アベイラビリティーゾーンの NAT ゲートウェイは冗長性を持たせて実装されます。アベイラビリティーゾーンごとに NAT ゲートウェイを作成し、ゾーンに依存しないアーキテクチャにします。	スクリプトを使用してインスタンス間のフェイルオーバーを管理します。
帯域幅	45 Gbps まで拡張できます。	インスタンスタイプの帯域幅に依存します。
メンテナンス	によって管理されますAWS ユーザーがメンテナンスを行う必要はありません。	ユーザーが管理します (インスタンスでソフトウェアアップデートやオペレーティングシステムのパッチをインストールするなど)。
パフォーマンス	ソフトウェアは NAT トラフィックを処理するように最適化されます。	一般的な AMI が NAT を実行するように設定されます。
Cost	NAT ゲートウェイの使用数、使用期間、NAT ゲートウェイを通じて送信するデータの量に応じて課金されます。	NATインスタンスの使用数、使用期間、インスタンスタイプとサイズに応じて課金されます。
タイプおよびサイズ	一律提供で、タイプやサイズを決める必要はありません。	予測されるワークロードに応じて適切なインスタンスタイプとサイズを選択します。
パブリック IP アドレス	作成時に パブリック NAT ゲートウェイに関連付ける elastic IP アドレスを選択します。	NAT インスタンスで Elastic IP アドレスまたはパブリック IP アドレスを使用します。インスタンスに新しい Elastic IP アドレスを関連付けることにより、パブリック IP アドレスをいつでも変更できます。
プライベート IP アドレス	ゲートウェイの作成時にサブネットの IP アドレス範囲から自動的に選択されます。	インスタンスの起動時にサブネットの IP アドレス範囲から特定のプライベート IP アドレスを割り当てます。
セキュリティグループ	NAT ゲートウェイにセキュリティグループを関連付けることはできません。NAT ゲートウェイの背後にあるリソースにセキュリティグループを関連付けて、インバウンドトラフィックとアウトバウンドトラフィックをコントロールできます。	NAT インスタンスおよび NAT インスタンスの背後にあるリソースに関連付けてインバウンドトラフィックとアウトバウンドトラフィックをコントロールできます。
ネットワーク ACL	ネットワーク ACL を使用して、NAT ゲートウェイがあるサブネットに出入りするトラフィックをコントロールします。	ネットワーク ACL を使用して、NAT インスタンスがあるサブネットに出入りするトラフィックをコントロールします。
フローログ	フローログを使用してトラフィックをキャプチャします。	フローログを使用してトラフィックをキャプチャします。
ポート転送	サポート外。	設定を手動でカスタマイズしてポート転送をサポートします。
踏み台サーバー	サポート外。	踏み台サーバーとして使用します。
トラフィック	NAT ゲートウェイの CloudWatch メトリクス (p. 245)を表示します。	インスタンスの CloudWatch メトリクスを表示します。

属性	NAT ゲートウェイ	NAT インスタンス
のメトリクス		
タイムアウト動作	接続がタイムアウトになると、NAT ゲートウェイは、NAT ゲートウェイの背後で接続を継続しようとするリソースすべてに RST パケットを返します (FIN パケットは送信しません)。	接続がタイムアウトになると、NAT インスタンスは、接続を閉じるために、NAT インスタンスの背後にあるリソースに FIN パケットを送信します。
IP フラグメント化	UDP プロトコルの IP フラグメント化されたパケットの転送をサポートします。 TCP および ICMP プロトコルのフラグメント化はサポートしていません。これらのプロトコルのフラグメント化されたパケットは削除されます。	TCP、UDP、ICMP プロトコルの IP フラグメント化されたパケットの再アセンブルをサポートします。

VPC の DHCP オプションセット

DHCP (Dynamic Host Configuration Protocol) は、TCP/IP ネットワークのホストに設定情報を渡すための規格です。DHCP メッセージの `options` フィールドには、ドメイン名、ドメインネームサーバー、netbios-node-type などの設定パラメータが含まれます。

VPC を作成する際、DHCP オプションのセットを自動的に作成し、VPC に関連付けます。VPC 用に独自の DHCP オプションセットを設定できます。

目次

DHCP オプションセットの概要

デフォルトでは、デフォルトではない VPC 内のすべてのインスタンスが、AWS によって割り当てられた解決できないホスト名を受け取ります (ip-10-0-0-202 など)。インスタンスに独自のドメイン名を割り当て、独自の DNS サーバーのうち 4 台までを使用できます。そのためには、VPC で使用する DHCP オプションのカスタムセットを作成する必要があります。

DHCP オプションセット用にサポートされるオプションと、VPC 用のデフォルトの DHCP オプションセットで提供される値は次のとおりです。DHCP オプションで必要なオプションのみ指定できます。オプションの詳細については、RFC 2132 を参照してください。

domain-name-servers

最大 4 つのドメインネームサーバーまたは AmazonProvidedDNS (p. 267)の IP アドレス。Amazon が提供する DNS サーバーの IPv4 のアドレスは 169.254.169.253 (または VPC IPv4 ネットワーク範囲のベースにある予約済み IP アドレスに 2 をプラスしたもの) および IPv6 アドレスは fd00:ec2::253 です。

複数のドメインネームサーバーを指定するには、カンマで区切ります。最大 4 つのドメインネームサーバーを指定できますが、オペレーションシステムによっては、制限がより低く適用されている場合があります。

このオプションを使用するには、AmazonProvidedDNS またはカスタムドメインネームサーバーのいずれかに設定します。両方を使用すると、予期しない動作を引き起こす可能性があります。

デフォルトの DHCP オプションセット: AmazonProvidedDNS

domain-name

インスタンスのカスタムドメイン名。AmazonProvidedDNS サーバーを使用しない場合は、必要に応じてカスタムドメインネームサーバーが hostName を解決する必要があります。Amazon Route 53 プライベートホストゾーンを使用する場合は、AmazonProvidedDNS を使用できます。詳細については、「」を参照してくださいVPC の DNS サポート (p. 271)

一部の Linux オペレーティングシステムでは、複数のドメイン名をスペースで区切って指定できます。ただし、他の Linux オペレーティングシステムや Windows では、この値は単一のドメインとして処理されるため、予期しない動作の原因となります。DHCP オプションセットが、すべて同じオペレーティングシステムを実行しているわけではないインスタンスを含む VPC に関連付けられている場合は、ドメイン名を 1 つだけ指定します。

デフォルトの DHCP オプションセット: `us-east-1` の場合、値は `ec2.internal` です。その他のリージョンの場合、値は *`region`*.compute.internal (例: `ap-northeast-1.compute.internal`) です。デフォルト値を使用するには、`domain-name-servers` を AmazonProvidedDNS に設定します。

ntp-servers

最大 4 つまでの Network Time Protocol (NTP) サーバーの IP アドレス。詳細については、RFC 2132 のセクション 8.3 を参照してください。Amazon Time Sync Service は、IPv4 アドレス `169.254.169.123` または IPv6 アドレス `fd00:ec2::123` で指定できます。IPv6 アドレスは、Nitro System 上に構築された EC2 インスタンスでのみアクセス可能です。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「インスタンスの時刻の設定」を参照してください。

デフォルトの DHCP オプションセット: なし

netbios-name-servers

最大 4 つまでの NetBIOS ネームサーバーの IP アドレス。

デフォルトの DHCP オプションセット: なし

netbios-node-type

NetBIOS ノードタイプ (1、2、4、8)。2 つ (ポイントからポイント、または P ノード) を指定することをお勧めします。ブロードキャストとマルチキャストは現在サポートされていません。これらのノードタイプの詳細については、RFC 2132 のセクション 8.7、および RFC1001 のセクション 10 を参照してください。

デフォルトの DHCP オプションセット: なし

Amazon DNS サーバー

VPC 用のデフォルトの DHCP オプションセットには次の 2 つのオプションが含まれています。

- `domain-name-servers=AmazonProvidedDNS`
- `domain-name=`*`domain-name-for-your-region`*

AmazonProvidedDNS は Amazon Route 53 Resolver サーバーです。このオプションは、VPC のインターネットゲートウェイを介した通信を必要とするインスタンスに対して DNS を有効にします。DNS サーバーは、VPC の特定のサブネットまたはアベイラビリティーゾーン内に存在しません。文字列 `AmazonProvidedDNS` は、169.254.169.253 (および VPC IPv4 ネットワークの範囲に 2 をプラスした値のリザーブド IP アドレスで) および fd00:ec2::253 で実行する DNS サーバーにマッピングします。例えば、10.0.0.0/16 ネットワークの DNS サーバーの位置は 10.0.0.2 となります。複数の IPv4 CIDR ブロックを持つ VPC の場合、DNS サーバーの IP アドレスはプライマリ CIDR ブロックにあります。

VPC 内に起動したインスタンスは、インスタンスにプライベート DNS ホスト名を提供します。パブリック IPv4 アドレスを使用してインスタンスが設定されており、VPC DNS 属性が有効になっている場合は、パブリック DNS ホスト名も提供します。DHCP オプション内の `domain-name-servers` が AmazonProvidedDNS に設定されている場合、パブリック DNS ホスト名として、us-east-1 リージョンには `ec2-public-ipv4-address.compute-1.amazonaws.com` 書式、その他のリージョンには `ec2-public-ipv4-address.region.compute.amazonaws.com` 書式が使用されます。プライベートホスト名には、us-east-1 リージョンには `ip-private-ipv4-address.ec2.internal` 書式、その他のリージョンには `ip-private-ipv4-address.region.compute.internal` 書式が使用されます。これらをカスタム DNS ホスト名に変更するには、カスタム DNS サーバーに `domain-name-servers` を設定する必要があります。

VPC の Amazon DNS サーバーは、Route 53 のプライベートホストゾーンで指定する DNS ドメイン名を解決するために使用されます。プライベートホストゾーンの詳細については、Amazon Route 53 デベロッパーガイドの「プライベートホストゾーンの使用」を参照してください。

ルールと考慮事項

Amazon DNS サーバーを使用する場合は、次のルールと考慮事項が適用されます。

- ネットワーク ACL またはセキュリティグループを使用して、Amazon DNS サーバーとの間のトラフィックをフィルタリングすることはできません。
- Amazon EMR のような、Hadoop フレームワークを使用するサービスは、インスタンスが自己の完全修飾ドメイン名 (FQDN) を解決する必要があります。このような場合、`domain-name-servers` オプションがカスタム値に設定されていると DNS 解決が失敗する場合があります。DNS 解決が適切に行われるようにするには、DNS サーバーに条件付きフォワーダーを追加して、*region-name*`.compute.internal` ドメインのクエリがAmazon DNS サーバーに転送されるようにする方法を検討します。詳細については、Amazon EMR 管理ガイドの「クラスターをホストするための VPC をセットアップする」を参照してください。
- Windows Server 2008 では、リンクローカルアドレス範囲 (169.254.0.0/16) にある DNS サーバーは使用できません。
- Amazon Route 53 Resolver は、再帰的な DNS クエリのみをサポートしています。

DHCP オプションの変更

DHCP オプションセットを作成後に変更することはできません。VPC で異なる DHCP オプションセットが必要な場合は、作成して VPC に関連付ける必要があります。または、VPC で DHCP オプションを使用しないことを指定することもできます。

複数セットの DHCP オプションを使用できますが、一度に VPC に関連付けることができる DHCP オプションセットは 1 つだけです。VPC を削除すると、その VPC に関連付けられている DHCP オプションセットは、VPC との関連付けが解除されます。

新しい DHCP オプションセットを VPC に関連付けた後、VPC 内で起動する既存のインスタンスとすべての新しいインスタンスのすべてで、それらの新しいオプションが使用されます。インスタンスを再作成または再起動する必要はありません。インスタンスで DHCP リースが更新される頻度に応じて、数時間以内に自動的に変更が反映されます。インスタンスのオペレーティングシステムを使用してリースを明示的に更新することもできます。

DHCP オプションセットの使用

このセクションでは、DHCP オプションセットの使用方法を示します。

タスク

DHCP オプションセットの作成

必要な数だけ追加の DHCP オプションセットを作成できます。ただし、一度に VPC に関連付けることができる DHCP オプションセットは 1 つだけです。DHCP オプションセットを作成した後、そのセットを使用するように VPC を設定する必要があります。詳細については、「」を参照してくださいVPC が使用する DHCP オプションセットを変更する (p. 269)

DHCP オプションセットを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [DHCP Options Sets] を選択します。
3. [Create DHCP options set] を選択します。
4. [Tag settings] (タグ設定) で、オプションで DHCP オプションセットの名前を入力します。DHCP オプションセットの [Name] (名前) タグを作成します。
5. [DHCP options] (DHCP オプション) で、必要な設定パラメータを入力します。

   Important

   VPC にインターネットゲートウェイがある場合は、[ドメインネームサーバ] の値に必ず独自の DNS サーバーまたは Amazon の DNS サーバー (AmazonProvidedDNS) を指定してください。そうしないと、インターネットと通信する必要があるインスタンスが DNS にアクセスできません。
6. [Tags] (タグ) で、オプションでタグを追加または削除します。
   - [Add a tag] (タグの追加) [Add new tag] (新しいタグを追加) を選択し、キーの名前と値を入力します。
   - [Remove a tag] (タグの削除) タグの横にある [Remove] (削除) を選択します。
7. [Create DHCP options set] を選択します。
8. 新しい DHCP オプションセットの ID を書き留めておいてください (dopt-xxxxxxxx)。この ID は、新しいオプションセットを VPC に関連付けるために必要です。

DHCP オプションセットを作成したので、オプションを有効に機能させるには、オプションを VPC に関連付ける必要があります。複数の DHCP オプションセットを作成できますが、一度に VPC に関連付けることができる DHCP オプションセットは 1 つだけです。

VPC が使用する DHCP オプションセットを変更する

VPC でどの DHCP オプションセットを使用するかを変更できます。新しい DHCP オプションセットを VPC に関連付けた後、VPC 内で起動する既存のインスタンスおよび新しいインスタンスのすべてで、それらの新しいオプションが使用されます。インスタンスを再作成または再起動する必要はありません。イ

ンスタンスで DHCP リースが更新される頻度に応じて、数時間以内に自動的に変更が反映されます。インスタンスのオペレーティングシステムを使用してリースを明示的に更新することもできます。

VPC で DHCP オプションを使用しない場合は、DHCP オプションを使用しないように VPC を変更する (p. 270) を参照してください。

Note

次の手順は、DHCP オプションセットがすでに作成済みであることを前提としています。そうでない場合は、前のセクションの説明に従って、オプションセットを作成してください。

VPC に関連付けられた DHCP オプションセットを変更するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC のチェックボックスを選択して、[Actions] (アクション)、[Edit DHCP options set] (DHCP オプションセットの編集) の順に選択します。
4. [DHCP options set] (DHCP オプションセット) で、DHCP オプションセットを選択します。
5. [Save changes] (変更を保存) をクリックします。

DHCP オプションを使用しないように VPC を変更する

DHCP オプションのセットを使用しないように VPC を設定できます。インスタンスを再作成または再起動する必要はありません。インスタンスで DHCP リースが更新される頻度に応じて、数時間以内に自動的に変更が反映されます。インスタンスのオペレーティングシステムを使用してリースを明示的に更新することもできます。

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC のチェックボックスを選択して、[Actions] (アクション)、[Edit DHCP options set] (DHCP オプションセットの編集) の順に選択します。
4. [DHCP options set] (DHCP オプションセット) で、[No DHCP options set] (DHCP オプションセットなし) を選択します。
5. [Save changes] (変更を保存) をクリックします。

DHCP オプションセットのタグを変更する

タグを使用して、オプションセットを簡単に識別できます。

DHCP オプションセットのタグを変更するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [DHCP オプションセット] を選択します。
3. DHCP オプションセットのラジオボタンを選択し、[Actions] (アクション)、[Manage tags] (タグの管理) の順に選択します。
4. [Tags] (タグ) で、必要に応じてタグを追加または削除します。
   - [Add a tag] (タグの追加) [Add new tag] (新しいタグを追加) を選択し、キーの名前と値を入力します。
   - [Remove a tag] (タグの削除) タグの横にある [Remove] (削除) を選択します。
5. [Save] を選択します。

DHCP オプションセットを削除する

DHCP オプションセットが不要になった場合は、次の手順にしたがって削除します。これらのオプションを使用する VPC を別のオプションセットに変更するか、オプションを変更しないようにしてください。詳細については、「the section called “VPC が使用する DHCP オプションセットを変更する” (p. 269)」および「the section called “DHCP オプションを使用しないように VPC を変更する” (p. 270)」を参照してください。

DHCP オプションセットを削除するには

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 [DHCP Options Sets] を選択します。
3. DHCP オプションセットのラジオボタンを選択し、[Actions] (アクション)、[Delete DHCP options set] (DHCP オプションセットの削除) の順に選択します。
4. 確認を求められたら、[delete] (削除) と入力し、[Delete DHCP options set] (DHCP オプションセットの削除) を選択します。

API とコマンドの概要

このトピックで説明しているタスクは、コマンドラインまたは API を使用して実行できます。コマンドラインインターフェイスの詳細および利用できる API の一覧については、「Amazon VPC にアクセスする (p. 1)」を参照してください。

VPC 用の DHCP オプションセットを作成する

- create-dhcp-options (AWS CLI)
- New-EC2DhcpOption (AWS Tools for Windows PowerShell)

指定した VPC に DHCP オプションセットを関連付ける、または DHCP オプションを使用しないように設定する

- associate-dhcp-options (AWS CLI)
- Register-EC2DhcpOption (AWS Tools for Windows PowerShell)

1 つ以上の DHCP オプションセットについて説明する

- describe-dhcp-options (AWS CLI)
- Get-EC2DhcpOption (AWS Tools for Windows PowerShell)

DHCP オプションセットを削除する

- delete-dhcp-options (AWS CLI)
- Remove-EC2DhcpOption (AWS Tools for Windows PowerShell)

VPC の DNS サポート

ドメインネームシステム (DNS) は、インターネットで使用する名前を対応する IP アドレスに解決するための標準です。DNS ホスト名はコンピュータを一意に識別する絶対名で、ホスト名とドメイン名で構成されます。DNS サーバーは DNS ホスト名を対応する IP アドレスに解決します。

パブリック IPv4 アドレスによってインターネットでの通信が可能になり、プライベート IPv4 アドレスによってインスタンスのネットワーク内部での通信が可能になります。詳細については、「VPC の IP アドレス指定 (p. 124)」を参照してください。

Amazon は、お客様の VPC 用の DNS サーバー (Amazon Route 53 Resolver (p. 267)) を提供しています。代わりに独自の DNS サーバーを使用するには、VPC 用の DHCP オプションの新しいセットを作成します。詳細については、「VPC の DHCP オプションセット (p. 266)」を参照してください。

目次

DNS ホスト名

インスタンスを起動すると、常にプライベート IPv4 アドレスと、プライベート IPv4 アドレスに対応するプライベート DNS ホスト名を受け取ります。インスタンスにパブリック IPv4 アドレスが割り当てられている場合、VPC の DNS 属性は、パブリック IPv4 アドレスに対応するパブリック DNS ホスト名を受け取るかどうかを決定します。詳細については、「VPC 内の DNS 属性 (p. 272)」を参照してください。

Amazon が提供する DNS サーバーを有効にすると、DNS ホスト名が次のように割り当てられ、解決されます。

プライベート DNS ホスト名

インスタンスのプライベート (内部) 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。プライベート DNS ホスト名は、`ip-`*`private-ipv4-address`*`.ec2.internal` リージョンの場合、`us-east-1` の形式になり、他のリージョンの場合は、`ip-`*`private-ipv4-address`*`.`*`region`*`.compute.internal` の形式になります ([*`private-ipv4-address`*] は逆引き参照 IP アドレスです)。プライベート DNS ホスト名は、インスタンス間の通信に使用できます。詳細については、Linux インスタンス用の Amazon EC2 ユーザーガイドの「プライベート IPv4 アドレスと内部 DNS ホスト名」を参照してください。

パブリック DNS ホスト名

インスタンスのパブリック (外部) DNS ホスト名がインスタンスのパブリック IPv4 アドレス (インスタンスのネットワークの外部の場合) およびインスタンスのプライベート IPv4 アドレス (インスタンスのネットワーク内からの場合) を解決します。パブリック DNS ホスト名には、`us-east-1` リージョンには `ec2-`*`public-ipv4-address`*`.compute-1.amazonaws.com` 書式、その他のリージョンには `ec2-`*`public-ipv4-address`*`.`*`region`*`.compute.amazonaws.com` 書式が使用されます。詳細については、Linux インスタンス用の Amazon EC2 ユーザーガイドの「パブリック IPv4 アドレスと外部 DNS ホスト名」を参照してください。

VPC 内の DNS 属性

次の VPC 属性は、VPC に提供される DNS サポートを決定します。両方の属性が有効になっている場合、VPC 内に起動されるインスタンスはパブリック DNS ホスト名を受け取ります。そのためには、インスタンスにパブリック IPv4 アドレスまたは Elastic IP アドレスが割り当てられている必要があります。両方とも有効になっていなかった VPC で両方の属性を有効にすると、その VPC ですでに起動されているインスタンスはパブリック DNS ホスト名を受け取ります。そのためには、インスタンスにパブリック IPv4 アドレスまたは Elastic IP アドレスが割り当てられている必要があります。

これらの属性が VPC で有効かどうかを確認するには、「VPC の DNS 属性の表示と更新 (p. 275)」を参照してください。

属性	説明
`enableDnsHostnames`	VPC がパブリック IP アドレスを持つインスタンスへのパブリック DNS ホスト名の割り当てをサポートするかどうかを決定します。 両方の DNS 属性が `true` の場合、VPC 内のインスタンスはパブリック DNS ホスト名を受け取ります。 この属性のデフォルトは `false` です。ただし、VPC がデフォルト VPC であるか、VPC コンソールウィザードを使用して VPC が作成された場合を除きます。
`enableDnsSupport`	VPC が Amazon 提供の DNS サーバーを介した DNS 解決策をサポートするかどうかを決定します。 この属性が `true` の場合、Amazon が提供した DNS サーバーへのクエリは成功します。詳細については、「Amazon DNS サーバー (p. 267)」を参照してください。 この属性のデフォルトは、VPC の作成方法に関係なく `true` です。

ルールと考慮事項

以下のルールが適用されます。

- 属性の両方が `true` に設定されている場合、次のようになります。
  - パブリック IP アドレスを持つインスタンスは、対応するパブリック DNS ホスト名を受け取ります。
  - Amazon Route 53 Resolver サーバーは、Amazon が提供するプライベート DNS ホスト名を解決できます。
- 少なくとも 1 つの属性が `false` に設定されている場合、次のようになります。
  - パブリック IP アドレスを持つインスタンスは、対応するパブリック DNS ホスト名を受け取りません。
  - Amazon Route 53 Resolver は、Amazon が提供するプライベート DNS ホスト名を解決できません。
  - DHCP オプションセット (p. 266)にカスタムドメイン名がある場合、インスタンスはカスタムプライベート DNS ホスト名を受け取ります。Amazon Route 53 Resolver サーバーを使用しない場合、必要に応じてカスタムドメインネームサーバーがホスト名を解決する必要があります。
- Amazon Route 53 のプライベートホストゾーンで定義されたカスタム DNS ドメイン名を使用する場合や、インターフェイス VPC エンドポイント (AWS PrivateLink) でプライベート DNS を使用する場合は、`enableDnsHostnames` 属性と `enableDnsSupport` 属性の両方を `true` に設定する必要があります。
- Amazon Route 53 Resolver は、プライベート DNS ホスト名を、すべてのアドレス空間のプライベート IPv4 アドレスに解決できます。これには、VPC の IPv4 アドレス範囲が、RFC 1918 に指定されているプライベート IPv4 アドレス範囲外になる場合も含まれます。ただし、2016 年 10 月より前に作成した VPC の場合、その IPv4 アドレス範囲がこれらの範囲外であると、Amazon Route 53 Resolver はプライベート DNS ホスト名を解決しません。このサポートを有効にするには、AWS Support までお問い合わせください。

DNS クォータ

各 EC2 インスタンスは Route 53 Resolver (具体的には 10.0.0.2 などの .2 アドレス、および 169.254.169.253) へパケット数をネットワークインターフェイスあたり 1024 パケット/秒を送信できま

す。このクォータを増やすことはできません。Route 53 Resolver でサポートされる 1 秒あたりの DNS クエリの数は、クエリのタイプ、レスポンスのサイズ、および使用中のプロトコルにより異なります。スケーラブルな DNS アーキテクチャの詳細および推奨については、「アクティブディレクトリを使用した AWS ハイブリッド DNS 技術ガイド」を参照してください。

クォータに達すると、Route 53 Resolver はトラフィックを拒否します。クォータに達する原因には、DNS スロットリングの問題や、Route 53 Resolver ネットワークインターフェイスを使用するインスタンスメタデータクエリがあります。VPC DNS スロットリングの問題を解決する方法については、「VPC DNS スロットリングが、Amazon が提供している DNS サーバーへのDNS クエリの失敗の原因となっているかどうかを判断する方法を教えてください。」を参照してください。インスタンスメタデータの詳細については、「Amazon EC2 Linux インスタンス用ユーザーガイド」の「Retrieve instance metadata (インスタンスメタデータの取得)」を参照してください。

EC2 インスタンスの DNS ホスト名を表示する

Amazon EC2 コンソールまたはコマンドラインを使用して、実行中のインスタンスまたはネットワークインターフェイスの DNS ホスト名を確認できます。

[Public DNS (IPv4) (パブリック DNS (IPv4))] フィールドと [Private DNS (プライベート DNS)] フィールドは、インスタンスに関連付けられている VPC で DNS オプションが有効になっている場合に使用できます。詳細については、「」を参照してくださいthe section called “VPC 内の DNS 属性” (p. 272)

Instance

コンソールを使用してインスタンスの DNS ホスト名を確認するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[インスタンス] を選択します。
3. リストから インスタンスを選択します。
4. 詳細ペインで、[Public DNS (IPv4)] および [Private DNS] フィールドに、該当する場合は DNS ホスト名が表示されます。

コマンドラインを使用してインスタンスの DNS ホスト名を確認するには

次のいずれかのコマンドを使用できます。これらのコマンドラインインターフェイスの詳細については、Amazon VPC にアクセスする (p. 1) を参照してください。

- describe-instances (AWS CLI)
- Get-EC2Instance (AWS Tools for Windows PowerShell)

ネットワークインターフェイス

コンソールを使用してネットワークインターフェイスのプライベート DNS ホスト名を確認するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. リストからネットワークインターフェイスを選択します。
4. 詳細ペインの [プライベート DNS (IPv4)] フィールドにプライベート DNS ホスト名が表示されます。

コマンドラインを使用してネットワークインターフェイスの DNS ホスト名を確認するには

次のいずれかのコマンドを使用できます。これらのコマンドラインインターフェイスの詳細については、「Amazon VPC にアクセスする (p. 1)」を参照してください。

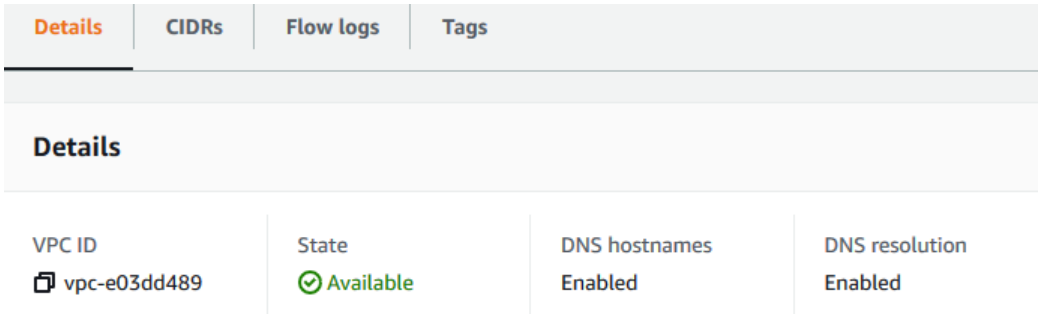
- describe-network-interfaces (AWS CLI)
- Get-EC2NetworkInterface (AWS Tools for Windows PowerShell)

VPC の DNS 属性の表示と更新

Amazon VPC コンソールを使用して、VPC の DNS サポート属性を表示および更新することができます。

コンソールを使用して VPC の DNS サポートの詳細を確認し更新するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC のチェックボックスをオンにします。
4. 情報の詳細を確認します。この例では、両方の DNS hostnames (DNS ホスト名) およびDNS resolution (DNS 解決方法) が有効です。



5. これらの設定を更新するには、[Actions] (アクション) を選択し、続いて [Edit DNS hostnames] (DNS ホスト名の編集) または [Edit DNS resolution] （DNS 解決）を選択します。プロンプトが表示されたら、[Enable] (有効化) を選択または未選択にして、[Save changes] (変更を保存) を選択します。

コマンドラインを使用して VPC の DNS サポートについて説明するには

次のいずれかのコマンドを使用できます。これらのコマンドラインインターフェイスの詳細については、Amazon VPC にアクセスする (p. 1) を参照してください。

- describe-vpc-attribute（AWS CLI）
- Get-EC2VpcAttribute（AWS Tools for Windows PowerShell）

コマンドラインを使用して VPC の DNS サポートを更新するには

次のいずれかのコマンドを使用できます。これらのコマンドラインインターフェイスの詳細については、Amazon VPC にアクセスする (p. 1) を参照してください。

- modify-vpc-attribute（AWS CLI）
- Edit-EC2VpcAttribute（AWS Tools for Windows PowerShell）

プライベートホストゾーン

プライベート IPv4 アドレスや AWS で提供されたプライベート DNS ホスト名の代わりに `example.com` のようなカスタム DNS ドメイン名を使用して VPC のリソースにアクセスする場合は、Route 53 でプライベートホストゾーンを作成できます。プライベートホストゾーンは、インターネットにリソースを公開することなく、1 つ以上の VPC 内のドメインとそのサブドメインにトラフィックをルーティングする方法に関する情報を保持するコンテナです。次に、Route 53 リソースレコードセットを作成できま

す。これにより、ドメインとサブドメインへのクエリに Route 53 が対応する方法が決定されます。例えば、example.com のブラウザリクエストが VPC のウェブサーバーにルーティングされるようにする場合、プライベートホストゾーンで A レコードを作成し、そのウェブサーバーの IP アドレスを指定します。プライベートホストゾーンの作成の詳細については、Amazon Route 53 開発者ガイドの「プライベートホストゾーンの使用」を参照してください。

カスタム DNS ドメイン名を使用してリソースにアクセスするには、VPC 内のインスタンスに接続している必要があります。インスタンスで、`ping` コマンド (`ping mywebserver.example.com` など) を使用してカスタム DNS 名からプライベートホストゾーンのリソースにアクセス可能なことをテストできます (`ping` コマンドが機能するには、インスタンスのセキュリティグループのルールでインバウンド ICMP トラフィックが許可されている必要があります)。

ClassicLink DNS サポートに対して VPC が有効になっていれば、ClassicLink を使用して VPC にリンクされた EC2-Classic インスタンスからプライベートホストゾーンにアクセスできます。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「ClassicLink DNS サポートを有効にする」を参照してください。それ以外の場合、プライベートホストゾーンは VPC 外部の推移的関係をサポートしていません。例えば、VPN 接続の他方の側からカスタムプライベート DNS 名を使用してリソースにアクセスすることはできません。詳細については、Linux インスタンス用の Amazon EC2 ユーザーガイドの「ClassicLink の制限」を参照してください。

> **Important**
>
> Amazon Route 53 のプライベートホストゾーンに定義されているカスタム DNS ドメイン名を使用している場合は、`enableDnsHostnames` 属性と `enableDnsSupport` 属性を `true` に設定する必要があります。

プレフィックスリスト

プレフィックスリストは、1 つ以上の CIDR ブロックのセットです。プレフィクスリストを使用すると、セキュリティグループとルートテーブルの設定と管理が容易になります。頻繁に使用する IP アドレスからプレフィクスリストを作成し、それらを個別に参照するのではなく、セキュリティグループのルールおよびルートでセットとして参照できます。例えば、CIDR ブロックは異なるが同じポートとプロトコルを持つセキュリティグループルールを、プレフィクスリストを使用する 1 つのルールに統合できます。ネットワークを拡張し、別の CIDR ブロックからのトラフィックを許可する必要がある場合は、関連するプレフィクスリストを更新し、プレフィクスリストを使用するすべてのセキュリティグループを更新します。

プレフィックスリストには、次の 2 つのタイプがあります。

- カスタマー管理プレフィクスリスト ：定義および管理する IP アドレス範囲のセット。プレフィックスリストは、他の AWS アカウントと共有できます。そのアカウントはそのリソース内で、このプレフィックスリストを参照できます。
- AWS マネージドプレフィクスリスト — AWS サービスの IP アドレス範囲のセット。AWS マネージドプレフィックスリストを作成、変更、共有、削除することはできません。

目次

プレフィックスリストの概念とルール

プレフィックスリストはエントリで構成されます。各エントリは、CIDR ブロックで構成されます。オプションで CIDR ブロックの説明も含まれます。

カスタマーマネージドプレフィックスリスト

カスタマーマネージドプレフィックスリストには、次のルールが適用されます。

- 1 つのプレフィックスリスト内では、単一タイプの IP アドレス指定 (IPv4 または IPv6) のみがサポートされます。IPv4 および IPv6 の CIDR ブロックを 1 つのプレフィックスリスト内で組み合わせることはできません。
- プレフィックスリストは、それを作成したリージョンにのみ適用されます。
- プレフィックスリストを作成するときは、プレフィックスリストがサポートできるエントリの最大数を指定する必要があります。
- リソース内でプレフィックスリストを参照する場合、プレフィックスリストのエントリの最大数は、リソースのエントリの数のクォータに対してカウントされます。例えば、エントリ数が 20 個のプレフィックスリストを作成し、セキュリティグループルール内でそのプレフィックスリストを参照する場合、セキュリティグループの 20 個のルールとしてカウントされます。
- ルートテーブル内でプレフィックスリストを参照する場合、ルート優先度ルールが適用されます。詳細については、「ルーティング優先度とプレフィックスリスト (p. 302)」を参照してください。
- プレフィックスリストを変更できます。プレフィックスリストのエントリを追加または削除するたびに、新しいバージョンのプレフィックスリストが作成されます。リソースがプレフィックスを参照する場合は、常に現在 (最新) のバージョンが使用されます。以前のバージョンのプレフィックスリストからエントリを復元できます。また、新しいバージョンも作成されます。
- プレフィックスリストに関連するクォータがあります。詳細については、「カスタマーマネージドプレフィックスリスト (p. 363)」を参照してください。

AWS マネージドプレフィックスリスト

AWS マネージドプレフィックスリストには、以下のルールが適用されます。

- AWS マネージドプレフィックスリストを作成、変更、共有、削除することはできません。
- リソース内で AWS マネージドプレフィックスリストを参照する場合、このリストはリソースのルールまたはエントリの 1 つとしてカウントされます。
- AWS マネージドプレフィックスリストのバージョン番号を表示することはできません。

プレフィックスリストの Identity and Access Management

デフォルトでは、IAM ユーザーには、プレフィックスリストを作成、表示、変更、または削除するためのアクセス許可はありません。ユーザーにプレフィックスリストの操作を許可する IAM ポリシーを作成することができます。

Amazon VPC アクションのリストと、IAM ポリシーで使用できるリソースと条件キーについては、IAM ユーザーガイドの「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。

次のポリシー例では、ユーザーに、プレフィックスリスト `pl-123456abcde123456` の表示と操作のみを許可しています。ユーザーがプレフィックスリストの作成または削除を行うことはできません。

```
{
   "Version": "2012-10-17",
   "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:GetManagedPrefixListAssociations",
        "ec2:GetManagedPrefixListEntries",
        "ec2:ModifyManagedPrefixList",
        "ec2:RestoreManagedPrefixListVersion"
```

```
      ],
      "Resource": "arn:aws:ec2:region:account:prefix-list/pl-123456abcde123456"
    },
    {
      "Effect": "Allow",
      "Action": "ec2:DescribeManagedPrefixLists",
      "Resource": "*"
    }
  ]
}
```

Amazon VPC での IAM の操作方法については、「Amazon VPC の Identity and Access Management (p. 169)」を参照してください。

カスタマーマネージドプレフィックスリストの操作

カスタマー管理のプレフィックスリストを作成して管理できます。AWS管理のレフィックスリストを表示できます。

タスク

- プレフィックスリストを作成する (p. 278)
- プレフィックスリストを表示する (p. 279)
- プレフィックスリストのエントリの表示 (p. 279)
- プレフィックスリストの関連付け (参照) の表示 (p. 279)
- プレフィックスリストの変更 (p. 280)
- プレフィックスリストの以前のバージョンを復元する (p. 280)
- プレフィックスリストを削除する (p. 281)
- AWS リソース内のプレフィックスリストの参照 (p. 281)

プレフィックスリストを作成する

プレフィックスリストを作成するときは、プレフィックスリストがサポートできるエントリの最大数を指定する必要があります。

Limitation

ルールの数とプレフィックスリストの最大エントリ数が、アカウントのセキュリティグループごとのルールのクォータを超える場合、プレフィックスリストをセキュリティグループルールに追加することはできません。

コンソールを使用してプレフィックスリストを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[マネージドプレフィックスリスト] を選択します。
3. [プレフィックスリストを作成] を選択します。
4. [プレフィックスリスト名] に、プレフィックスリストの名前を入力します。
5. [最大エントリ] に、プレフィックスリストの最大エントリ数を入力します。
6. [アドレスファミリー] で、プレフィックスリストでサポートするエントリのタイプとして IPv4 または IPv6 を選択します。
7. [プレフィックスリストのエントリ] で、[新しいエントリを追加] を選択し、エントリの CIDR ブロックと説明を入力します。各エントリに対してこのステップを実行します。
8. (オプション) [タグ] では、後で識別するためのタグをプレフィックスリストに追加します。
9. [プレフィックスリストを作成] を選択します。

AWS CLI を使用してプレフィックスリストを作成するには

create-managed-prefix-list コマンドを使用します。

プレフィックスリストを表示する

プレフィックスリスト、共有されているプレフィックスリスト、および AWS 管理のプレフィックスリストを表示できます。

コンソールを使用してプレフィックスリストを表示するには

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、[マネージドプレフィックスリスト] を選択します。
3. [所有者 ID] 列には、プレフィックスリストの所有者の AWS アカウント ID が表示されます。AWS マネージドプレフィックスリストの場合、[所有者 ID] は AWS です。

AWS CLI を使用してプレフィックスリストを表示するには

describe-managed-prefix-lists コマンドを使用します。

プレフィックスリストのエントリの表示

プレフィックスリスト、共有されているプレフィックスリスト、および AWS 管理のプレフィックスリストのエントリを表示できます。

コンソールを使用してプレフィックスリストのエントリを表示するには

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、[マネージドプレフィックスリスト] を選択します。
3. プレフィックスリストのチェックボックスをオンにします。
4. 下部のペインで [エントリ] を選択して、プレフィックスリストのエントリを表示します。

AWS CLI を使用してプレフィックスリストのエントリを表示するには

get-managed-prefix-list-entries コマンドを使用します。

プレフィックスリストの関連付け (参照) の表示

プレフィックスリストに関連付けられたリソースの ID と所有者を表示することができます。関連付けられたリソースとは、エントリまたはルール内でお客様のプレフィックスリストを参照しているリソースです。

Limitation

AWS マネージドプレフィックスリストに関連付けられたリソースを表示することはできません。

コンソールを使用してプレフィックスリストの関連付けを表示するには

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、[マネージドプレフィックスリスト] を選択します。
3. プレフィックスリストのチェックボックスをオンにします。
4. 下部のペインで [関連付け] を選択して、プレフィックスリストを参照しているリソースを表示します。

AWS CLI を使用してプレフィックスリストの関連付けを表示するには

get-managed-prefix-list-associations コマンドを使用します。

プレフィックスリストの変更

お客様のプレフィックスリストについては、名前を変更することも、エントリを追加または削除することもできます。AWS Management Console を使ってエントリの最大数を後で変更することはできません。エントリの最大数を更新するには、AWS CLI または AWS SDK を使用します。

プレフィックスリストのエントリを更新すると、新しいバージョンのプレフィックスリストが作成されます。プレフィックスリストの名前を更新、あるいはプレフィックスリストのエントリ最大数を更新しても、新しいバージョンのプレフィックスリストが作成されません。

Considerations

- AWS 管理プレフィックスリストは変更できません。
- プレフィクスリスト内のエントリ最大数を増やすと、増加した最大サイズがプレフィクスリストを参照するリソースのエントリのクォータに適用されます。これらのリソースのすべてが増加した最大サイズをサポートできない場合、変更操作は失敗し、以前の最大サイズに戻されます。

コンソールを使用してプレフィックスリストを変更するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[マネージドプレフィックスリスト] を選択します。
3. プレフィックスリストのチェックボックスを選択し、[Actions] (アクション) 、[Modify prefix list] (プレフィックスリストを変更) の順に選択します。
4. [プレフィックスリスト名] に、プレフィックスリストの新しい名前を入力します。
5. [プレフィックスリストのエントリ] で、既存のエントリを削除するには [削除] を選択します。新しいエントリを追加するには、[新しいエントリを追加] を選択し、エントリの CIDR ブロックと説明を入力します。
6. [プレフィックスリストを保存] を選択します。

AWS CLI を使用してプレフィックスリストを変更するには

modify-managed-prefix-list コマンドを使用します。

プレフィックスリストの以前のバージョンを復元する

お客様の以前のバージョンのプレフィックスリストのエントリを新しいバージョンに復元できます。これにより、プレフィックスリストの新しいバージョンが作成されます。

プレフィクスリストのサイズを小さくした場合は、プレフィクスリストが、前のバージョンのエントリを格納するのに十分なサイズであるかをを確認する必要があります。

コンソールを使用して以前のバージョンのプレフィックスリストを復元するには

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、[マネージドプレフィックスリスト] を選択します。
3. プレフィックスリストのチェックボックスを選択し、[Actions] (アクション)、[Restore prefix list] (プレフィックスリストを復元) の順に選択します。
4. [Select prefix list version] (プレフィックスリストのバージョンを選択) で、以前のバージョンを選択します。選択したバージョンのエントリが [Prefix list entries] (プレフィックスリストエントリ) に表示されます。
5. [プレフィックスリストを復元] を選択します。

AWS CLI を使用して以前のバージョンのプレフィックスリストを復元するには

restore-managed-prefix-list-version コマンドを使用します。

プレフィックスリストを削除する

プレフィックスリストを削除するには、まずリソース内 (ルートテーブル内など) で、そのプレフィックスリストへの参照をすべて削除する必要があります。AWS RAM を使用してプレフィックスリストを共有している場合は、コンシューマーが所有するリソース内の参照を先に削除する必要があります。

Limitation

AWS マネージドプレフィックスリストは削除できません。

コンソールを使用してプレフィックスリストを削除するには

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、[マネージドプレフィックスリスト] を選択します。
3. プレフィックスリストを選択し、[アクション]、[プレフィックスリストを削除] の順に選択します。
4. 確認ダイアログボックスで、`delete` と入力し、[削除] を選択します。

AWS CLI を使用してプレフィックスリストを削除するには

delete-managed-prefix-list コマンドを使用します。

AWS リソース内のプレフィックスリストの参照

以下の AWS リソースでプレフィックスリストを参照できます。

リソース

- VPC セキュリティグループ (p. 281)
- サブネットルートテーブル (p. 282)
- トランジットゲートウェイルートテーブル (p. 282)

VPC セキュリティグループ

プレフィックスリストは、インバウンドルールの送信元またはアウトバウンドルールの送信先として指定できます。セキュリティグループの詳細については、VPC のセキュリティグループ (p. 188) を参照してください。

コンソールを使用してセキュリティグループルール内でプレフィックスリストを参照するには

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] を選択します。
3. 更新するセキュリティグループを選択します。
4. [アクション]、[Edit inbound rules (インバウンドルールを編集)] を選択するか、[アクション]、[Edit outbound rules (アウトバウンドルールを編集)] を選択します。
5. [Add rule] を選択します。[タイプ] で、トラフィックタイプを選択します。[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、プレフィックスリストの ID を選択します。
6. [Save Rules (ルールの保存)] を選択します。

AWS CLI を使用してセキュリティグループルール内でプレフィックスリストを参照するには

authorize-security-group-ingress コマンドおよび authorize-security-group-egress コマンドを使用します。`--ip-permissions` パラメータには、`PrefixListIds` を使用してプレフィックスリストの ID を指定します。

サブネットルートテーブル

ルートテーブルエントリの送信先としてプレフィックスリストを指定できます。ゲートウェイルートテーブル内でプレフィックスリストを参照することはできません。ルートテーブルの詳細については、「VPC のルートテーブル (p. 294)」を参照してください。

コンソールを使用してルートテーブル内でプレフィックスリストを参照するには

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 [ルートテーブル] を選択して、ルートテーブルを選択します。
3. [アクション]、[ポリシーの編集] の順に選択します。
4. ルートを追加するには、[ルートの追加] を選択します。
5. [送信先] に、プレフィックスリストの ID を入力します。
6. [ターゲット] で、ターゲットを選択します。
7. [Save changes] を選択します。

AWS CLI を使用してルートテーブル内でプレフィックスリストを参照するには

create-route (AWS CLI) コマンドを使用します。`--destination-prefix-list-id` パラメータを使用して、プレフィックスリストの ID を指定します。

トランジットゲートウェイルートテーブル

ルートの送信先としてプレフィックスリストを指定できます。詳細については、Amazon VPC トランジットゲートウェイの「プレフィックスリストの参照」を参照してください。

共有プレフィックスリストの操作

カスタマーマネージドプレフィックスリストは、AWS Resource Access Manager (AWS RAM) と統合できます。AWS RAM を使用すると、リソース共有を作成することで、AWS アカウント全体で所有するリソースを共有できます。リソース共有では、共有対象のリソースと、共有先となるコンシューマーを指定します。コンシューマーには、個人の AWS アカウントや、AWS Organizations 内の組織単位または組織全体を指定できます。

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

プレフィックスリストの所有者は、プレフィックスリストを次の対象と共有できます。

- AWS の組織内または組織外の特定の AWS Organizations アカウント
- の組織内の組織単位AWS Organizations
- の組織全体AWS Organizations

プレフィックスリストの共有先であるコンシューマーは、プレフィックスリストとそのエントリを表示でき、そのプレフィックスリストを AWS リソース内で参照できます。

目次

プレフィックスリストを共有するための前提条件

- プレフィックスリストを共有するには、それを AWS アカウント内で所有している必要があります。自身が共有を受けているプレフィックスリストは共有できません。AWS マネージドプレフィックスリストを共有することはできません。
- AWS Organizations の組織や組織単位とプレフィックスリストを共有するには、AWS Organizations との共有を有効にする必要があります。詳細については、AWS RAM ユーザーガイドの「AWS Organizations で共有を有効化する」を参照してください。

プレフィックスリストを共有する

プレフィックスリストを共有するには、そのプレフィックスリストをリソース共有に追加する必要があります。リソース共有がない場合は、まず AWS RAM コンソールを使用してリソース共有を作成する必要があります。

AWS Organizations の組織に属している場合、組織内での共有が有効になっていると、組織内のコンシューマーには共有プレフィックスリストへのアクセス許可が自動的に付与されます。それ以外の場合、コンシューマーはリソース共有への参加の招待を受け取り、その招待を受け入れた後で、共有プレフィックスリストへのアクセス許可が付与されます。

AWS RAM コンソールまたは AWS CLI を使用してリソース共有を作成し、自己所有のプレフィックスリストを共有できます。

AWS RAM コンソールを使用してリソース共有を作成し、プレフィックスリストを共有するには

AWS RAM ユーザーガイドの「リソース共有を作成する」の手順に従います。[リソースタイプを選択] で、[プレフィックスリスト] を選択し、プレフィックスリストのチェックボックスをオンにします。

AWS RAM コンソールを使用して既存のリソース共有にプレフィックスリストを追加するには

所有するマネージドプレフィックスリストを既存のリソース共有に追加するには、AWS RAM ユーザーガイドの「リソース共有の更新」のステップに従います。[リソースタイプを選択] で、[プレフィックスリスト] を選択し、プレフィックスリストのチェックボックスをオンにします。

AWS CLI を使用して自己所有のプレフィックスリストを共有するには

リソース共有を作成および更新するには、以下のコマンドを使用します。

- create-resource-share
- associate-resource-share
- update-resource-share

共有プレフィックスリストの特定

所有者とコンシューマーは、Amazon VPC コンソールまたは AWS CLI を使用して、共有プレフィックスリストを特定できます。

Amazon VPC コンソールを使用して共有プレフィックスリストを特定するには

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、[マネージドプレフィックスリスト] を選択します。
3. このページには、自己所有のプレフィックスリストと、共有を受けているプレフィックスリストが表示されます。[所有者 ID] 列には、プレフィックスリストの所有者の AWS アカウント ID が表示されます。
4. プレフィックスリストのリソース共有情報を表示するには、プレフィックスリストを選択し、下部のペインで [共有] を選択します。

AWS CLI を使用して共有プレフィックスリストを特定するには

describe-managed-prefix-lists コマンドを使用します。このコマンドでは、自己所有のプレフィックスリストおよび共有を受けているプレフィックスリストが返されます。`OwnerId` は、プレフィックスリストの所有者の AWS アカウント ID を示します。

共有プレフィックスリストへの参照の特定

所有者は、 共有プレフィックスリストを参照しているコンシューマ所有のリソースを特定できます。

Amazon VPC コンソールを使用して共有プレフィックスリストへの参照を特定するには

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、[マネージドプレフィックスリスト] を選択します。
3. プレフィックスリストを選択し、下部のペインで [関連付け] を選択します。
4. プレフィックスリストを参照しているリソースの ID が、[リソース ID] 列に表示されます。リソースの所有者は、[リソース所有者] 列に表示されます。

AWS CLI を使用して共有プレフィックスリストへの参照を特定するには

get-managed-prefix-list-associations コマンドを使用します。

共有プレフィックスリストの共有解除

プレフィックスリストの共有を解除すると、コンシューマーはアカウント内でプレフィックスリストまたはそのエントリを表示できず、リソース内でプレフィックスリストを参照することもできなくなります。プレフィックスリストがコンシューマーのリソース内ですでに参照されている場合、参照の動作は維持され、引き続きその参照を表示できます (p. 284)。プレフィックスリストを新しいバージョンに更新すると、参照では最新のバージョンが使用されます。

自己所有の共有プレフィックスリストを共有解除するには、AWS RAM を使用してリソース共有から削除する必要があります。

AWS RAM コンソールを使用して、自己所有の共有プレフィックスリストを共有解除するには

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

AWS CLI を使用して、自己所有の共有プレフィックスリストを共有解除するには

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

共有プレフィックスリストのアクセス許可

所有者のアクセス許可

所有者は、共有プレフィックスリストとそのエントリを管理する必要があります。所有者は、プレフィックスリストを参照する AWS リソースの ID を表示できます。ただし、コンシューマーが所有する AWS リソース内でプレフィックスリストへの参照を追加および削除することはできません。

コンシューマーが所有するリソース内でプレフィックスリストが参照されている場合、所有者はプレフィックスリストを削除できません。

コンシューマーのアクセス許可

コンシューマーは共有プレフィックスリストのエントリを表示でき、AWS リソース内で共有プレフィックスリストを参照できます。ただし、共有プレフィックスリストを変更、復元、または削除することはできません。

請求と使用量測定

プレフィックスリストの共有に追加料金はかかりません。

Quotas

AWS RAM に関連するクォータ (制限) の詳細については、AWS RAM ユーザーガイドの「サービスの制限」を参照してください。

Amazon EC2 ネットワーキングコンポーネント

VPC のネットワーキングを設定するには、次の Amazon EC2 ネットワーキングコンポーネントを使用します。

コンポーネント

Elastic Network Interface

Elastic Network Interface (このドキュメントではネットワークインターフェイスと呼びます) は、仮想ネットワークカードを表す VPC 内の論理ネットワーキングコンポーネントです。次の属性を含めることができます。

- プライマリプライベート IPv4 アドレス
- 1 つ以上のセカンダリプライベート IPv4 アドレス
- プライベート IPv4 アドレスごとに 1 つの Elastic IP アドレス
- インスタンス起動時に eth0 のネットワークインターフェイスに自動割り当て可能な 1 つのパブリック IPv4 アドレス
- 1 つ以上の IPv6 アドレス
- 1 つ以上のセキュリティグループ
- MAC アドレス
- 送信元/送信先チェックフラグ
- 説明

ネットワークインターフェイスを作成して、同じアベイラビリティーゾーン内のインスタンスに接続できます。インスタンスにアタッチまたはデタッチしたり、別のインスタンスに再アタッチしたりするときには、ネットワークインターフェイスとしての属性が保持されます。、インスタンス間でネットワークインターフェイスを移動すると、ネットワークトラフィックは新しいインスタンスにリダイレクトされます。

ネットワークインターフェイスの詳細および Amazon EC2 コンソールを使用してネットワークインターフェイスを操作する手順については、Linux インスタンス用の Amazon EC2 ユーザーガイドの「Elastic Network Interface」を参照してください。

アカウントでは、AWS のサービスで作成および管理されるリクエスタマネージド型のネットワークインターフェイスも使用できます。これらを通じて他のリソースやサービスを利用できます。これらは、ユーザーが直接管理できないネットワークインターフェイスです。詳細については、「Linux インスタンス用 Amazon EC2 ユーザーガイド」の「リクエスタマネージド型のネットワークインターフェイス」を参照してください。

サブネット CIDR 予約

サブネット CIDR 予約は、サブネット内の IPv4 アドレスまたは IPv6 アドレスの範囲で行われます。予約を作成するときに、予約範囲の使用方法を指定します。以下のオプションが利用できます。

- プレフィックス — インスタンスに関連付けられたネットワークインターフェイスに IP アドレスを割り当てることができます。詳細は、「Linux インスタンス用 Amazon EC2 ユーザーガイド」の「Amazon EC2 ネットワークインターフェイスへのプレフィックスの割り当て」を参照してください。
- 明示的 — AWS は IP アドレスを使用しません。サブネット内に存在するリソースに IP アドレスを手動で割り当てます。

サブネット CIDR 予約には、次のルールが適用されます。

- サブネットごとに複数の CIDR 範囲を予約できます。各範囲の予約タイプは、同じタイプ (例: プレフィックス) でも、異なるタイプ (例: プレフィックスおよび明示的) でも構いません。
- 同じ VPC 内で複数の CIDR 範囲を予約する場合、CIDR 範囲は重複できません。
- サブネット内でプレフィックス委任用に複数の範囲を予約し、プレフィックス委任が自動割り当て用に設定されている場合、ネットワークインターフェイスに割り当てる IP アドレスをランダムに選択します。
- 予約を削除しても、リソースに割り当てられている IP アドレスは変更されません。すでに使用されていない IP アドレスのみが使用可能になります。

コンソールを使用してサブネット CIDR 予約を操作する

以下のように、サブネット CIDR 予約を作成および管理できます。

サブネット CIDR 予約を編集するには

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、[Subnets] を選択します。
3. サブネットを選択します。
4. [Actions]、[Edit CIDR reservations] を選択し、以下の操作を行います。
   - IPv4 CIDR 予約を追加するには、[IPv4]、[Add IPv4 CIDR reservations] を選択します。予約タイプを選択し、CIDR 範囲を入力し、[Add] をクリックします。
   - IPv6 CIDR 予約を追加するには、[IPv6]、[Add IPv6 CIDR reservations] を選択します。予約タイプを選択し、CIDR 範囲を入力し、[Add] をクリックします。
   - CIDR 予約を削除するには、エントリーの後ろの [Remove] をクリックします。

AWS CLI を使用してサブネット CIDR 予約を操作する

AWS CLI を使って、サブネット CIDR 予約を作成および管理できます。

タスク

- サブネット CIDR 予約の作成 (p. 288)
- サブネット CIDR 予約の表示 (p. 288)
- サブネット CIDR 予約の削除 (p. 288)

サブネット CIDR 予約の作成

create-subnet-cidr-reservation を使って、サブネット CIDR 予約を作成できます。

```
aws ec2 create-subnet-cidr-reservation --subnet-id subnet-03c51e2eEXAMPLE --reservation-
type prefix --cidr 2600:1f13:925:d240:3a1b::/80
```

以下は出力例です。

```
{
    "SubnetCidrReservation": {
        "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
        "SubnetId": "subnet-03c51e2ef5EXAMPLE",
        "Cidr": "2600:1f13:925:d240:3a1b::/80",
        "ReservationType": "prefix",
        "OwnerId": "123456789012"
    }
}
```

サブネット CIDR 予約の表示

get-subnet-cidr-reservations を使って、サブネット CIDR 予約の詳細を表示できます。

```
aws ec2 get-subnet-cidr-reservations --subnet-id subnet-05eef9fb78EXAMPLE
```

サブネット CIDR 予約の削除

delete-subnet-cidr-reservation を使って、サブネット CIDR 予約を削除できます。

```
aws ec2 delete-subnet-cidr-reservation --subnet-cidr-reservation-id scr-044f977c4eEXAMPLE
```

Elastic IP アドレス

Elastic IP アドレスは、動的なクラウドコンピューティング向けに設計された静的なパブリック IPv4 アドレスです。Elastic IP アドレスは、アカウントのすべての VPC の任意のインスタンスまたはネットワークインターフェイスに関連付けることができます。Elastic IP アドレス を使用すると、インスタンスに障害が発生しても、そのアドレスを VPC 内の別のインスタンスにすばやく再マッピングすることで、インスタンスの障害を隠すことができます。

Elastic IP アドレスの概念とルール

Elastic IP アドレスを使用するには、まずアカウントで使用するために割り当てます。次に、VPC のインスタンスまたはネットワークインターフェイスに関連付けることができます。elastic IP アドレスは、明示的に解放するまで AWS アカウントに割り当てられたままです。

Elastic IP アドレスはネットワークインターフェイスのプロパティの 1 つです。Elastic IP アドレス をインスタンスに割り当てるには、そのインスタンスにアタッチされているネットワークインターフェイスを更新します。Elastic IP アドレスを直接インスタンスに関連付けずにネットワークインターフェイスに関連付ける利点は、1 つのステップでネットワークインターフェイスの全属性を 1 つのインスタンスから別のインスタンスに移動できることです。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「Elastic Network Interface」をご参照ください。

以下のルールが適用されます。

- Elastic IP アドレスは、一度に 1 つのインスタンスまたはネットワークインターフェイスに関連付けることができます。
- Elastic IP アドレスは、あるインスタンスまたはネットワークインターフェイスから別のインスタンスまたはネットワークインターフェイスに移動できます。
- Elastic IP アドレス をインスタンスの eth0 ネットワークインターフェイスに関連付けると、現在のパブリック IPv4 アドレス (割り当てられている場合) は EC2-VPC パブリック IP アドレスプールに解放されます。Elastic IP アドレスの関連付けを解除すると、数分以内に新しいパブリック IPv4 アドレスが自動的に eth0 ネットワークインターフェイスに割り当てられます。2 番目のネットワークインターフェイスをインスタンスにアタッチした場合、これは適用されません。
- Elastic IP アドレスを効率的に使用するため、Elastic IP アドレスが実行中のインスタンスに関連付けられていない場合や、停止しているインスタンスやアタッチされていないネットワークインターフェイスに関連付けられている場合は、時間単位で小額の料金が請求されます。インスタンスを実行しているときは、インスタンスに関連付けられた 1 つの Elastic IP アドレスに対して料金は発生しませんが、インスタンスに関連付けられた追加の Elastic IP アドレスがある場合、その追加分に対しては料金が発生します。詳細については、「Amazon EC2 の料金表」を参照してください。
- Elastic IP アドレスは 5 つに制限されています。これらを節約するために、NAT デバイスを使用できます。詳細については、「」を参照してくださいVPC の NAT デバイス (p. 236)
- IPv6 の Elastic IP アドレスはサポートされていません。
- VPC 用に割り当てられた Elastic IP アドレスにタグを適用することはできますが、コスト配分タグはサポートされていません。Elastic IP アドレスを復旧する場合、タグは復旧されません。
- セキュリティグループとネットワーク ACL が送信元 IP アドレスからのトラフィックを許可している場合、インターネットから Elastic IP アドレスにアクセスできます。VPC 内からインターネットに戻る応答トラフィックには、インターネットゲートウェイが必要です。詳細については、「the section called “セキュリティグループ” (p. 188)」および「the section called “ネットワーク ACL” (p. 200)」を参照してください。
- Elastic IP アドレスには、次のいずれかのオプションを使用できます。
  - Amazon に Elastic IP アドレスを提供してもらいます。このオプションを選択すると、Elastic IP アドレスをネットワーク境界グループに関連付けることができます。これは、CIDR ブロックをアドバタイズする場所です。ネットワーク境界グループを設定すると、CIDR ブロックがこのグループに制限されます。
  - 自分の IP アドレスを使用します。独自の IP アドレスの取得については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「自分の IP アドレス (BYOIP) を使用する」を参照してください。

VPC で使用する Elastic IP アドレスと、EC2-Classic で使用する Elastic IP アドレスには違いがあります。詳細については、Linux インスタンス用の Amazon EC2 ユーザーガイドの「EC2-Classic と VPC の違い」を参照してください。EC2-Classic プラットフォームで使用するために割り当てた Elastic IP アドレスを VPC プラットフォームに移行できます。詳細については、「EC2-Classic からの Elastic IP アドレスの移行」を参照してください。

Elastic IP アドレスはリージョン固有のものです。Global Accelerator を使用してグローバル IP アドレスをプロビジョニングする方法の詳細については、 AWS Global Accelerator デベロッパーガイドの「リージョン固有の静的 IP アドレスの代わりにグローバル静的 IP アドレスを使用する」をご参照ください。

Elastic IP アドレスの操作

以下のセクションでは、Elastic IP アドレスの使用方法について説明します。

タスク

- Elastic IP アドレスを割り当てる (p. 290)
- Elastic IP アドレスの関連付け (p. 290)
- Elastic IP アドレスの表示 (p. 291)
- Elastic IP アドレスにタグを適用する (p. 291)

Elastic IP アドレスを割り当てる

Elastic IP を使用する前に、VPC で使用するために Elastic IP を割り当てる必要があります。

コンソールを使用して Elastic IP アドレスを割り当てるには

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 [Elastic IP] を選択します。
3. [Allocate Elastic IP address] を選択します。
4. [Public IPv4 address pool (パブリック IPv4 アドレスプール)] で、以下のいずれかを選択します。
   - [Amazon の IP アドレスプール] — Amazon の IP アドレスプールから IPv4 アドレスを割り当てる場合。
   - [パブリック IPv4 アドレスのプール] - AWS アカウントに持ち込んだ IP アドレスプールから IPv4 アドレスを割り当てる場合。IP アドレスプールがない場合、このオプションは無効になります。
   - 顧客所有の IPv4 アドレスのプール—Outpost で使用するために、オンプレミスネットワークから作成されたプールから IPv4 アドレスを割り当てる場合。Outpost がない場合、このオプションは使用できません。
5. (オプション) タグを追加または削除します。

   [タグの追加] [新しいタグの追加] を選択して、以下を実行します。

   - [キー] にはキー名を入力します。
   - [値] にキー値を入力します。

   [タグを削除] タグのキーと値の右側にある [削除] を選択します。
6. [Allocate] を選択します。

   Note

   アカウントが EC2-Classic をサポートしている場合には、まず [VPC] を選択します。

コマンドラインを使用して Elastic IP アドレスを割り当てるには

- allocate-address (AWS CLI)
- New-EC2Address (AWS Tools for Windows PowerShell)

Elastic IP アドレスの関連付け

Elastic IP を VPC で実行中のインスタンスまたはネットワークインターフェイスに関連付けることができます。

Elastic IP アドレスをインスタンスに関連付けると、インスタンスは、DNS ホスト名が有効な場合は DNS ホスト名を受け取ります。詳細については、「」を参照してくださいVPC の DNS サポート (p. 271)

Elastic IP アドレスをインスタンスまたはネットワークインターフェイスに関連付けるには

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 [Elastic IP] を選択します。

3. VPC で使用するために割り当てられた Elastic IP アドレス ([Scope (スコープ)] 列に値 `vpc` が含まれています) を選択し、[Actions (アクション)]、[Associate Elastic IP address (Elastic IP アドレスの関連付け)] の順に選択します。
4. [Instance] または [Network interface] を選択してから、インスタンスまたはネットワークインターフェイス ID を選択します。Elastic IP アドレスに関連付けるプライベート IP アドレスを選択します。[Associate] を選択します。

Elastic IP アドレスをコマンドラインを使用してインスタンスまたはネットワークインターフェイスに関連付けるには

- associate-address (AWS CLI)
- Register-EC2Address (AWS Tools for Windows PowerShell)

Elastic IP アドレスの表示

アカウントに割り当てられている Elastic IP アドレスを表示できます。

Elastic IP アドレスを表示するには

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 [Elastic IP] を選択します。
3. 表示されたリストをフィルタリングするには、Elastic IP アドレスの一部またはその属性のいずれかを検索ボックスに入力します。

コマンドラインを使用して Elastic IP アドレスを表示するには

- describe-addresses (AWS CLI)
- Get-EC2Address (AWS Tools for Windows PowerShell)

Elastic IP アドレスにタグを適用する

Elastic IP アドレスにタグを適用し、組織のニーズに応じて識別または分類できます。

コンソールを使用して Elastic IP アドレスにタグを付けるには

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 [Elastic IP] を選択します。
3. Elastic IP アドレスを選択し、[Tags] を選択します。
4. [Manage tags (タグの管理)] を選択し、必要に応じてタグのキーと値を入力して、[Save (保存)] を選択します。

コマンドラインを使用して Elastic IP アドレスにタグを付けるには

- create-tags (AWS CLI)
- New-EC2Tag (AWS Tools for Windows PowerShell)

Elastic IP アドレスの関連付けを解除する

Elastic IP アドレスが関連付けられているリソースを変更するには、まず、現在関連付けられているリソースとの関連付けを解除する必要があります。

Elastic IP アドレスの関連付けを解除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Elastic IP] を選択します。
3. Elastic IP アドレスを選択してから、[Actions (アクション)]、[Elastic IP アドレスの関連付けの解除] の順に選択します。
4. プロンプトが表示されたら、[Disassociate (関連付けの解除)] を選択します。

コマンドラインを使用して Elastic IP アドレスを別のインスタンスに関連付けるには

- disassociate-address (AWS CLI)
- Unregister-EC2Address (AWS Tools for Windows PowerShell)

Elastic IP アドレスを解放する

Elastic IP アドレスが不要になった場合は、解放することをお勧めします。VPC で使用するために割り当てられているがインスタンスに関連付けられていない Elastic IP アドレス に対しては料金が発生します。Elastic IP アドレスは、インスタンスまたはネットワークインターフェイスに関連付けることはできません。

Elastic IP アドレスを解放するには

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 [Elastic IP] を選択します。
3. Elastic IP アドレスを選択してから、[Actions (アクション)]、[Release Elastic IP addresses (Elastic IP アドレスの解放)] の順に選択します。
4. プロンプトが表示されたら、[Release] を選択します。

コマンドラインを使用して Elastic IP アドレスを解放するには

- release-address (AWS CLI)
- Remove-EC2Address (AWS Tools for Windows PowerShell)

Elastic IP アドレスの復元

Elastic IPアドレスを解放した場合でも、復元できる場合があります。他の AWS アカウントに割り当てられている場合、または Elastic IP アドレスのクォータを超過する場合、Elastic IP アドレスを復元することはできません。

Elastic IP アドレスは、Amazon EC2 API またはコマンドラインツールで復元できます。

AWS CLI を使用して Elastic IP アドレスを復元するには

`--address` パラメータを使用した allocate-address コマンドを使用して、IP アドレスを指定します。

```
aws ec2 allocate-address --domain vpc --address 203.0.113.3
```

ClassicLink

ClassicLink を使用すると、EC2-Classic インスタンスを同じリージョンにあるお客様のアカウントの VPC にリンクできます。これによって、VPC のセキュリティグループを EC2-Classic インスタンスに関連付け

ることができ、プライベート IPv4 アドレスを使用して EC2-Classic インスタンスと VPC 内のインスタンスが通信できるようになります。ClassicLink により、パブリック IPv4 アドレスや Elastic IP アドレスを使用しなくても、これらのプラットフォーム内のインスタンス間で通信できます。プライベートおよびパブリック IPv4 アドレスについては、「VPC の IP アドレス指定 (p. 124)」を参照してください。

ClassicLink は、EC2-Classic プラットフォームをサポートするアカウントを持つすべてのユーザーが利用でき、任意のインスタンスタイプの EC2-Classic インスタンスで使用できます。

ClassicLink は追加料金なしで使用できます。データ転送とインスタンス時間の使用量に対する標準料金が適用されます。

ClassicLink とその使用方法の詳細については、Amazon EC2 ユーザーガイドの以下のトピックを参照してください。

- ClassicLink の基本
- ClassicLink の制限事項
- ClassicLink の操作
- ClassicLink API と CLI の概要

VPC のルートテーブル

ルートテーブルには、サブネットまたはゲートウェイからのネットワークトラフィックの経路を判断する際に使用される、ルートと呼ばれる一連のルールが含まれます。

目次

ルートテーブルの概念

ルートテーブルの主な概念は次のとおりです。

- メインルートテーブル — VPC に自動的に割り当てられるルートテーブル。これは、他のルートテーブルに明示的に関連付けられていないすべてのサブネットのルーティングを制御します。
- カスタムルートテーブル — VPC 用に作成するルートテーブル。
- エッジの関連付け — インバウンド VPC トラフィックをアプライアンスにルーティングするために使用するルートテーブル。ルートテーブルをインターネットゲートウェイまたは仮想プライベートゲートウェイに関連付け、アプライアンスのネットワークインターフェイスを VPC トラフィックのターゲットとして指定します。
- ルートテーブルの関連付け — ルートテーブルとサブネット、インターネットゲートウェイ、または仮想プライベートゲートウェイの間の関連付け。
- サブネットルートテーブル — サブネットに関連付けられたルートテーブル。
- ゲートウェイルートテーブル — インターネットゲートウェイまたは仮想プライベートゲートウェイに関連付けられたルートテーブル。
- ローカルゲートウェイルートテーブル — Outposts ローカルゲートウェイに関連付けられているルートテーブル。ローカルゲートウェイの詳細については、AWS Outposts ユーザーガイドの「ローカルゲートウェイ」を参照してください。
- [送信先] — トラフィックを送信する IP アドレスの範囲 (送信先 CIDR)。例えば、CIDR `172.16.0.0/12` がある外部企業ネットワークなどです。
- 伝達 — ルート伝達により、仮想プライベートゲートウェイはルートテーブルにルートを自動的に伝達できます。つまり、ルートテーブルへの VPN ルートを手動で入力する必要はありません。VPN ルーティングオプションの詳細については、Site-to-Site VPN ユーザーガイドの「Site-to-Site VPN ルーティングオプション」を参照してください。
- [ターゲット] — 送信先トラフィックの送信に使用するゲートウェイ、ネットワークインターフェイス、または接続 (インターネットゲートウェイなど)。
- ローカルルート — VPC 内の通信のデフォルトルート。

ルーティングオプションの例については、「the section called “ルーティングオプションの例” (p. 303)」を参照してください。

ルートテーブルの仕組み

VPC には暗黙的なルーターがあり、ルートテーブルを使用してネットワークトラフィックの送信先を制御します。VPC の各サブネットをルートテーブルに関連付ける必要があり、ルートテーブルはサブネットのルーティング (サブネットルートテーブル) を制御します。サブネットを特定のルートテーブルに明示的に関連付けることができます。それ以外の場合、サブネットはメインルートテーブルに暗黙的に関連付けられます。1 つのサブネットは同時に 1 つのルートテーブルにしか関連付けることはできませんが、複数のサブネットを同じサブネットルートテーブルに関連付けることはできます。

必要に応じて、ルートテーブルをインターネットゲートウェイまたは仮想プライベートゲートウェイ (ゲートウェイルートテーブル) に関連付けることができます。これにより、ゲートウェイを介して VPC に入るインバウンドトラフィックのルーティングルールを指定できます。詳細については、「」を参照してくださいゲートウェイルートテーブル (p. 299)

VPC ごとに作成できるルートテーブルの数にはクォータがあります。ルートテーブルごとに追加できるルート数にもクォータがあります。詳細については、「」を参照してくださいAmazon VPC クォータ (p. 362)

目次

Routes

テーブル内の各ルートは、送信先とターゲットを指定します。例えば、サブネットがインターネットゲートウェイ経由でインターネットにアクセスできるようにするには、サブネットルートテーブルに次のルートを追加します。ルートの送信先は `0.0.0.0/0` です。これは、すべての IPv4 アドレスを表します。ターゲットは、VPC にアタッチされているインターネットゲートウェイです。

送信先	ターゲット
0.0.0.0/0	*igw-id*

IPv4 と IPv6 の CIDR ブロックは、個別に処理されます。例えば、送信先が `0.0.0.0/0` の CIDR のルーティングの場合は、IPv6 アドレスが自動的に含まれることはありません。すべての IPv6 アドレスの送信先が `::/0` の CIDR のルートを作成する必要があります。

AWS リソース全体で同じ CIDR ブロックのセットを頻繁に参照する場合は、カスタマーマネージドプレフィックスリスト (p. 276)を作成して、それらをグループ化できます。その後、ルートテーブルエントリの送信先としてプレフィックスリストを指定できます。

各ルートテーブルには、VPC 内で通信を有効にするローカルルートが含まれます。このルートは、デフォルトですべてのルートテーブルに追加されます。VPC に複数の IPv4 CIDR ブロックがある場合、ルートテーブルには各 IPv4 CIDR ブロックのローカルルートが含まれます。IPv6 CIDR ブロックを VPC に関連付けた場合、ルートテーブルには IPv6 CIDR ブロックのローカルルートが含まれます。サブネットルートテーブルまたはメインルートテーブルでこれらのルートを変更または削除することはできません。

ローカルルートよりも具体的なルートを追加できます。送信先は、VPC 内のサブネットの IPv4 または IPv6 CIDR ブロック全体と一致する必要があります。

ルートテーブルに複数のルートがある場合、AWS では、トラフィックと一致する (最長プレフィックス一致) 最も具体的なルートを使用して、トラフィックをルーティングする方法を決定します。

ゲートウェイルートテーブル内のルートとローカルルートの詳細については、「ゲートウェイルートテーブル (p. 299)」を参照してください。

Example

以下の図では、VPC に IPv4 CIDR ブロックとIPv6 CIDR ブロックの両方があります。ルートテーブル:

- VPC (`2001:db8:1234:1a00::/56`) 内に留まる送信先の IPv6 トラフィックは、`Local` ルートによってカバーされ、VPC 内でルーティングされます。
- IPv4 ルートと IPv6 ルートに対して個別に適用されます。そのため、IPv6 トラフィックはすべて (VPC 内のトラフィックを除く) 、Egress-Only インターネットゲートウェイにルーティングされます。
- ピア接続を指す `172.31.0.0/16` IPv4 トラフィックのルートがあります。
- インターネットゲートウェイを指すすべての IPv4 トラフィック (`0.0.0.0/0`) のルートがあります。
- Egress-only インターネットゲートウェイを指すすべての IPv6 トラフィック (`::/0`) のルートがあります。

送信先	ターゲット
10.0.0.0/16	ローカル
2001:db8:1234:1a00::/56	ローカル
172.31.0.0/16	pcx-11223344556677889
0.0.0.0/0	igw-12345678901234567
::/0	eigw-aabbccddee1122334

メインルートテーブル

VPC を作成するときに、メインルートテーブルが自動的に割り当てられます。サブネットに明示的なルーティングテーブルが関連付けられていない場合、デフォルトではメインのルーティングテーブルが使用されます。Amazon VPC コンソールの [ルートテーブル] ページで、[メイン] 列の [はい] を探すことによって VPC のメインルートテーブルを表示できます。

デフォルトでは、デフォルト以外の VPC を作成すると、メインルートテーブルにはローカルルートのみが含まれます。コンソールで VPC ウィザードを使用して NAT ゲートウェイまたは仮想プライベートゲートウェイを持つデフォルト以外の VPC を作成すると、ウィザードによりそれらのゲートウェイのメインルートテーブルにルートが自動的に追加されます。

メインルートテーブルには、次のルールが適用されます。

- メインルートテーブルを削除することはできません。
- ゲートウェイルートテーブルをメインルートテーブルとして設定することはできません。
- メインルートテーブルは、カスタムサブネットルートテーブルに置き換えることができます。
- メインルートテーブルで、ルートを追加、削除、変更することができます。
- すでに暗黙的に関連付けられている場合でも、サブネットをメインルートテーブルに明示的に関連付けることができます。

  この作業は、メインルートテーブルにするテーブルを変更するときに行います。メインルートテーブルであるテーブルを変更する場合、これにより、新しい追加のサブネット、または他のルートテーブルに

明示的に関連付けられていないサブネットのデフォルトも変更されます。詳細については、「」を参照してくださいメインルートテーブルの置換 (p. 317)

カスタムルートテーブル

デフォルトでは、カスタムルートテーブルは空で、必要に応じてルートを追加します。コンソールで VPC ウィザードを使用してインターネットゲートウェイを持つ VPC を作成すると、ウィザードによってカスタムルートテーブルが作成され、インターネットゲートウェイにルートが追加されます。VPC を保護する 1 つの方法は、メインルートテーブルを元のデフォルトの状態のままにすることです。次に、作成するそれぞれの新しいサブネットが、作成したカスタムルートテーブルの 1 つに明示的に関連付けられます。これにより、各サブネットがトラフィックをルーティングする方法を明示的にコントロールします。

カスタムルートテーブルで、ルートを追加、削除、変更することができます。カスタムルートテーブルは、関連付けがない場合にのみ削除できます。

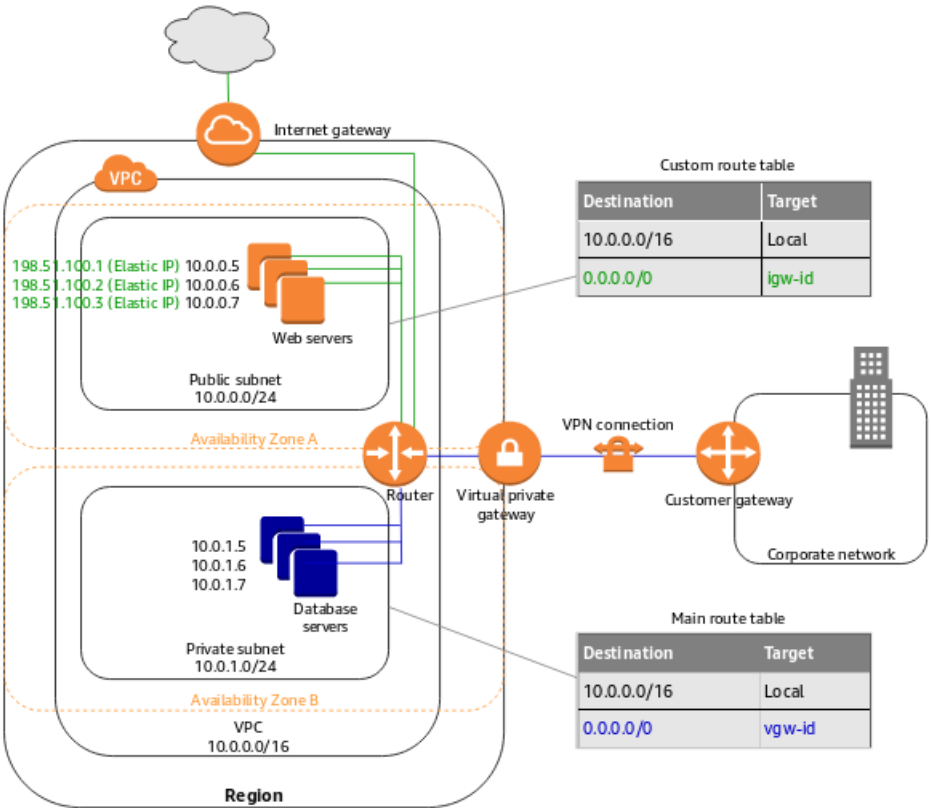
サブネットとルートテーブルの関連付け

VPC 内の各サブネットは、ルートテーブルと関連付ける必要があります。サブネットは、カスタムルートテーブルに明示的に関連付けることも、メインルートテーブルに暗黙的または明示的に関連付けることもできます。サブネットとルートテーブルの関連付けの表示の詳細については、「テーブルに明示的に関連付けられているサブネットまたはゲートウェイを特定する (p. 313)」を参照してください。

Outposts に関連付けられた VPC 内のサブネットには、ローカルゲートウェイの追加ターゲットタイプを設定できます。これは、Outposts 以外のサブネットとの唯一のルーティングの違いです。

例 1: 暗黙的および明示的なサブネットの関連付け

次の図は、インターネットゲートウェイ、仮想プライベートゲートウェイ、パブリックサブネット、および VPN のみのサブネットを持つ VPC のルーティングを示しています。メインルートテーブルには、仮想プライベートゲートウェイへのルートがあります。カスタムルートテーブルは、パブリックサブネットに明示的に関連付けられています。カスタムルートテーブルには、インターネットゲートウェイを経由するインターネット (`0.0.0.0/0`) へのルートがあります。

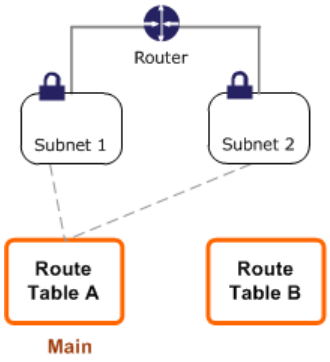


この VPC で新しいサブネットを作成すると、そのサブネットはメインルートテーブルに自動的に暗黙的に関連付けられ、メインルートテーブルは、そのトラフィックを仮想プライベートゲートウェイにルーティングします。逆の設定 (メインルートテーブルにインターネットゲートウェイへのルートが含まれ、カスタムルートテーブルに仮想プライベートゲートウェイへのルートが含まれる) を行うと、新しいサブネットには自動的に、インターネットゲートウェイへのルートが含まれるようになります。

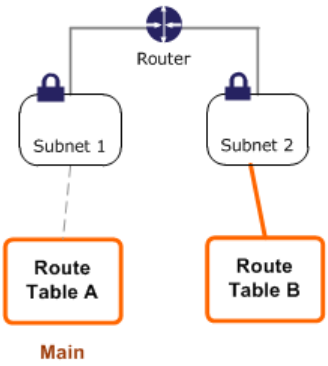
例 2: メインルートテーブルを置き換える

メインルートテーブルに変更を加えることもできます。トラフィックの中断を避けるために、まずカスタムルートテーブルを使用してルート変更をテストすることをお勧めします。テストの結果に満足したら、メインルートテーブルを新しいカスタムテーブルに置き換えられます。

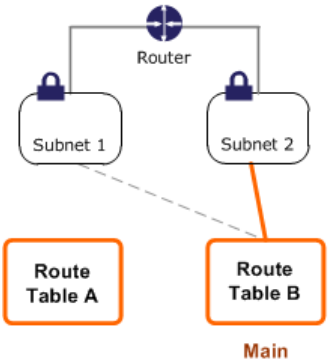
次の図は、メインルートテーブル (ルートテーブル A) に暗示的に関連付けられている 2 つのサブネットを持つ VPC を示しています。カスタムルートテーブル (ルートテーブル B) は、どのサブネットにも関連付けられていません。



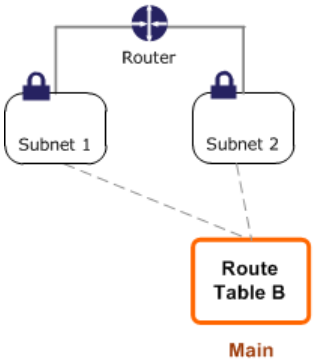
サブネット 2 とルートテーブル B の間には明示的な関連付けを作成できます。



ルートテーブル B をテストしたら、そのテーブルをメインルートテーブルにできます。サブネット 2 とルートテーブル B との間に、まだ明示的な関連付けがあることに注意してください。また、ルートテーブル B は新しいメインルートテーブルなので、サブネット 1 とルートテーブル B の間には暗示的な関連付けがあります。ルートテーブル A はもう使用されていません。



サブネット 2 とルートテーブル B の関連付けを解除しても、サブネット 2 とルートテーブル B との間の暗示的な関連付けは残ります。不要になったルートテーブル A は削除できます。



ゲートウェイルートテーブル

ルートテーブルは、インターネットゲートウェイまたは仮想プライベートゲートウェイに関連付けることができます。ルートテーブルがゲートウェイに関連付けられている場合、ゲートウェイルートテーブルと呼ばれます。ゲートウェイルートテーブルを作成して、VPC に入るトラフィックのルーティングパスを細かく制御できます。例えば、インターネットゲートウェイを介して VPC に入るトラフィックを VPC 内のミドルボックスアプライアンス (セキュリティアプライアンスなど) にリダイレクトして、そのトラフィックをインターセプトできます。

インターネットゲートウェイに関連付けられたゲートウェイルートテーブルは、次のターゲットを持つルートをサポートします。

- デフォルトのローカルルート
- Gateway Load Balancer エンドポイント
- ミドルボックスアプライアンスのネットワークインターフェイス

仮想プライベートゲートウェイに関連付けられたゲートウェイルートテーブルは、次のターゲットを持つルートをサポートします。

- デフォルトのローカルルート
- ミドルボックスアプライアンスのネットワークインターフェイス

ターゲットが Gateway Load Balancer エンドポイントまたはネットワークインターフェイスの場合、次の送信先が許可されます。

- VPC の IPv4 または IPv6 の CIDR ブロック全体。この場合、デフォルトのローカルルートのターゲットを置き換えます。
- VPC 内のサブネットの IPv4 または IPv6 CIDR ブロック全体。これは、デフォルトのローカルルートよりも具体的なルートです。

ゲートウェイルートテーブルのローカルルートのターゲットを VPC のネットワークインターフェイスに変更した場合、後でデフォルトの `local` ターゲットに復元できます。詳細については、「」を参照してくださいローカルルートのターゲットを置換または復元する (p. 319)

次のゲートウェイルートテーブルでは、`172.31.0.0/20` CIDR ブロックを持つサブネット宛てのトラフィックは、特定のネットワークインターフェイスにルーティングされます。VPC 内の他のすべてのサブネット宛てのトラフィックは、ローカルルートを使用します。

送信先	Target
172.31.0.0/16	ローカル
172.31.0.0/20	*eni-id*

次のゲートウェイルートテーブルでは、ローカルルートのターゲットがネットワークインターフェイス ID に置き換えられます。VPC 内のすべてのサブネット宛てのトラフィックは、ネットワークインターフェイスにルーティングされます。

送信先	Target
172.31.0.0/16	*eni-id*

ルールと考慮事項

次のいずれかに該当する場合、ルートテーブルをゲートウェイに関連付けることはできません。

- ルートテーブルには、ネットワークインターフェイス、Gateway Load Balancer エンドポイント、またはデフォルトのローカルルート以外のターゲットを持つ既存のルートが含まれています。
- ルートテーブルには、VPC の範囲外の CIDR ブロックへの既存のルートが含まれます。
- ルートテーブルに対してルート伝達が有効です。

さらに、次のルールと考慮事項が適用されます。

- 個々の VPC CIDR ブロックより大きい範囲も含め、VPC の範囲外の CIDR ブロックにルートを追加することはできません。
- ターゲットとして指定できるのは、`local`、Gateway Load Balancer のエンドポイント、またはネットワークインターフェイスのみです。個々のホスト IP アドレスを含む他のタイプのターゲットは指定できません。詳細については、「the section called “ルーティングオプションの例” (p. 303)」を参照してください。
- 仮想プライベートゲートウェイから Gateway Load Balancer エンドポイントにトラフィックをルーティングすることはできません。ルートテーブルを仮想プライベートゲートウェイに関連付けて、Gateway Load Balancer エンドポイントをターゲットとして使用してルートを追加すると、エンドポイントを送信先とするトラフィックはドロップされます。
- プレフィックスリストを送信先として指定することはできません。
- ゲートウェイルートテーブルを使用して、VPC 外のトラフィック (アタッチされたトランジットゲートウェイを通過するトラフィックなど) を制御またはインターセプトすることはできません。VPC に入るトラフィックをインターセプトし、同じ VPC 内の別のターゲットにのみリダイレクトできます。
- トラフィックがミドルボックスアプライアンスに到達するようにするには、ターゲットネットワークインターフェイスを実行中のインスタンスにアタッチする必要があります。インターネットゲートウェイを流れるトラフィックでは、ターゲットネットワークインターフェイスにはパブリック IP アドレスも必要です。
- ミドルボックスアプライアンスを設定するときは、アプライアンスに関する考慮事項 (p. 309)に注意してください。
- ミドルボックスアプライアンスを介してトラフィックをルーティングする場合、送信先サブネットからのリターントラフィックを同じアプライアンスを介してルーティングする必要があります。非対称ルーティングはサポートされていません。
- ルートテーブルルールは、サブネットから出るすべてのトラフィックに適用されます。サブネットから出るトラフィックは、そのサブネットのゲートウェイルーターの MAC アドレスを送信先とするトラフィックとして定義されます。サブネット内の別のネットワークインターフェイスの MAC アドレスを送信先とするトラフィックは、ネットワーク (レイヤ 3) ではなくデータリンク (レイヤ 2) ルーティングを使用するため、このトラフィックにはルールが適用されません。

セキュリティアプライアンスのルーティングの例については、「ミドルボックスアプライアンスのルーティング (p. 308)」を参照してください。

ルーティングの優先度

一般的に、トラフィックと一致する最も具体的なルートを使用してトラフィックを誘導します。これは、プレフィックスの最長一致と呼ばれます。ルートテーブルに重複または一致するルートがある場合は、追加のルールが適用されます。

最長のプレフィックスの一致

IPv4 および IPv6 アドレスまたは CIDR ブロックへのルートは、互いに独立しています。IPv4 トラフィックまたは IPv6 トラフィックのいずれかに一致する最も具体的なルートを使用して、トラフィックのルーティング方法を決定します。

次の例のサブネットルートテーブルには、インターネットゲートウェイを指す IPv4 インターネットトラフィック (`0.0.0.0/0`) のルートと、ピアリング接続 (`172.31.0.0/16`) を指す IPv4 トラフィック (`pcx-11223344556677889`) のルートが含まれます。`172.31.0.0/16` IP アドレス範囲あてのサブネットからのトラフィックでは、ピアリング接続が使用されます。このルートはインターネットゲートウェイのルートよりも制限が高いためです。VPC 内のターゲットに向けられたすべてのトラフィック (`10.0.0.0/16`) には `local` ルートが適用されるため、VPC 内でルーティングされます。サブネットからのその他のすべてのトラフィックでは、インターネットゲートウェイが使用されます。

送信先	ターゲット
10.0.0.0/16	local
172.31.0.0/16	pcx-11223344556677889
0.0.0.0/0	igw-12345678901234567

ルートプライオリティと伝播ルート

仮想プライベートゲートウェイを VPC にアタッチし、サブネットルートテーブルでルート伝達を有効にしている場合は、Site-to-Site VPN 接続を表すルートが伝達済みルートとしてルートテーブルに自動的に表示されます。

伝播ルートの送信先がローカルルートと重なる場合、伝播ルートがより具体的であっても、ローカルルートが優先されます。伝播ルートの送信先が静的ルートと重複する場合、静的ルートが優先されます。

伝播ルートの送信先が静的ルートの送信先と同じ場合、ターゲットが次のいずれかであれば、静的ルートが優先されます。

- インターネットゲートウェイ
- NAT ゲートウェイ
- ネットワークインターフェイス
- インスタンス ID
- ゲートウェイ VPC エンドポイント
- トランジットゲートウェイ
- VPC ピア接続
- Gateway Load Balancer エンドポイント

詳細については、AWS Site-to-Site VPN ユーザーガイドの「ルートテーブルと VPN ルーティングの優先度」を参照してください。

次のルートテーブルの例にはインターネットゲートウェイへの静的ルート、および仮想プライベートゲートウェイへの伝播されたルートがあります。両方のルートとも、送信先は `172.31.0.0/24` です。インターネットゲートウェイへの静的ルートが優先されるため、`172.31.0.0/24` のすべてのトラフィックがインターネットゲートウェイにルーティングされます。

送信先	ターゲット	伝播済み
10.0.0.0/16	local	いいえ
172.31.0.0/24	vgw-11223344556677889	はい
172.31.0.0/24	igw-12345678901234567	いいえ

ルーティング優先度とプレフィックスリスト

ルートテーブルでプレフィックスリストが参照されている場合は、次のルールが適用されます。

- ルートテーブルに、プレフィックスリストを持つ静的ルートと重複する送信先の CIDR ブロックを持つ静的ルートが含まれている場合、CIDR ブロックを持つ静的ルートが優先されます。

- 伝播ルートがルートテーブルに含まれていて、プレフィックスリストを持つルートと重複する場合は、プレフィックスリストを持つルートが優先されます。
- ルートテーブルで複数のプレフィックスリストが参照されていて、異なるターゲットへの CIDR ブロックが重複する場合、優先されるルートはランダムに選択されます。その後は、同じルートが常に優先されます。
- プレフィックスリストエントリ内の CIDR ブロックがルートテーブルに対して有効でない場合、その CIDR ブロックは無視されます。

ルーティングオプションの例

以下のトピックでは、VPC の特定のゲートウェイまたは接続のルーティングについて説明します。

Options

インターネットゲートウェイへのルーティング

サブネットルートテーブル内のルートをインターネットゲートウェイに追加することで、サブネットをパブリックサブネットにすることができます。そのためには、インターネットゲートウェイを作成して VPC にアタッチ後、IPv4 トラフィックの場合は `0.0.0.0/0`、IPv6 トラフィックの場合は `::/0` を送信先に指定し、インターネットゲートウェイ ID (`igw-xxxxxxxxxxxxxxxxx`) のターゲットを指定してルートを追加します。

送信先	ターゲット
0.0.0.0/0	*igw-id*
::/0	*igw-id*

詳細については、「」を参照してくださいインターネットゲートウェイ (p. 221)

NAT デバイスへのルーティング

プライベートサブネットのインスタンスがインターネットに接続できるようにするには、パブリックサブネットで NAT ゲートウェイを作成するか、NAT インスタンスを起動します。次に、IPv4 インターネット

トラフィック (0.0.0.0/0) を NAT デバイスにルーティングするプライベートサブネットのルートテーブルのルートを追加します。

送信先	ターゲット
0.0.0.0/0	*nat-gateway-id*

また、NAT ゲートウェイを使用するための不要なデータ処理料金を回避したり、特定のトラフィックをプライベートにルーティングしたりするために、他のターゲットへのより具体的なルートを作成することもできます。次の例では、Amazon S3 トラフィック (pl-xxxxxxxx。Amazon S3 の具体的な IP アドレス範囲) はゲートウェイ VPC エンドポイントにルーティングされ、10.25.0.0/16 トラフィックは VPC ピア接続にルーティングされます。pl-xxxxxxxx および 10.25.0.0/16 IP アドレスの範囲は、0.0.0.0/0 よりも具体的です。インスタンスが Amazon S3 またはピア VPC にトラフィックを送信すると、トラフィックはゲートウェイ VPC エンドポイントまたは VPC ピア接続に送信されます。その他のトラフィックはすべて NAT ゲートウェイに送信されます。

送信先	ターゲット
0.0.0.0/0	*nat-gateway-id*
pl-xxxxxxxx	*vpce-id*
10.25.0.0/16	*pcx-id*

詳細については、「NAT ゲートウェイ (p. 237)」および「NAT インスタンス (p. 257)」を参照してください。NAT デバイスは IPv6 トラフィックに使用することはできません。

仮想プライベートゲートウェイへのルーティング

AWS Site-to-Site VPN 接続を使用して、VPC 内のインスタンスが自ネットワークと通信できるようにできます。これを行うには、仮想プライベートゲートウェイを作成し、VPC にアタッチします。次に、ネットワークの送信先と仮想プライベートゲートウェイ (vgw-xxxxxxxxxxxxxxxxx) のターゲットを含むルートをサブネットルートテーブルに追加します。

送信先	ターゲット
10.0.0.0/16	*vgw-id*

その後、Site-to-Site VPN 接続を作成し、設定することができます。詳細については、AWS Site-to-Site VPN ユーザーガイドの「AWS Site-to-Site VPN とは」および「ルートテーブルと VPN ルーティングの優先度」を参照してください。

仮想プライベートゲートウェイ上の Site-to-Site VPN 接続は、IPv6 トラフィックをサポートしません。ただし、仮想プライベートゲートウェイを介した AWS Direct Connect 接続への IPv6 トラフィックのルーティングはサポートされています。詳細については、AWS Direct Connect ユーザーガイドを参照してください。

AWS Outposts ローカルゲートウェイへのルーティング

AWS Outposts に関連付けられた VPC 内のサブネットには、ローカルゲートウェイの追加ターゲットタイプを設定できます。送信先アドレス 192.168.10.0/24 のトラフィックをローカルゲートウェイでカスタ

マーネットワークにルーティングする場合を考えます。これを行うには、送信先ネットワークとローカルゲートウェイ (`lgw-xxxx`) のターゲットで次のルートを追加します。

送信先	ターゲット
192.168.10.0/24	*lgw-id*

Wavelength ゾーンキャリアゲートウェイへのルーティング

Wavelength Zones にあるサブネットには、キャリアゲートウェイの追加のターゲットタイプを設定できます。すべての非 VPC トラフィックをキャリアネットワークにルーティングするために、キャリアゲートウェイでトラフィックをルーティングする場合を考えてみます。これを行うには、VPC へのキャリアゲートウェイを作成し、アタッチしてから、次のルートを追加します。

送信先	ターゲット
0.0.0.0/0	*cagw-id*
::/0	*cagw-id*

VPC ピア接続へのルーティング

VPC ピアリング接続は、プライベート IPv4 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 ピア接続にある VPC 間のトラフィックのルーティングを有効にするには、VPC ピア接続を指す 1 つ以上のサブネットルートテーブルにルートを追加する必要があります。これにより、ピア接続で他の VPC の CIDR ブロックのすべてまたは一部にアクセスできます。同様に、他の VPC の所有者は、自分のサブネットのルートテーブルにルートを追加して、ルーティング対象の VPC にトラフィックを送り返す必要があります。

例えば、次の情報を持つ 2 つの VPC 間に VPC ピアリング接続 (`pcx-11223344556677889`) があるとします。

- VPC A: CIDR ブロックは 10.0.0.0/16 です
- VPC B: CIDR ブロックは 172.31.0.0/16 です

VPC 間のトラフィックを有効にし、両方の VPC の IPv4 CIDR ブロック全体にアクセスできるようにするには、VPC A のルートテーブルを次のように設定します。

送信先	ターゲット
10.0.0.0/16	ローカル
172.31.0.0/16	pcx-11223344556677889

VPC B のルートテーブルは次のように設定します。

送信先	Target
172.31.0.0/16	ローカル
10.0.0.0/16	pcx-11223344556677889

VPC ピアリング接続では、VPC とインスタンスで IPv6 通信が有効な場合、VPC のインスタンス間で IPv6 通信をサポートできます。詳細については、「」を参照してくださいVPC とサブネット (p. 106) VPC 間の IPv6 トラフィックのルーティングを有効にするには、VPC ピアリング接続をポイントするルートテーブルにルートを追加して、ピア VPC の IPv6 CIDR ブロックのすべての部分にアクセスできるようにする必要があります。

例えば、同じ VPC ピアリング接続 (`pcx-11223344556677889`) を使用して、VPC に次の情報を含めるとします。

- VPC A: IPv6 CIDR ブロックは `2001:db8:1234:1a00::/56`
- VPC B: IPv6 CIDR ブロックは `2001:db8:5678:2b00::/56`

VPC ピアリング接続で IPv6 通信を有効にするには、VPC A のサブネットルートテーブルに次のルートを追加します。

送信先	ターゲット
10.0.0.0/16	ローカル
172.31.0.0/16	pcx-11223344556677889
2001:db8:5678:2b00::/56	pcx-11223344556677889

VPC B のルートテーブルに次のルートを追加します。

送信先	Target
172.31.0.0/16	ローカル
10.0.0.0/16	pcx-11223344556677889
2001:db8:1234:1a00::/56	pcx-11223344556677889

VPC ピア接続の詳細については、「Amazon VPC ピアリングガイド」を参照してください。

ClassicLink のルーティング

ClassicLink は、VPC に EC2 Classic インスタンスをリンクし、プライベート IPv4 アドレスを使用して EC2-Classic インスタンスと VPC のインスタンス間の通信を可能にする機能です。ClassicLink の詳細については、「ClassicLink (p. 292)」を参照してください。

ClassicLink 用に VPC を有効にすると、すべてのサブネットルートテーブルに、送信先が `10.0.0.0/8` で、ターゲットが `local` であるルートが追加されます。これによって、VPC 内のインスタンスと、VPC にリンクされている EC2-Classic インスタンスとの間で通信が可能になります。ClassicLink が有効な VPC に別のルートテーブルを追加する場合、送信先が `10.0.0.0/8` で、ターゲットが `local` であるルートが自動的に追加されます。VPC の ClassicLink を無効にすると、このルートはサブネットのすべてのルートテーブルから自動的に削除されます。

サブネットのいずれかのルートテーブルに、10.0.0.0/8 CIDR 内のアドレス範囲で既存のルートが存在する場合、ClassicLink 用に VPC を有効にすることができません。これには、10.0.0.0/16 および 10.1.0.0/16 の IP アドレス範囲を持つ、VPC のローカルルートは含まれません。

既に ClassicLink 用に VPC を有効にしている場合、10.0.0.0/8 IP アドレス範囲のルートテーブルに、より詳細なルートを追加できない場合があります。

VPC ピアリング接続を変更して、VPC のインスタンスとピア VPC にリンクされた EC2-Classic インスタンス間の通信を有効にするため、送信先を 10.0.0.0/8、ターゲットを local として、静的ルートが自動的にルートテーブルに追加されます。VPC ピアリング接続を変更して、VPC にリンクされたローカルの EC2-Classic インスタンスと、ピア VPC のインスタンス間で通信を有効にする場合、送信先をピア VPC CIDR ブロック、ターゲットを VPC ピアリング接続として、メインルートテーブルにルートを手動で追加する必要があります。EC2-Classic インスタンスは、ピア VPC へのルーティングについてメインルートテーブルに依存します。詳細については、Amazon VPC ピアリングガイドの「ClassicLink を使用した設定」を参照してください。

ゲートウェイ VPC エンドポイントへのルーティング

ゲートウェイ VPC エンドポイントにより、VPC と他の AWS のサービスとをプライベートに接続できます。ゲートウェイエンドポイントを作成するときは、ゲートウェイエンドポイントによって使用されるサブネットルートテーブルを VPC で指定します。ルートは自動的に各ルートテーブル追加されて、送信先としてサービス (pl-*xxxxxxxx*) のプレフィックスリスト ID、ターゲットとしてエンドポイント ID (vpce-*xxxxxxxxxxxxxxxxx*) が登録されます。エンドポイントルートを明示的に削除または変更することはできませんが、エンドポイントで使用されるルートテーブルは変更できます。

エンドポイントのルーティングの詳細について、また AWS のサービスへのルートに対する影響については、「ゲートウェイエンドポイントのルーティング」を参照してください。

Egress-Only インターネットゲートウェイへのルーティング

VPC で Egress-Only インターネットゲートウェイを作成して、プライベートサブネットのインスタンスを有効にしてインターネットへのアウトバウンド通信を開始することができますが、インターネットはインスタンスとの接続を開始することはできません。Egress-Only インターネットゲートウェイは、IPv6 トラフィックでのみ使用されます。Egress-Only インターネットゲートウェイのルーティングを設定するには、Egress-Only インターネットゲートウェイに IPv6 インターネットトラフィック (::/0) をルーティングするプライベートサブネットのルートテーブルにルートを追加します。

送信先	ターゲット
::/0	*eigw-id*

詳細については、「」を参照してくださいEgress-Only インターネットゲートウェイ (p. 228)

トランジットゲートウェイのルーティング

VPC をトランジットゲートウェイにアタッチするときは、トラフィックがトランジットゲートウェイを通過してルーティングするよう、サブネットルートテーブルにルートを追加する必要があります。

トランジットゲートウェイに 3 つの VPC がアタッチされている次のシナリオを検討します。このシナリオでは、アタッチメントはすべて、トランジットゲートウェイのルートテーブルに関連付けられ、トランジットゲートウェイのルートテーブルに伝播されます。そのため、アタッチメントはすべて、単純なレイ

ヤー 3 IP ハブとしてトランジットゲートウェイを提供しながら、パケットを相互にルーティングできます。

例えば、次の情報を持つ 2 つの VPC があるとします。

- VPC A: 10.1.0.0/16, attachment ID tgw-attach-11111111111111111
- VPC B: 10.2.0.0/16, attachment ID tgw-attach-22222222222222222

VPC 間のトラフィックを有効にし、トランジットゲートウェイにアクセスできるようにするには、VPC A のルートテーブルを次のように設定します。

送信先	ターゲット
10.1.0.0/16	ローカル
10.0.0.0/8	*tgw-id*

以下は、VPC アタッチメントのトランジットゲートウェイルートテーブルエントリの例です。

送信先	ターゲット
10.1.0.0/16	tgw-attach-11111111111111111
10.2.0.0/16	tgw-attach-22222222222222222

Transit Gateway ルートテーブルの詳細については、Amazon VPC Transit Gateway の「ルーティング」を参照してください。

ミドルボックスアプライアンスのルーティング

ミドルボックスアプライアンスを VPC のルーティングパスに追加できます。以下は想定されるユースケースです。

- インターネットゲートウェイまたは仮想プライベートゲートウェイを介して VPC に入るトラフィックを、VPC のミドルボックスアプライアンスにルーティングして、インターセプトします。ミドルボックスのルーティングウィザードを使用して、AWS がゲートウェイ、ミドルボックス、送信先サブネットの適切なルートテーブルを自動的に設定できるようにします。詳細については、「the section called “ミドルボックスルーティングウィザードの操作” (p. 320)」を参照してください。
- 2 つのサブネット間のトラフィックをミドルボックスアプライアンスに転送します。そのためには、一方のサブネットのサブネットCIDR と一致させるサブネットルートテーブルのルートを作成して、Gateway Load Balancer エンドポイント、NAT ゲートウェイ、Network Firewall endpoint エンドポイント、またはアプライアンスのネットワークインターフェイスをターゲットとして指定します。または、サブネットから他のサブネットにすべてのトラフィックをリダイレクトするには、ローカルルートのターゲットを Gateway Load Balancer エンドポイント、NAT ゲートウェイ、またはネットワークインターフェイスに置き換えます。

ニーズに合わせてアプライアンスを設定できます。例えば、すべてのトラフィックをスクリーニングするセキュリティアプライアンス、または WAN アクセラレーションアプライアンスを設定できます。アプライアンスは VPC のサブネットで Amazon EC2 インスタンスとしてデプロイされ、サブネット内の Elastic Network Interface (ネットワークインターフェイス) で表されます。

目的のサブネットのルートテーブルでルート伝達を有効にしている場合は、ルートの優先順位に注意してください。最も具体的なルートが優先され、ルートが一致する場合は、伝達されたルートよりも静的ルー

トが優先されます。ルートを確認して、トラフィックが正しくルーティングされていること、およびルート伝達を有効または無効にした場合に (ジャンボフレームをサポートする AWS Direct Connect 接続にはルート伝達が必要など)、意図しない結果がないことを確認します。

インバウンド VPC トラフィックをアプライアンスにルーティングするには、ルートテーブルをインターネットゲートウェイまたは仮想プライベートゲートウェイに関連付け、アプライアンスのネットワークインターフェイスを VPC トラフィックのターゲットとして指定します。詳細については、「」を参照してくださいゲートウェイルートテーブル (p. 299) また、サブネットから別のサブネットのミドルボックスアプライアンスにアウトバウンドトラフィックをルーティングすることもできます。

ミドルボックスのルーティングの例については、「ミドルボックスルーティング (p. 82)」を参照してください。

目次

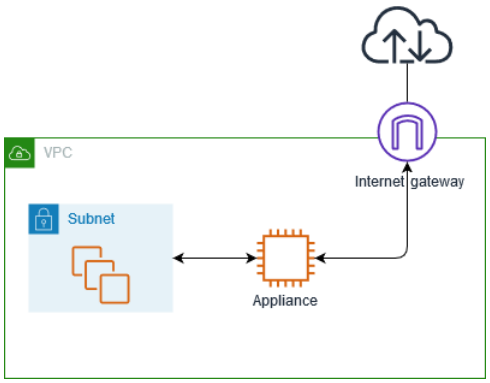
アプライアンスに関する考慮事項

AWS Marketplace からサードパーティー製アプライアンスを選択することも、独自のアプライアンスを設定することもできます。アプライアンスを作成または設定するときは、次の点に注意してください。

- アプライアンスは、送信元トラフィックまたは送信先トラフィックとは別のサブネットに設定する必要があります。
- アプライアンスでの送信元/送信先のチェックを無効にする必要があります。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「送信元または送信先チェックの変更」を参照してください。
- アプライアンスを経由して、同じサブネットのホスト間でトラフィックをルーティングすることはできません。
- アプライアンスは、ネットワークアドレス変換 (NAT) を実行する必要はありません。
- ローカルルートよりも具体的なルートを追加できます。より具体的なルートを使用して、VPC 内のサブネット間のトラフィック (East-West トラフィック) をミドルボックスアプライアンスにリダイレクトできます。ルートの送信先は、VPC 内のサブネットの IPv4 または IPv6 CIDR ブロック全体と一致させる必要があります。
- IPv6 トラフィックをインターセプトするには、必ず IPv6 に VPC、サブネット、アプライアンスを設定します。詳細については、「」を参照してくださいVPC とサブネットの使用 (p. 115) 仮想プライベートゲートウェイは IPv6 トラフィックをサポートしません。

ゲートウェイとアプライアンス間のトラフィックのルーティング

インバウンド VPC トラフィックをアプライアンスにルーティングするには、ルートテーブルをインターネットゲートウェイまたは仮想プライベートゲートウェイに関連付け、アプライアンスのネットワークインターフェイスを VPC トラフィックのターゲットとして指定します。次の例では、VPC にはインターネットゲートウェイ、アプライアンス、およびインスタンスを持つサブネットがあります。インターネットからのトラフィックは、アプライアンスを介してルーティングされます。



このルートテーブルをインターネットゲートウェイまたは仮想プライベートゲートウェイに関連付けます。最初のエントリはローカルルートです。2 番目のエントリは、サブネット宛ての IPv4 トラフィックをアプライアンスのネットワークインターフェイスに送信します。このルートは、デフォルトのローカルルートよりも具体的なルートです。

送信先	ターゲット
VPC CIDR	ローカル
Subnet CIDR	*########################## ID*

または、ローカルルートのターゲットをアプライアンスのネットワークインターフェイスに置き換えることもできます。これを行うと、後で VPC に追加するサブネットを送信先とするトラフィックを含め、すべてのトラフィックがアプライアンスに自動的にルーティングされるようになります。

送信先	ターゲット
VPC CIDR	*########################## ID*

サブネットから別のサブネットのアプライアンスにトラフィックをルーティングするには、アプライアンスのネットワークインターフェイスにトラフィックをルーティングするルートをサブネットルートテーブルに追加します。この送信先は、ローカルルートの宛先より具体性を低くする必要があります。例えば、インターネットを送信先とするトラフィックの場合、宛先に `0.0.0.0/0` (すべての IPv4 アドレス) を指定します。

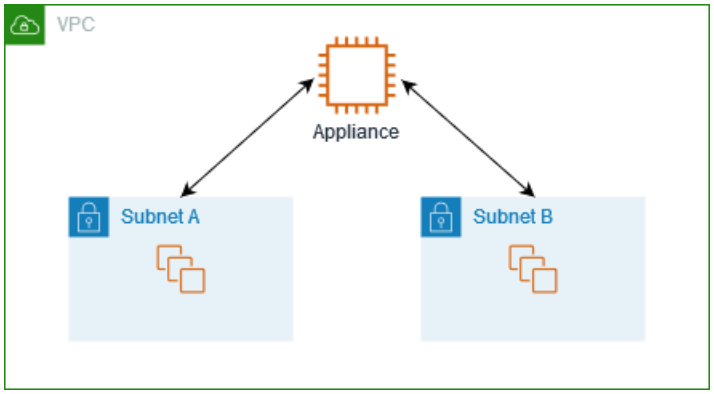
送信先	ターゲット
VPC CIDR	ローカル
0.0.0.0/0	*########################## ID*

次に、アプライアンスのサブネットに関連付けられたルートテーブルで、トラフィックをインターネットゲートウェイまたは仮想プライベートゲートウェイに送り返すルートを追加します。

送信先	ターゲット
VPC CIDR	ローカル
0.0.0.0/0	*igw-id*

サブネット間トラフィックをアプライアンスへルーティング

特定のサブネットを送信先とするトラフィックを、アプライアンスのネットワークインターフェイスにルーティングできます。次の例では、VPC に 2 つのサブネットと 1 つのアプライアンスが含まれています。サブネット間のトラフィックは、アプライアンスを介してルーティングされます。



セキュリティグループ

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のインスタンス間でトラフィックをルーティングする場合、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フローを許可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。

Routing

次に、サブネット A のルートテーブルの例を示します。最初のエントリにより、VPC 内のインスタンスが通信できるようになります。2 番目のエントリは、サブネット A からサブネット B へのすべてのトラフィックをアプライアンスのネットワークインターフェイスにルーティングします。

送信先	ターゲット
VPC CIDR	ローカル
Subnet B CIDR	*####################### ID*

次に、サブネット B のルートテーブルの例を示します。最初のエントリにより、VPC 内のインスタンスが相互に通信できるようになります。2 番目のエントリは、サブネット B からサブネット A へのすべてのトラフィックをアプライアンスのネットワークインターフェイスにルーティングします。

送信先	ターゲット
VPC CIDR	ローカル
Subnet A CIDR	*####################### ID*

または、ローカルルートのターゲットをアプライアンスのネットワークインターフェイスに置き換えることもできます。これを行うと、後で VPC に追加するサブネットを送信先とするトラフィックを含め、すべてのトラフィックがアプライアンスに自動的にルーティングされるようになります。

送信先	ターゲット
VPC CIDR	*##################### ID*

プレフィックスリストを使用したルーティング

AWS リソース全体で同じ CIDR ブロックのセットを頻繁に参照する場合は、カスタマーマネージドプレフィックスリスト (p. 276)を作成して、それらをグループ化できます。その後、ルートテーブルエントリの送信先としてプレフィックスリストを指定できます。後でプレフィックスリストのエントリを追加または削除でき、ルートテーブルを更新する必要はありません。

例えば、複数の VPC アタッチメントを持つトランジットゲートウェイがあるとします。VPC は、次の CIDR ブロックを持つ 2 つの特定の VPC アタッチメントと通信できる必要があります。

- 10.0.0.0/16
- 10.2.0.0/16

両方のエントリを持つプレフィックスリストを作成します。サブネットルートテーブルで、ルートを作成し、送信先としてプレフィックスリストを指定して、ターゲットとしてトランジットゲートウェイを指定します。

送信先	Target
172.31.0.0/16	ローカル
pl-123abc123abc123ab	*tgw-id*

プレフィックスリストのエントリの最大数は、ルートテーブル内のエントリ数と同じになります。

Gateway Load Balancer エンドポイントにルーティングする

Gateway Load Balancer を使用すると、ファイアウォールなどの仮想アプライアンスのフリートにトラフィックを分散できます。VPC エンドポイントサービス設定を作成して、ロードバランサーをサービスとして設定できます。その後、VPC 内に Gateway Load Balancer エンドポイントを作成し、VPC をサービスに接続します。

トラフィックを (例えば、セキュリティ検査のために) Gateway Load Balancer にルーティングするには、ルートテーブルで Gateway Load Balancer エンドポイントをターゲットとして指定します。

Gateway Load Balancer の背後にあるセキュリティアプライアンスの例については、「the section called “セキュリティ VPC のゲートウェイロードバランサーの背後にあるセキュリティアプライアンス” (p. 85)」を参照してください。

ルートテーブルで Gateway Load Balancer エンドポイントを指定するには、VPC エンドポイントの ID を使用します。例えば、10.0.1.0/24 のトラフィックを Gateway Load Balancer エンドポイントにルーティングするには、次のルートを追加します。

送信先	ターゲット
10.0.1.0/24	*vpc-endpoint-id*

詳細については、「Gateway Load Balancer の開始方法」をご参照ください。

ルートテーブルの使用

以下のタスクは、ルートテーブルを操作する方法を示しています。

> Note
>
> コンソールで VPC ウィザードを使用して、ゲートウェイが含まれる VPC を作成すると、そのゲートウェイを使用するようにルートテーブルが自動的に更新されます。コマンドラインツールまたは API を使用して VPC をセットアップする場合、ルートテーブルはご自身で更新する必要があります。

タスク

サブネット用のルートテーブルの決定

サブネットが関連付けられているルートテーブルを特定するには、Amazon VPC コンソールでサブネットの詳細を確認します。

サブネットのルートテーブルを決定するには

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、[Subnets] を選択します。
3. [Route Table] タブを選択すると、ルートテーブルの ID とそのルートが表示されます。メインルートテーブルの場合、関連付けが暗示的か明示的かはコンソールに表示されません。メインルートテーブルとの関連付けが明示的かどうかを特定する方法については、「テーブルに明示的に関連付けられているサブネットまたはゲートウェイを特定する (p. 313)」を参照してください。

テーブルに明示的に関連付けられているサブネットまたはゲートウェイを特定する

ルートテーブルに明示的に関連付けられているサブネットまたはゲートウェイとその数を特定できます。

メインルートテーブルは、サブネットとの明示的な関連付けと暗示的な関連付けを持つことができます。カスタムルートテーブルは、明示的な関連付けしか持つことができません。

どのルートテーブルにも明示的に関連付けられていないサブネットは、メインルートテーブルに暗示的に関連付けられています。メインルートテーブルには、サブネットを明示的に関連付けることができます。その理由の例については、「メインルートテーブルの置換 (p. 317)」を参照してください。

コンソールを使用して明示的に関連付けられているサブネットを特定するには

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] を選択します。
3. [Explicit subnet association (明示的なサブネットの関連付け)] 列を表示して、明示的に関連付けられたサブネットを特定します。
4. 必要なルートテーブルを選択します。
5. 詳細ペインの [Subnet Associations] タブを選択します。このタブには、テーブルに明示的に関連付けられているサブネットが表示されています。また、どのルートテーブルにも関連付けられていない (つまり、メインルートテーブルに暗示的に関連付けられている) サブネットも表示されます。

コンソールを使用して明示的に関連付けられているゲートウェイを特定するには

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] を選択します。
3. [Edge associations (エッジの関連付け)] 列を表示して、関連付けられたゲートウェイを特定します。
4. 必要なルートテーブルを選択します。
5. 詳細ペインの [Edge Associations (エッジの関連付け)] タブを選択します。ルートテーブルに関連付けられているゲートウェイが一覧表示されます。

コマンドラインを使用して 1 つ以上のルートテーブルを記述し、その関連付けを表示するには

- describe-route-tables (AWS CLI)
- Get-EC2RouteTable (AWS Tools for Windows PowerShell)

カスタムルートテーブルを作成する

Amazon VPC コンソールを使用して VPC のカスタムルートテーブルを作成できます。

コンソールを使用してカスタムルートテーブルを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Route Tables] を選択します。
3. [ルートテーブルの作成] を選択します。
4. (オプション) [名前タグ] には、ルートテーブルの名前を入力します。
5. [VPC] で、ユーザーの VPC を選択します。
6. (オプション) タグを追加または削除します。

   [タグの追加] [タグの追加] を選択して、以下を実行します。

   - [キー] にはキー名を入力します。
   - [値] にキー値を入力します。

   [タグの削除] タグのキーと値の右側にある [削除] ボタン ("X") を選択します。
7. [Create] を選択します。

コマンドラインを使用してカスタムルートテーブルを作成するには

- create-route (AWS CLI)
- New-EC2RouteTable (AWS Tools for Windows PowerShell)

ルートテーブルのルートの追加と削除

ルートテーブルのルートは追加、削除、変更できます。変更できるのは、追加したルートのみです。

Site-to-Site VPN 接続の静的ルートの操作の詳細については、AWS Site-to-Site VPN ユーザーガイドの「Site-to-Site VPN 接続の静的ルートの編集」を参照してください。

コンソールを使用してルートを変更するには、またはルートテーブルに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 [ルートテーブル] を選択して、ルートテーブルを選択します。
3. [アクション]、[ポリシーの編集] の順に選択します。
4. ルートを追加するには、[ルートの追加] を選択します。[送信先] に、送信先 CIDR ブロック、単一の IP アドレス、またはプレフィックスリストの ID を入力します。
5. 既存のルートを変更するには、[送信先] で、宛先 CIDR ブロックまたは 1 つの IP アドレスを置き換えます。[ターゲット] で、ターゲットを選択します。
6. [Save routes] を選択します。

コマンドラインを使用してルートテーブルにルートを追加するには

- create-route (AWS CLI)
- New-EC2Route (AWS Tools for Windows PowerShell)

> **Note**
>
> コマンドラインツールまたは API を使用してルートを追加すると、送信先 CIDR ブロックは自動的に正規形式に変更されます。例えば、CIDR ブロックに `100.68.0.18/18` を指定した場合、送信先 CIDR ブロックが `100.68.0.0/18` であるルートが作成されます。

コマンドラインを使用してルートテーブル内の既存のルートを置き換えるには

- replace-route (AWS CLI)
- Set-EC2Route (AWS Tools for Windows PowerShell)

コンソールを使用してルートをルートテーブルから削除するには

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 [ルートテーブル] を選択して、ルートテーブルを選択します。
3. [アクション]、[ポリシーの編集] の順に選択します。
4. 削除するルートの右側にある削除ボタン ([x]) を選択します。
5. 完了したら、[Save routes (ルートを保存)] を選択します。

コマンドラインを使用してルートテーブルからルートを削除するには

- delete-route (AWS CLI)

- Remove-EC2Route (AWS Tools for Windows PowerShell)

ルート伝達は有効または無効にできます。

ルート伝達により、仮想プライベートゲートウェイはルートテーブルにルートを自動的に伝達できます。つまり、ルートテーブルへの VPN ルートを手動で入力する必要はありません。ルートの伝播は有効または無効にできます。

このプロセスを完了するには、仮想プライベートゲートウェイが必要です。

VPN ルーティングオプションの詳細については、Site-to-Site VPN ユーザーガイドの「Site-to-Site VPN ルーティングオプション」を参照してください。

コンソールを使用してルート伝達を有効にするには

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] を選択して、ルートテーブルを選択します。
3. [アクション]、[Edit route propagation (ルート伝達を編集)] の順に選択します。
4. 仮想プライベートゲートウェイの横にある [Enable] (有効化) チェックボックスをオンにし、[Save] (保存) を選択します。

コマンドラインを使用してルート伝達を有効にするには

- enable-vgw-route-propagation (AWS CLI)
- Enable-EC2VgwRoutePropagation (AWS Tools for Windows PowerShell)

コンソールを使用してルート伝達を無効にするには

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] を選択して、ルートテーブルを選択します。
3. [アクション]、[Edit route propagation (ルート伝達を編集)] の順に選択します。
4. [Propagate] チェックボックスをオフにし、[Save] を選択します。

コマンドラインを使用してルート伝達を無効にするには

- disable-vgw-route-propagation (AWS CLI)
- Disable-EC2VgwRoutePropagation (AWS Tools for Windows PowerShell)

サブネットをルートテーブルに関連付ける

ルートテーブルのルートを特定のサブネットに適用するには、ルートテーブルをサブネットに関連付ける必要があります。ルートテーブルは複数のサブネットに関連付けることができます。ただし、サブネットは一度に 1 つのルートテーブルにのみ関連付けることができます。どのテーブルにも明示的に関連付けられていないサブネットは、デフォルトでメインルートテーブルに暗示的に関連付けられています。

コンソールを使用してルートテーブルをサブネットに関連付けるには

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] を選択して、ルートテーブルを選択します。
3. [Subnet Associations] (サブネットの関連付け) タブで、 [Edit subnet associations] (サブネットの関連付けの編集) を選択します。

4. ルートテーブルに関連付けるサブネットのチェックボックスをオンにしてから、[Save associations] (関連付けの保存) を選択します。

コマンドラインを使用してサブネットをルートテーブルに関連付けるには

- associate-route-table (AWS CLI)
- Register-EC2RouteTable (AWS Tools for Windows PowerShell)

サブネット用のルートテーブルの編集

サブネットのルートテーブルの関連付けを変更できます。

ルートテーブルを変更すると、変更後のルートテーブルに同じターゲットへの同じトラフィックのルートが含まれていない限り、サブネット内の既存の接続は削除されます。

コンソールを使用してサブネットとルートテーブルの関連付けを変更するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Subnets] を選択し、サブネットを選択します。
3. [ルートテーブル] タブで、[Edit route table association (ルートテーブルの関連付けを編集)] を選択します。
4. [ルートテーブル ID] リストから、サブネットを関連付ける新しいルートテーブルを選択し、[保存] を選択します。

コマンドラインを使用してサブネットに関連付けられたルートテーブルを変更するには

- replace-route-table-association (AWS CLI)
- Set-EC2RouteTableAssociation (AWS Tools for Windows PowerShell)

サブネットとルートテーブルの関連付けを解除する

サブネットとルートテーブルの関連付けを解除することができます。別のルートテーブルにサブネットを関連付けるまでは、メインルートテーブルに暗示的に関連付けられています。

コンソールを使用してサブネットとルートテーブルの関連付けを解除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Route Tables] を選択して、ルートテーブルを選択します。
3. [Subnet associations] (サブネットの関連付け) タブで、[Edit subnet associations] (サブネットの関連付けの編集) を選択します。
4. サブネットのチェックボックスをオフにして、[Save associations] (関連付けの保存) を選択します。

コマンドラインを使用してサブネットとルートテーブルの関連付けを解除するには

- disassociate-route-table (AWS CLI)
- Unregister-EC2RouteTable (AWS Tools for Windows PowerShell)

メインルートテーブルの置換

VPC でメインルートテーブルを別のルートテーブルに変更できます。

コンソールを使用してメインルートテーブルを置き換えるには

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] を選択します。
3. 新しいメインルートテーブルにするサブネットルートテーブルを選択し、[Actions] (アクション)、[Set Main Route Table] (メインルートテーブルの設定) の順に選択します。
4. 確認のダイアログボックスで [OK] を選択します。

コマンドラインを使用してメインルートテーブルを置き換えるには

- replace-route-table-association (AWS CLI)
- Set-EC2RouteTableAssociation (AWS Tools for Windows PowerShell)

次の手順では、サブネットとメインルートテーブルの間の明示的な関連付けを解除する方法について説明します。これにより、サブネットとメインルートテーブルが暗示的に関連付けられます。そのプロセスは、サブネットと任意のルートテーブルの関連付け解除と同じです。

メインルートテーブルとの明示的な関連付けを解除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [Route Tables] を選択して、ルートテーブルを選択します。
3. [Subnet associations] (サブネットの関連付け) タブで、[Edit subnet associations] (サブネットの関連付けの編集) を選択します。
4. サブネットを選択し、[Save] (保存) を選択します。

ゲートウェイとルートテーブルの関連付け

インターネットゲートウェイまたは仮想プライベートゲートウェイをルートテーブルに関連付けることができます。詳細については、「」を参照してくださいゲートウェイルートテーブル (p. 299)

コンソールを使用してゲートウェイをルートテーブルに関連付けるには

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] を選択して、ルートテーブルを選択します。
3. [アクション]、[Edit edge associations (エッジの関連付けの編集)] の順に選択します。
4. ゲートウェイを選択し、[保存] を選択します。

AWS CLI を使用してゲートウェイをルートテーブルに関連付けるには

[associate-route-table] コマンドを使用します。次の例では、インターネットゲートウェイ `igw-11aa22bb33cc44dd1` をルートテーブル `rtb-01234567890123456` に関連付けます。

```
aws ec2 associate-route-table --route-table-id rtb-01234567890123456 --gateway-id
 igw-11aa22bb33cc44dd1
```

ルートテーブルからゲートウェイの関連付けを解除する

インターネットゲートウェイまたは仮想プライベートゲートウェイをルートテーブルから関連付け解除できます。

コンソールを使用してゲートウェイをルートテーブルに関連付けるには

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] を選択して、ルートテーブルを選択します。
3. [アクション]、[Edit edge associations (エッジの関連付けの編集)] の順に選択します。
4. 関連付けを解除するゲートウェイを選択します。
5. [Save] を選択します。

コマンドラインを使用してゲートウェイとルートテーブルの関連付けを解除するには

- disassociate-route-table (AWS CLI)
- Unregister-EC2RouteTable (AWS Tools for Windows PowerShell)

ローカルルートのターゲットを置換または復元する

デフォルトのローカルルートのターゲットを変更できます。ローカルルートのターゲットを置き換えた場合は、後でデフォルトの `local` ターゲットに戻すことができます。VPC に複数の CIDR ブロック (p. 110)がある場合、ルートテーブルには複数のローカルルートが、CIDR ブロックごとに 1 つあります。必要に応じて、各ローカルルートのターゲットを置き換えまたは復元できます。

コンソールを使用してローカルルートのターゲットを置き換えるには

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] を選択して、ルートテーブルを選択します。
3. [アクション]、[ポリシーの編集] の順に選択します。
4. [ターゲット] で、ターゲットを選択します。
5. [Save routes] を選択します。

コンソールを使用してローカルルートのターゲットを復元するには

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] を選択して、ルートテーブルを選択します。
3. [アクション]、[ポリシーの編集] の順に選択します。
4. [ターゲット] で、[ローカル] を選択します。
5. [Save routes] を選択します。

AWS CLI を使用してローカルルートのターゲットを置き換えるには

[replace-route] コマンドを使用します。次の例では、ローカルルートのターゲットを `eni-11223344556677889` に置き換えます 。

```
aws ec2 replace-route --route-table-id rtb-01234567890123456 --destination-cidr-block
 10.0.0.0/16 --network-interface-id eni-11223344556677889
```

AWS CLI を使用してローカルルートのターゲットを復元するには

次の例では、ルートテーブル `rtb-01234567890123456` のローカルターゲットを復元します。

```
aws ec2 replace-route --route-table-id rtb-01234567890123456 --destination-cidr-block
 10.0.0.0/16 --local-target
```

ルートテーブルを削除する

ルートテーブルは、サブネットが関連付けられていない場合にのみ削除できます。メインルートテーブルを削除することはできません。

コンソールを使用してルートテーブルを削除するには

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、[Route Tables] を選択します。
3. ルートテーブルを選択してから [アクション]、[ルートテーブルの削除] の順に選択します。
4. 確認ダイアログボックスで、[ルートテーブルの削除] を選択します。

コマンドラインを使用してルートテーブルを削除するには

- delete-route-table (AWS CLI)
- Remove-EC2RouteTable (AWS Tools for Windows PowerShell)

ミドルボックスルーティングウィザードの操作

例えば、トラフィックをセキュリティアプライアンスにリダイレクトするなど、VPC に出入りするトラフィックのルーティングパスを細かく制御する場合は、VPC コンソールでミドルボックスルーティングウィザードを使用できます。ミドルボックスルーティングウィザードを使用すると、必要なルートテーブルとルート (ホップ) を自動的に作成して、必要に応じてトラフィックをリダイレクトできます。

ミドルボックスルーティングウィザードで、次のシナリオでルーティングを設定できます。

- ミドルボックスアプライアンス (セキュリティアプライアンスとして設定された Amazon EC2 インスタンスなど) にトラフィックをルーティングします。
- Gateway Load Balancer へのトラフィックのルーティング 詳細については、「Gateway Load Balancer ユーザーガイド」を参照してください。

詳細については、「the section called “ミドルボックスルーティング” (p. 82)」を参照してください。

目次

ミドルボックスルーティングウィザードの前提条件

「the section called “ミドルボックスルーティングウィザードに関する考慮事項” (p. 323)」を確認します。ミドルボックスルーティングウィザードを使用する前に、次の情報を確認してください。

- VPC。
- インターネットゲートウェイ、仮想プライベートゲートウェイ、ネットワークインターフェイスなど、トラフィックが送信される VPC のリソース。
- ミドルボックスのネットワークインターフェイスまたは Gateway Load Balancer エンドポイント。

- トラフィックの送信先サブネットです。

ミドルボックスのルーティングウィザードを使用する

ミドルボックスのルーティングウィザードは Amazon Virtual Private Cloud Console で利用できます。

目次

ミドルボックスルーティングウィザードを使用したルートの作成

ミドルボックスのルーティングウィザードを使用してルートを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択し、続いて Actions (アクション)、Manage middlebox routes (ミドルボックスのルートを管理) を選択します。
4. [Create routes] (ルートの作成) を選択します。
5. [Specify routes] (詳細の指定) ページで、以下の作業を行います。
   - [Source] (送信元) で、トラフィックの送信元を選択します。仮想プライベートゲートウェイを選択した場合は、[Destination IPv4 CIDR] (送信先 IPv4 CIDR) に、仮想プライベートゲートウェイから VPC に入るオンプレミストラフィックの CIDR を入力します。
   - [Middlebox] (ミドルボックスボックス) で、ミドルボックスアプライアンスに関連付けられているネットワークインターフェイス ID を選択します。また、Gateway Load Balancer エンドポイントを使用する場合は、VPC エンドポイント ID を選択します。
   - [Destination subnet] (送信先サブネット) で、送信先サブネットを選択します。
6. (オプション) 別の送信先サブネットを追加するには、[Add additional subnet] (サブネットの追加) で、次の作業を行います。
   - [Middlebox] (ミドルボックスボックス) で、ミドルボックスアプライアンスに関連付けられているネットワークインターフェイス ID を選択します。また、Gateway Load Balancer エンドポイントを使用する場合は、VPC エンドポイント ID を選択します。

     複数のサブネットに同じミドルボックスアプライアンスを使用する必要があります。
   - [Destination subnet] (送信先サブネット) で、送信先サブネットを選択します。
7. (オプション) 別の送信元を追加するには、[Add source] (送信元の追加) をクリックし、前の手順を繰り返します。
8. [Next (次へ)] を選択します。
9. Review and create] (確認と作成) ページで、ルートを確認し、[Create routes (ルートの作成) を選択します。

ミドルボックスルートの変更

ゲートウェイ、ミドルボックス、または送信先サブネットを変更することで、ルート設定を編集できます。

変更を加えると、ミドルボックスルーティングウィザードは自動的に以下の操作を実行します。

- ゲートウェイ、ミドルボックス、送信先サブネットの新しいルートテーブルを作成します。
- 必要なルートを新しいルートテーブルに追加します。
- ミドルボックスルーティングウィザードがリソースに関連付けた現在のルートテーブルの関連付けを解除します。
- ミドルボックスルーティングウィザードで作成された新しいルートテーブルをリソースに関連付けます。

ミドルボックスルーティングウィザードを使用してミドルボックスルートを変更するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択し、続いて Actions (アクション)、Manage middlebox routes (ミドルボックスのルートを管理) を選択します。
4. [Export routes] (ルートのエクスポート) を選択します。
5. ゲートウェイを変更するには、[Source] (送信元) で、トラフィックが VPC に入るゲートウェイを選択します。仮想プライベートゲートウェイを選択した場合は、[Destination IPv4 CIDR (送信先 IPv4 CIDR) に、送信先サブネットの CIDR を入力します。
6. 別の送信先サブネットを追加するには、[Add additional subnet] (サブネットの追加) で、次の作業を行います。
   - [Middlebox] (ミドルボックスボックス) で、ミドルボックスアプライアンスに関連付けられているネットワークインターフェイス ID を選択します。また、Gateway Load Balancer エンドポイントを使用する場合は、VPC エンドポイント ID を選択します。

     複数のサブネットに同じミドルボックスアプライアンスを使用する必要があります。
   - [Destination subnet] (送信先サブネット) で、送信先サブネットを選択します。
7. [Next (次へ)] を選択します。
8. リポジトリの [Review and update] (確認と更新) ページには、ミドルボックスルーティングウィザードで作成されるルートテーブルとそのルートのリストが表示されます。ルートを確認し、確認ダイアログボックスで、[Update routes] (ルートの更新) を選択します。

ミドルボックスルーティングウィザードのルートテーブルを表示する

ミドルボックスルーティングウィザードのルートテーブルを表示するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. VPC を選択し、続いて Actions (アクション)、Manage middlebox routes (ミドルボックスのルートを管理) を選択します。
4. [Middlebox route tables] (ミドルボックスルートテーブル) で、この数字はミドルボックスルーティングウィザードが作成したルート数を示します。ルート表示の番号を選択します。

ミドルボックスルーティングウィザードのルートは、別のルートテーブルページに表示されます。

ミドルボックスルーティングウィザード設定を削除する

ミドルボックスルーティングウィザードの設定が不要になった場合は、ルートテーブルを手動で削除してください。

ミドルボックスルーティングウィザードの設定を削除するには

1. ミドルボックスルーティングウィザードのルートテーブルを表示します。詳細については、「the section called “ミドルボックスルーティングウィザードのルートテーブルを表示する” (p. 322)」を参照してください。

   操作を実行すると、ミドルボックスルーティングウィザードで作成したルートテーブルが別のルートテーブルページに表示されます。
2. 表示される各ルートテーブルを削除します。詳細については、「the section called “ルートテーブルを削除する” (p. 320)」を参照してください。

ミドルボックスルーティングウィザードに関する考慮事項

ミドルボックスルーティングウィザードを使用する場合は、次の点に注意してください。

- トラフィックを検査する場合は、送信元のインターネットゲートウェイまたは仮想プライベートゲートウェイを使用できます。
- 同じ VPC 内の複数のミドルボックス設定で同じミドルボックスを使用する場合は、ミドルボックスが両方のサブネットで同じホップ位置にあることを確認してください。
- アプライアンスは、送信元または送信先サブネットとは別のサブネットで構成する必要があります。
- アプライアンスでの送信元/送信先のチェックを無効にする必要があります。詳細については、Linux インスタンス用 Amazon EC2 ユーザーガイドの「送信元または送信先チェックの変更」を参照してください。
- ミドルボックスルーティングウィザードで作成したルートテーブルとルートは、クォータに対してカウントされます。詳細については、「the section called “ルートテーブル” (p. 364)」を参照してください。
- ネットワークインターフェイスなどのリソースを削除すると、リソースとのルートテーブルの関連付けが削除されます。リソースがターゲットである場合、ルート削除はブラックホールに設定されます。ルートテーブルは削除されません。
- ミドルボックスサブネットと送信先サブネットは、デフォルト以外のルートテーブルに関連付ける必要があります。

  Note

  ミドルボックスルーティングウィザードを使用して作成したルートテーブルを変更または削除するには、ミドルボックスルーティングウィザードを使うことをお勧めします。

関連情報

ミドルボックスルーティングウィザードで使用するリソースを作成する方法の詳細については、以下を参照してください。

- VPC とサブネット (p. 106)
- the section called “インターネットゲートウェイ” (p. 221)
- the section called “ネットワークインターフェイス” (p. 286)
- Gateway Load Balancer エンドポイント (AWS PrivateLink)
- Elastic Load Balancing - Gateway Load Balancer

VPC ピアリング接続

VPC ピアリング接続は、2 つの VPC 間でプライベートなトラフィックのルーティングを可能にするネットワーキング接続です。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、自分の VPC 間、別の AWS アカウントの VPC との間、または別の AWS リージョンの VPC との間に作成できます。

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 AWS Site-to-Site VPN 接続でもなく、別個の物理ハードウェアにも依存しません。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 ピアリング接続の詳細および VPC ピアリング接続を使用できるシナリオの例については、Amazon VPC ピア機能ガイドを参照してください。

VPC フローログ

VPC フローログは、VPC のネットワークインターフェイスとの間で行き来する IP トラフィックに関する情報をキャプチャできるようにする機能です。フローログデータは Amazon CloudWatch Logs または Amazon S3 に発行できます。フローログを作成すると、選択した送信先でそのデータを取得して表示できます。

フローログは、以下のような多くのタスクに役立ちます。

- 制限の過度に厳しいセキュリティグループルールを診断する
- インスタンスに到達するトラフィックをモニタリングする
- ネットワークインターフェイスに出入りするトラフィックの方向を決定する

フローログデータはネットワークトラフィックのパスの外で収集されるため、ネットワークのスループットやレイテンシーには影響しません。ネットワークパフォーマンスに影響を与えるリスクなしに、フローログを作成または削除できます。

目次

フローログの基礎

VPC、サブネット、またはネットワークインターフェイスのフローログを作成できます。サブネットまたは VPC のフローログを作成する場合、そのサブネットまたは VPC 内の各ネットワークインターフェイスがモニタリングされます。

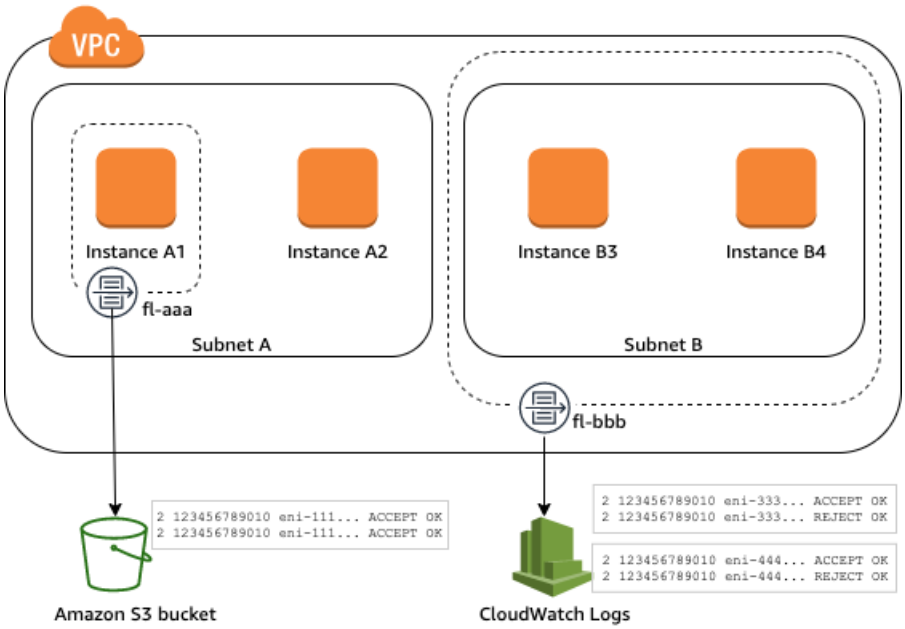
モニタリングされるネットワークインターフェイスのフローログデータは、フローログレコードとして記録されます。これは、トラフィックフローについて説明するフィールドで構成されるログイベントです。詳細については、「」を参照してくださいフローログレコード (p. 327)

フローログを作成するには、以下の内容を指定します。

- フローログを作成するリソース
- キャプチャするトラフィックの種類 (許可されたトラフィック、拒否されたトラフィック、またはすべてのトラフィック)
- フローログデータを発行する送信先

次の例では、インスタンス A1 のネットワークインターフェイスで承諾されたトラフィックをキャプチャし、フローログレコードを Amazon S3 バケットに発行するフローログ (`fl-aaa`) を作成します。サブネット B のすべてのトラフィックをキャプチャし、フローログレコードを Amazon CloudWatch Logs に発行す

る 2 番目のフローログを作成します。フローログ (fl-bbb) は、サブネット B のすべてのネットワークインターフェイスのトラフィックをキャプチャします。インスタンス A2 のネットワークインターフェイスのトラフィックをキャプチャするフローログはありません。



フローログを作成した後で、データの収集と選択された送信先への発行が開始されるまでに数分かかる場合があります。フローログで、ネットワークインターフェイスのリアルタイムのログストリームはキャプチャされません。詳細については、「」を参照してくださいフローログの作成 (p. 349)

サブネットまたは VPC のフローログを作成した後で、サブネットにさらに多くのインスタンスを起動する場合、新しいネットワークインターフェイスごとに新しいログストリーム (CloudWatch Logs の場合) またはログファイルオブジェクト (Amazon S3 の場合) が作成されます。これは、そのネットワークインターフェイス用にネットワークトラフィックが記録されるとすぐに行われます。

他の AWS サービスによって作成されたネットワークインターフェイスのフローログを作成できます。例えば、次のとおりです。

- Elastic Load Balancing
- Amazon RDS
- Amazon ElastiCache
- Amazon Redshift
- Amazon WorkSpaces
- NAT ゲートウェイ
- トランジットゲートウェイ

ネットワークインターフェイスの種類にかかわらず、Amazon EC2 コンソールまたは Amazon EC2 API を使用してネットワークインターフェイスのフローログを作成する必要があります。

フローログにタグを適用できます。タグはそれぞれ、1 つのキーとオプションの 1 つの値で構成されており、どちらもお客様側が定義します。タグは、目的や所有者などによって、フローログを整理するのに役立ちます。

フローログが不要になった場合には、それを削除することができます。フローログを削除すると、リソースのフローログサービスは無効になり、新しいフローログレコードは作成されず、CloudWatch Logs または Amazon S3 にも発行されません。フローログを削除しても、ネットワークインターフェイスの既

存のフローログレコードやログストリーム (CloudWatch Logs の場合) またはログファイルオブジェクト (Amazon S3 の場合) は削除されません。既存のログストリームを削除するには、CloudWatch Logs コンソールを使用します。既存のログファイルオブジェクトを削除するには、Amazon S3 コンソールを使用します。フローログを削除した後で、データの収集が中止するまでに数分かかる場合があります。詳細については、「」を参照してくださいフローログの削除 (p. 351)

フローログレコード

フローログレコードは、VPC のネットワークの流れを表します。デフォルトでは、各レコードは、集約間隔 (キャプチャウィンドウとも呼ばれる) 内で発生するネットワークインターネットプロトコル (IP) トラフィックフロー (ネットワークインターフェイスごとに 5 タプルによって特徴付けられる) をキャプチャします。

各レコードは、スペースで区切られたフィールドから成る文字列です。送信元、送信先、プロトコルなど、レコードには IP フローのさまざまなコンポーネントの値が含まれています。

フローログを作成するときは、フローログレコードのデフォルトの形式を使用するか、カスタム形式を指定できます。

目次

集約間隔

集約間隔は、特定のフローがキャプチャされ、フローログレコードに集約される期間です。デフォルトでは、最大の集約間隔が 10 分に設定されています。フローログを作成する場合、オプションで最大集約間隔を 1 分に指定できます。最大集約間隔が 1 分のフローログでは、最大集約間隔が 10 分のフローログよりも多くのフローログレコードが生成されます。

ネットワークインターフェイスが Nitro ベースのインスタンスにアタッチされている場合、指定した最大集約間隔に関係なく、集約間隔は常に 1 分以下になります。

集約間隔内にデータが取得された後、データの処理および CloudWatch Logs または Amazon S3 へのパブリッシュにさらに時間がかかります。フローログサービスは、通常、約 5 分で CloudWatch Logs に、約 10 分で Amazon S3 にログを配信します。ただし、ログの配信はベストエフォートベースであり、通常の配信時間を超えてログが遅れる可能性があります。

デフォルトの形式

デフォルトの形式では、フローログレコードには、使用可能なフィールドテーブル (p. 328)に表示される順序でバージョン 2 のフィールドが含まれます 。デフォルトの形式をカスタマイズまたは変更することはできません。使用可能なすべてのフィールドまたはフィールドの異なるサブセットをキャプチャするには、代わりにカスタム形式を指定します。

カスタム形式

カスタム形式を使用して、フローログレコードに含めるフィールドと順序を指定します。これにより、ニーズに合ったフローログを作成し、関連のないフィールドを省略できます。カスタム形式を使用すると、発行されたフローログから特定の情報を抽出する別個のプロセスが不要になります。使用可能なフローログフィールドは任意の数指定できますが、少なくとも 1 つ指定する必要があります。

使用可能なフィールド

次に表に、フローログレコードの使用可能なすべてのフィールドを示します。[Version (バージョン)] 列には、フィールドが導入された VPC フローログのバージョンが表示されます。デフォルトの形式には、すべてのバージョン 2 フィールドが含まれ、順番はテーブルと同じです。

フィールドが特定のレコードに該当しないか、特定のレコードに対して計算できなかった場合、レコードでそのエントリには「-」記号が表示されます。パケットヘッダーから直接取得されないメタデータフィールドは、ベストエフォート近似値であり、値が欠落しているか、不正確である可能性があります。

フィールド	説明	バージョン
version	VPC フローログバージョン。デフォルトの形式を使用する場合、バージョンは 2 です。カスタム形式を使用する場合、そのバージョンは指定されたフィールドの中で最も高いバージョンです。例えば、バージョン 2 のフィールドのみを指定した場合、バージョンは 2 です。バージョン 2、3、4 のフィールドを混在させて指定すると、バージョンは 4 になります。	2
account-id	トラフィックが記録されるソースネットワークインターフェイスの所有者の AWS アカウント ID。ネットワークインターフェイスが AWS のサービスによって作成された場合 (VPC エンドポイントまたは Network Load Balancer の作成時など)、このフィールドに対してレコードに unknown と表示されることがあります。	2
interface-id	トラフィックが記録されるネットワークインターフェイスの ID。	2
srcaddr	受信トラフィックの送信元アドレスか、ネットワークインターフェイスにおける送信トラフィックのネットワークインターフェイスの IPv4 または IPv6 アドレス。ネットワークインターフェイスの IPv4 アドレスは常にそのプライベート IPv4 アドレスです。「pkt-srcaddr」も参照してください。	2
dstaddr	送信トラフィックの送信先アドレスか、ネットワークインターフェイスにおける受信トラフィックのネットワークインターフェイスの IPv4 または IPv6 アドレス。ネットワークインターフェイスの IPv4 アドレスは常にそのプライベート IPv4 アドレスです。「pkt-dstaddr」も参照してください。	2
srcport	トラフィックの送信元ポート。	2
dstport	トラフィックの送信先ポート。	2
protocol	トラフィックの IANA プロトコル番号。詳細については、「割り当てられたインターネットプロトコル番号」を参照してください。	2
packets	フロー中に転送されたパケットの数。	2
bytes	フロー中に転送されたバイト数。	2
start	集約間隔内にフローの最初のパケットが受信された時間 (UNIX 秒)。これは、パケットがネットワークインターフェイス上で送信または受信されてから最大 60 秒になる場合があります。	2
end	集約間隔内にフローの最後のパケットが受信された時間 (UNIX 秒)。これは、パケットがネットワークインターフェイス上で送信または受信されてから最大 60 秒になる場合があります。	2

フィールド	説明	バージョン
action	トラフィックに関連付けられたアクション: • ACCEPT — 記録されたトラフィックは、セキュリティグループおよびネットワーク ACL で許可されています。 • REJECT — 記録されたトラフィックは、セキュリティグループまたはネットワーク ACL で許可されていません。	2
log-status	フローログのロギングステータス。 • OK — データは選択された送信先に正常に記録されます。 • NODATA — 集約間隔内にネットワークインターフェイスとの間で行き来するネットワークトラフィックはありませんでした。 • SKIPDATA — 集約間隔内に一部のフローログレコードがスキップされました。これは、内部的なキャパシティー制限、または内部エラーが原因である可能性があります。	2
vpc-id	トラフィックが記録されるネットワークインターフェイスが含まれる VPC の ID。	3
subnet-id	トラフィックが記録されるネットワークインターフェイスが含まれるサブネットの ID。	3
instance-id	インスタンスをお客様が所有している場合、トラフィックが記録されるネットワークインターフェイスに関連するインスタンスの ID。リクエスタマネージド型のネットワークインターフェイス (NAT ゲートウェイのネットワークインターフェイスなど) の場合、「-」記号を返します。	3
tcp-flags	次の TCP フラグのビットマスク値: • SYN — 2 • SYN-ACK — 18 • FIN — 1 • RST — 4 ACK は、SYN に付随する場合のみ報告されます。 TCP フラグは、集約間隔内に OR 処理することができます。短い接続の場合、フラグがフローログレコードの同じ行に設定されることがあります (例えば、SYN-ACK と FIN の場合は 19、SYN と FIN の場合は 3 など)。例については、「TCP フラグシーケンス (p. 333)」を参照してください。	3
type	トラフィックの種類。指定できる値は次のとおりです: IPv4、IPv6、および EFA。Elastic Fabric Adapter (EFA) の詳細については、「Elastic Fabric Adapter」を参照してください。	3

フィールド	説明	バージョン
pkt-srcaddr	トラフィックのパケットレベルの (元の) 送信元 IP アドレス。srcaddr フィールドとともにこのフィールドを使用し、トラフィックが通過する中間レイヤーの IP アドレスとトラフィックの元の送信元 IP アドレスを区別します。例えば、トラフィックが NAT ゲートウェイのネットワークインターフェイス (p. 334)を通過する場合や、Amazon EKS 内のポッドの IP アドレスが、ポッドが実行されているインスタンスノードのネットワークインターフェイスの IP アドレスとは異なる場合などです (VPC 内の通信の場合)。	3
pkt-dstaddr	トラフィックのパケットレベルの (元の) 送信先 IP アドレス。dstaddr フィールドとともにこのフィールドを使用し、トラフィックが通過する中間レイヤーの IP アドレスとトラフィックの最終的な送信元 IP アドレスを区別します。例えば、トラフィックが NAT ゲートウェイのネットワークインターフェイス (p. 334)を通過する場合や、Amazon EKS 内のポッドの IP アドレスが、ポッドが実行されているインスタンスノードのネットワークインターフェイスの IP アドレスとは異なる場合などです (VPC 内の通信の場合)。	3
region	トラフィックが記録されるネットワークインターフェイスが含まれるリージョン。	4
az-id	トラフィックが記録されるネットワークインターフェイスが含まれるアベイラビリティーゾーンの ID。トラフィックがサブロケーションからの場合、レコードにはこのフィールドに「-」記号が表示されます。	4
sublocation-type	sublocation-id フィールドに返されるサブロケーションのタイプ。指定可能な値は次のとおりです: wavelength | outpost | localzone。トラフィックがサブロケーションからではない場合、レコードにはこのフィールドに「-」記号が表示されます。	4
sublocation-id	トラフィックが記録されるネットワークインターフェイスが含まれるサブロケーションの ID。トラフィックがサブロケーションからではない場合、レコードにはこのフィールドに「-」記号が表示されます。	4
pkt-src-aws-service	pkt-srcaddr フィールド用の IP アドレスの範囲のサブセットの名前 (送信元 IP アドレスが AWS のサービス用の場合)。指定可能な値は次のとおりです:AMAZON |AMAZON_APPFLOW |AMAZON_CONNECT | API_GATEWAY |CHIME_MEETINGS |CHIME_VOICECONNECTOR |CLOUD9 |CLOUDFRONT |CODEBUILD |DYNAMODB |EBS | EC2 |EC2_INSTANCE_CONNECT |GLOBALACCELERATOR | KINESIS_VIDEO_STREAMS |ROUTE53 |ROUTE53_HEALTHCHECKS |ROUTE53_HEALTHCHECKS_PUBLISHING |ROUTE53_RESOLVER | S3 |WORKSPACES_GATEWAYS。	5
pkt-dst-aws-service	pkt-dstaddr フィールド用の IP アドレスの範囲のサブセットの名前 (送信先 IP アドレスが AWS のサービス用の場合)。可能な値の一覧については、pkt-src-aws-service フィールドをご参照ください。	5
flow-direction	トラフィックがキャプチャされるインターフェイスに対するフローの方向。指定できる値は次のとおりです: ingress | egress。	5

フィールド	説明	バージョン
traffic-path	出力トラフィックが送信先につながるパス。トラフィックが出力トラフィックであるかどうかを判断するには、flow-direction フィールドを確認します。指定できる値は次のとおりです。いずれの値も適用されない場合、フィールドは - に設定されます。 • 1 — 同じ VPC 内の別のリソース経由 • 2 — インターネットゲートウェイまたはゲートウェイ VPC エンドポイント経由 • 3 — 仮想プライベートゲートウェイ経由 • 4 — リージョン内 VPC ピア接続経由 • 5 — リージョン間 VPC ピア接続経由 • 6 — ローカルゲートウェイ経由 • 7 — ゲートウェイ VPC エンドポイント経由 (Nitro ベースのインスタンスのみ) • 8 — インターネットゲートウェイ経由 (Nitro ベースのインスタンスのみ)	5

フローログレコードの例

特定のトラフィックフローをキャプチャするフローログレコードの例を以下に示します。

フローログレコード形式の詳細については、「フローログレコード (p. 327)」を参照してください。フローログの作成方法については、「フローログの使用 (p. 348)」をご参照ください。

目次

承認されたトラフィックと拒否されたトラフィック

デフォルトフローログレコードの例を以下に示します。

この例では、アカウント 123456789010 のネットワークインターフェイス eni-1235b8ca123456789 への SSH トラフィック (送信先ポート 22、TCP プロトコル) が許可されています。

```
2 123456789010 eni-1235b8ca123456789 172.31.16.139 172.31.16.21 20641 22 6 20 4249
 1418530010 1418530070 ACCEPT OK
```

この例では、アカウント 123456789010 のネットワークインターフェイス eni-1235b8ca123456789 への RDP トラフィック (送信先ポート 3389、TCP プロトコル) が拒否されています。

```
2 123456789010 eni-1235b8ca123456789 172.31.9.69 172.31.9.12 49761 3389 6 20 4249
 1418530010 1418530070 REJECT OK
```

データなしおよびスキップされたレコード

デフォルトフローログレコードの例を以下に示します。

この例では、集約間隔内にデータは記録されませんでした。

```
2 123456789010 eni-1235b8ca123456789 - - - - - - - 1431280876 1431280934 - NODATA
```

この例では、集約間隔内にレコードがスキップされました。VPC Flog Logs は、内部容量を超えるため、集約間隔でフローログデータをキャプチャできない場合に、レコードをスキップします。単一のスキップレコードは、集約間隔内にネットワークインターフェイスでキャプチャされなかった複数のフローを表すことができます。

```
2 123456789010 eni-11111111aaaaaaaaa - - - - - - - 1431280876 1431280934 - SKIPDATA
```

セキュリティグループとネットワーク ACL ルール

フローログを使用して過度に制限されているか制限のないセキュリティグループルールまたはネットワーク ACL ルールを診断している場合は、これらのリソースのステートフルさに注意してください。セキュリティグループはステートフルです。つまり、セキュリティグループのルールで許可されていない場合でも、許可されたトラフィックへの応答も許可されます。逆に、ネットワーク ACL はステートレスです。したがって、許可されたトラフィックへの応答は、ネットワーク ACL ルールに従って行われます。

例えば、ホームコンピュータ (IP アドレスが 203.0.113.12) からインスタンス (ネットワークインターフェイスのプライベート IP アドレスが 172.31.16.139) へは、ping コマンドを使用します。セキュリティルールのインバウンドルールでは ICMP トラフィックが許可されますが、アウトバウンドルールでは ICMP トラフィックが許可されません。セキュリティグループがステートフルの場合、インスタンスからのレスポンス ping が許可されます。ネットワーク ACL でインバウンド ICMP トラフィックが許可されますが、アウトバウンド ICMP トラフィックは許可されません。ネットワーク ACL はステートレスであるため、ping 応答は削除され、ホームコンピュータに達しません。デフォルトフローログで、これは 2 つのフローログレコードとして表示されます。

- ネットワーク ACL とセキュリティグループの両方で許可され、したがってインスタンスへの到達を許可された発信元の ping の ACCEPT レコード。
- ネットワーク ACL で拒否された応答 ping の REJECT レコード。

```
2 123456789010 eni-1235b8ca123456789 203.0.113.12 172.31.16.139 0 0 1 4 336 1432917027
 1432917142 ACCEPT OK
```

```
2 123456789010 eni-1235b8ca123456789 172.31.16.139 203.0.113.12 0 0 1 4 336 1432917094
 1432917142 REJECT OK
```

ネットワーク ACL でアウトバウンド ICMP トラフィックを許可している場合、フローログには 2 つの ACCEPT レコード (1 つは発信元の ping、もう 1 つは応答 ping) が表示されます。セキュリティグループがインバウンド ICMP トラフィックを拒否する場合、トラフィックに対してインスタンスへの到達が許可されなかったため、フローログには 1 つの REJECT レコードが表示されます。

IPv6 トラフィック

デフォルトフローログレコードの例を以下に示します。この例では、IPv6 アドレス 2001: db 8:1234: a 100:8 d6e: 3477: df66: f105 からアカウント 123456789010 のネットワークインターフェイス eni-1235b8ca123456789 への SSH トラフィック (ポート 22) が許可されています。

```
2 123456789010 eni-1235b8ca123456789 2001:db8:1234:a100:8d6e:3477:df66:f105
 2001:db8:1234:a102:3304:8879:34cf:4071 34892 22 6 54 8855 1477913708 1477913820 ACCEPT OK
```

TCP フラグシーケンス

次のフィールドを次の順序でキャプチャするカスタムフローログの例を以下に示します。

```
version vpc-id subnet-id instance-id interface-id account-id type srcaddr dstaddr srcport
 dstport pkt-srcaddr pkt-dstaddr protocol bytes packets start end action tcp-flags log-
status
```

tcp-flags フィールドは、トラフィックの方向 (接続を開始したサーバーなど) を識別するのに役立ちます。次のレコード (午後 7:47:55 PM に開始して午後 7:48:53 に終了) では、ポート 5001 で実行されているサーバーに対する接続がクライアントにより開始されています。クライアントの異なる送信元ポート (43416 および 43418) から送信された 2 つの SYN フラグ (2) をサーバーが受け取っています。SYN ごとに、サーバーから対応するポートのクライアント (18) に SYN-ACK が送信されています。

```
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456 eni-1235b8ca123456789
 123456789010 IPv4 52.213.180.42 10.0.0.62 43416 5001 52.213.180.42 10.0.0.62 6 568 8
 1566848875 1566848933 ACCEPT 2 OK
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456 eni-1235b8ca123456789
 123456789010 IPv4 10.0.0.62 52.213.180.42 5001 43416 10.0.0.62 52.213.180.42 6 376 7
 1566848875 1566848933 ACCEPT 18 OK
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456 eni-1235b8ca123456789
 123456789010 IPv4 52.213.180.42 10.0.0.62 43418 5001 52.213.180.42 10.0.0.62 6 100701 70
 1566848875 1566848933 ACCEPT 2 OK
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456 eni-1235b8ca123456789
 123456789010 IPv4 10.0.0.62 52.213.180.42 5001 43418 10.0.0.62 52.213.180.42 6 632 12
 1566848875 1566848933 ACCEPT 18 OK
```

2 つ目の集約間隔では、前のフローで確立された接続の 1 つが閉じられます。クライアントは、ポート 43418 での接続に対してサーバーに FIN フラグ (1) を送信しています。サーバーは、クライアントのポート 43418 に FIN を送信しています。

```
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456 eni-1235b8ca123456789
 123456789010 IPv4 10.0.0.62 52.213.180.42 5001 43418 10.0.0.62 52.213.180.42 6 63388 1219
 1566848933 1566849113 ACCEPT 1 OK
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456 eni-1235b8ca123456789
 123456789010 IPv4 52.213.180.42 10.0.0.62 43418 5001 52.213.180.42 10.0.0.62 6 23294588
 15774 1566848933 1566849113 ACCEPT 1 OK
```

単一の集約間隔内で開かれて閉じられた短い接続の場合 (数秒など)、同じ方向のトラフィックフローに関して、フローログレコードの同じ行にフラグが設定されることがあります。次の例では、同じ集約間隔内で接続が確立および終了されています。1 行目では、TCP フラグ値が 3 です。これは、SYN と FIN メッセージがクライアントからサーバーに送信されたことを示しています。2 行目では、TCP フラグ値が 19 です。これは、SYN-ACK と FIN メッセージがサーバーからクライアントに送信されたことを示しています。

```
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456 eni-1235b8ca123456789
 123456789010 IPv4 52.213.180.42 10.0.0.62 43638 5001 52.213.180.42 10.0.0.62 6 1260 17
 1566933133 1566933193 ACCEPT 3 OK
```

```
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456 eni-1235b8ca123456789
 123456789010 IPv4 10.0.0.62 52.213.180.42 5001 43638  10.0.0.62 52.213.180.42 6 967 14
 1566933133 1566933193 ACCEPT 19 OK
```

NAT ゲートウェイ経由のトラフィック

この例では、プライベートサブネットのインスタンスが、パブリックサブネットにある NAT ゲートウェイ経由でインターネットに接続しています。

NAT ゲートウェイネットワークインターフェイスの次のカスタムフローログでは、次のフィールドが次の順序でキャプチャされています。

```
instance-id interface-id srcaddr dstaddr pkt-srcaddr pkt-dstaddr
```

フローログには、インスタンス IP アドレス (10.0.1.5) から NAT ゲートウェイネットワークインターフェイス経由でインターネット上のホスト (203.0.113.5) に送信されるトラフィックのフローを示しています。NAT ゲートウェイネットワークインターフェイスは、リクエスタが管理するネットワークインターフェイスのため、フローログレコードの instance-id フィールドには「-」記号が表示されます。次の行は、送信元インスタンスから NAT ゲートウェイネットワークインターフェイスへのトラフィックを示しています。dstaddr フィールドと pkt-dstaddr フィールドの値は異なります。dstaddr フィールドには、NAT ゲートウェイネットワークインターフェイスのプライベート IP アドレスが表示されており、pkt-dstaddr フィールドにはインターネット上のホストの最終的な送信先 IP アドレスが表示されています。

```
- eni-1235b8ca123456789 10.0.1.5 10.0.0.220 10.0.1.5 203.0.113.5
```

次の 2 行は、NAT ゲートウェイトラフィックインターフェイスからインターネット上の送信先ホストへのトラフィックと、ホストから NAT ゲートウェイネットワークインターフェイスへのレスポンストラフィックを示しています。

```
- eni-1235b8ca123456789 10.0.0.220 203.0.113.5 10.0.0.220 203.0.113.5
- eni-1235b8ca123456789 203.0.113.5 10.0.0.220 203.0.113.5 10.0.0.220
```

次の行は、NAT ゲートウェイネットワークインターフェイスから送信元インスタンスへのトラフィックを示しています。srcaddr フィールドと pkt-srcaddr フィールドの値は異なります。srcaddr フィールドには、NAT ゲートウェイネットワークインターフェイスのプライベート IP アドレスが表示されており、pkt-srcaddr フィールドにはインターネット上のホストの IP アドレスが表示されています。

```
- eni-1235b8ca123456789 10.0.0.220 10.0.1.5 203.0.113.5 10.0.1.5
```

上記と同じフィールドセットを使用して別のカスタムフローログを作成できます。プライベートサブネット内のインスタンスのネットワークインターフェイスのフローログを作成します。この場合、instance-id フィールドはネットワークインターフェイスに関連するインスタンスの ID を返します。dstaddr および pkt-dstaddr フィールドと srcaddr および pkt-srcaddr フィールドの間に差異はありません。NAT ゲートウェイのネットワークインターフェイスとは異なり、このネットワークインターフェイスはトラフィックの中間ネットワークではありません。

```
i-01234567890123456 eni-1111aaaa2222bbbb3 10.0.1.5 203.0.113.5 10.0.1.5 203.0.113.5
 #Traffic from the source instance to host on the internet
i-01234567890123456 eni-1111aaaa2222bbbb3 203.0.113.5 10.0.1.5 203.0.113.5 10.0.1.5
 #Response traffic from host on the internet to the source instance
```

転送ゲートウェイ経由のトラフィック

この例では、VPC A 内のクライアントがトランジットゲートウェイ経由で VPC B 内のウェブサーバーに接続します。クライアントとサーバーは、異なるアベイラビリティーゾーンにあります。したがって、eni-1111111111111111 を使用するとトラフィックは VPC B 内のサーバーで受信され、eni-22222222222222222 を使用すると VPC B から送信されます。

VPC B のカスタムフローログは、次の形式で作成できます。

```
version interface-id account-id vpc-id subnet-id instance-id srcaddr dstaddr srcport
 dstport protocol tcp-flags type pkt-srcaddr pkt-dstaddr action log-status
```

フローログレコードの次の行は、ウェブサーバーのネットワークインターフェイスにあるトラフィックのフローを示しています。1 行目は、クライアントからのリクエストトラフィックであり、最後の行はウェブサーバーからのレスポンストラフィックです。

```
3 eni-33333333333333333 123456789010 vpc-abcdefab012345678 subnet-22222222bbbbbbbbb
 i-01234567890123456 10.20.33.164 10.40.2.236 39812 80 6 3 IPv4 10.20.33.164 10.40.2.236
 ACCEPT OK
...
3 eni-33333333333333333 123456789010 vpc-abcdefab012345678 subnet-22222222bbbbbbbbb
 i-01234567890123456 10.40.2.236 10.20.33.164 80 39812 6 19 IPv4 10.40.2.236 10.20.33.164
 ACCEPT OK
```

次の行は eni-11111111111111111 (サブネット subnet-11111111aaaaaaaaa にあるトランジットゲートウェイのリクエスタマネージド型のネットワークインターフェイス) 上のリクエストトラフィックです。したがって、フローログレコードの instance-id フィールドには「-」記号が表示されます。srcaddr フィールドには、トランジットゲートウェイネットワークインターフェイスのプライベート IP アドレスが表示されており、pkt-srcaddr フィールドには VPC A 上のクライアントの送信元 IP アドレスが表示されています。

```
3 eni-11111111111111111 123456789010 vpc-abcdefab012345678 subnet-11111111aaaaaaaaa -
 10.40.1.175 10.40.2.236 39812 80 6 3 IPv4 10.20.33.164 10.40.2.236 ACCEPT OK
```

次の行は eni-22222222222222222 (サブネット subnet-22222222bbbbbbbbb にあるトランジットゲートウェイのリクエスタマネージド型のネットワークインターフェイス) 上のレスポンストラフィックです。dstaddr フィールドには、トランジットゲートウェイネットワークインターフェイスのプライベート IP アドレスが表示されており、pkt-dstaddr フィールドには VPC A 上のクライアントの IP アドレスが表示されています。

```
3 eni-22222222222222222 123456789010 vpc-abcdefab012345678 subnet-22222222bbbbbbbbb -
 10.40.2.236 10.40.2.31 80 39812 6 19 IPv4 10.40.2.236 10.20.33.164 ACCEPT OK
```

サービス名、トラフィックパス、およびフロー方向

カスタムフローログレコードのフィールドの例を次に示します。

```
version srcaddr dstaddr srcport dstport protocol start end type packets bytes account-id
 vpc-id subnet-id instance-id interface-id region az-id sublocation-type sublocation-id
 action tcp-flags pkt-srcaddr pkt-dstaddr pkt-src-aws-service pkt-dst-aws-service traffic-
path flow-direction log-status
```

次の例では、レコードにバージョン 5 フィールドが含まれているので、バージョンは 5 です。EC2 インスタンスは Amazon S3 サービスを呼び出します。フローログは、インスタンスのネットワークインター

フェイスでキャプチャされます。最初のレコードのフロー方向は ingress で、2 番目のレコードのフロー方向は egress です。egressレコードの場合、traffic-path は 8 で、トラフィックがインターネットゲートウェイを通過することを示します。traffic-path フィールドは、ingress トラフィックではサポートされません。pkt-srcaddr または pkt-dstaddr がパブリック IP アドレスの場合は、サービス名が表示されます。

```
5 52.95.128.179 10.0.0.71 80 34210 6 1616729292 1616729349 IPv4 14 15044 123456789012 vpc-
abcdefab012345678 subnet-aaaaaaaa012345678 i-0c50d5961bcb2d47b eni-1235b8ca123456789 ap-
southeast-2 apse2-az3 - - ACCEPT 19 52.95.128.179 10.0.0.71 S3 - - ingress OK
5 10.0.0.71 52.95.128.179 34210 80 6 1616729292 1616729349 IPv4 7 471 123456789012 vpc-
abcdefab012345678 subnet-aaaaaaaa012345678 i-0c50d5961bcb2d47b eni-1235b8ca123456789 ap-
southeast-2 apse2-az3 - - ACCEPT 3 10.0.0.71 52.95.128.179 - S3 8 egress OK
```

フローログの制限事項

フローログを使用するには、次の制限事項に注意する必要があります。

- EC2-Classic プラットフォームにあるネットワークインターフェイスのフローログを有効にすることはできません。これには、ClassicLink を使用して VPC にリンクされた EC2-Classic インスタンスが含まれます。
- ピア VPC がアカウントにない限り、VPC とピアリング接続された VPC のフローログを有効にすることはできません。
- フローログを作成すると、その設定やフローログレコードの形式を変更することはできません。例えば、異なる IAM ロールをフローログに関連付けたり、フローログレコードのフィールドを追加または削除したりすることはできません。代わりにフローログを削除し、必要な設定で新しいログを作成できます。
- ネットワークインターフェイスに複数の IPv4 アドレスがある場合、トラフィックがセカンダリプライベート IPv4 アドレスに送信されても、フローログの `dstaddr` フィールドにはプライマリプライベート IPv4 アドレスが表示されます。元の送信先 IP アドレスをキャプチャするには、`pkt-dstaddr` フィールドを含むフローログを作成します。
- トラフィックがネットワークインターフェイスに送信され、送信先がネットワークインターフェイスの IP アドレスのいずれでもない場合、フローログの `dstaddr` フィールドにはプライマリプライベート IPv4 アドレスが表示されます。元の送信先 IP アドレスをキャプチャするには、`pkt-dstaddr` フィールドを含むフローログを作成します。
- トラフィックがネットワークインターフェイスから送信され、送信元がネットワークインターフェイスの IP アドレスのいずれでもない場合、フローログの `srcaddr` フィールドにはプライマリプライベート IPv4 アドレスが表示されます。元の送信元 IP アドレスをキャプチャするには、`pkt-srcaddr` フィールドを含むフローログを作成します。
- ネットワークインターフェイスとの間でトラフィックが送受信される場合、パケットの送信元または送信先にかかわらず、フローログの `srcaddr` フィールドと `dstaddr` フィールドには常にプライマリのプライベート IPv4 アドレスが表示されます。パケットの送信元または送信先をキャプチャするには、`pkt-srcaddr` フィールドと `pkt-dstaddr` フィールドを含むフローログを作成します。
- ネットワークインターフェイスが Nitro ベースのインスタンスにアタッチされている場合、指定した最大集約間隔に関係なく、集約間隔は常に 1 分以下になります。

フローログですべての IP トラフィックはキャプチャされません。以下のトラフィックの種類は記録されません。

- Amazon DNS サーバーに接続したときにインスタンスによって生成されるトラフィック。独自の DNS サーバーを使用する場合は、その DNS サーバーへのすべてのトラフィックが記録されます。
- Amazon Windows ライセンスのアクティベーション用に Windows インスタンスによって生成されたトラフィック。
- インスタンスメタデータ用に `169.254.169.254` との間を行き来するトラフィック。

- Amazon Time Sync Service の 169.254.169.123 との間でやり取りされるトラフィック。
- DHCP トラフィック。
- ミラートラフィック。
- デフォルト VPC ルーターの予約済み IP アドレスへのトラフィック。
- エンドポイントのネットワークインターフェイスと Network Load Balancer のネットワークインターフェイスの間のトラフィック。

フローログの料金

フローログを CloudWatch Logs または Amazon S3 に発行すると、提供されたログに対するデータの取り込み料金とアーカイブ料金が適用されます。詳細と例については、「Amazon CloudWatch の料金」を参照してください。

Amazon S3 バケットへのフローログの発行に伴う料金を追跡するには、コスト配分タグをフローログサブスクリプションに適用できます。CloudWatch Logs へのフローログの発行に伴う料金を追跡するには、コスト配分タグを送信先の CloudWatch Logs ロググループに適用できます。これにより、AWS コスト配分レポートに、これらのタグで集計された使用量とコストが表示されます。ビジネスカテゴリ (コストセンター、アプリケーション名、所有者など) 別のタグを適用すると、コストを分類できます。詳細については、AWS Billing and Cost Management の「コスト配分タグの使用」を参照してください。

CloudWatch Logs へのフローログの発行

フローログはフローログデータを直接 Amazon CloudWatch に発行できます。フローログデータは、CloudWatch Logs に対して発行するときはロググループに発行され、各ネットワークインターフェイスにはロググループに一意のログストリームがあります。ログストリームにはフローログレコードが含まれます。同じロググループにデータを公開する複数のフローログを作成できます。同じネットワークインターフェイスが同じロググループの 1 つ以上のフローログに存在する場合、1 つの組み合わされたログストリームがあります。1 つのフローログで、拒否されたトラフィックをキャプチャし、別のフローログで、許可されたトラフィックをキャプチャするよう指定した場合、組み合わされたログストリームですべてのトラフィックがキャプチャされます。

フローログを CloudWatch Logs に発行すると、提供されたログに対するデータの取り込み料金とアーカイブ料金が適用されます。詳細については、「Amazon CloudWatch の料金」を参照してください。

CloudWatch Logs では、[timestamp] フィールドはフローログレコードでキャプチャされた開始時刻に対応します。[ingestionTime] フィールドは、CloudWatch Logs によってフローログレコードが受信された日時を示します。このタイムスタンプは、フローログレコードでキャプチャされた終了時刻より後です。

目次

CloudWatch Logs へのフローログ発行のための IAM ロール

フローログに関連付けられた IAM ロールには、CloudWatch Logs の指定されたロググループにフローログを発行するために十分なアクセス許可が必要です。IAM ロールは AWS アカウントに属している必要があります。

IAM ロールにアタッチされた IAM ポリシーには、少なくとも以下のアクセス許可が含まれている必要があります。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

フローログサービスがロールを引き受けることができる信頼関係がロールにあることも確認します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "vpc-flow-logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

既存ロールを更新するか、次の手順を使用してフローログで使用する新しいロールを作成できます。

フローログロールの作成

フローログの IAM ロールを作成するには

1. IAM コンソール (https://console.aws.amazon.com/iam/) を開きます。
2. ナビゲーションペインで [Roles]、[Create role] の順に選択します。
3. [Select type of trusted entity (信頼されたエンティティのタイプの選択)] で、[AWS のサービス] を選択します。[ユースケース] で、[EC2] を選択します。[Next: Permissions (次へ: アクセス許可)] を選択します。
4. [アクセス権限ポリシーをアタッチする] ページで、[Next: Review (次へ: レビュー)] を選択し、オプションでタグを追加します。[Next: Review] を選択します。
5. ロールの名前を入力し、オプションで説明を入力します。[ロールの作成] を選択します。
6. ロールの名前を選択します。[アクセス許可] で [インラインポリシーの追加]、[JSON] の順に選択します。
7. 「CloudWatch Logs へのフローログ発行のための IAM ロール (p. 337)」から最初のポリシーをコピーして、ウィンドウに貼り付けます。[ポリシーの確認] を選択します。
8. ポリシーの名前を入力し、[ポリシーの作成] を選択します。
9. ロールの名前を選択します。[Trust relationships] で、[Edit trust relationship] を選択します。既存のポリシードキュメントで、サービスを `ec2.amazonaws.com` から `vpc-flow-logs.amazonaws.com` に変更します。[Update Trust Policy] を選択します。

10. [Summary] ページで、ロールの ARN を書き留めます。フローログを作成するときに、この ARN が必要になります。

IAM ユーザーがロールを渡すためのアクセス許可

フローログに関連付けられた IAM ロール用に `iam:PassRole` アクションを使用するアクセス許可もユーザーに必要です。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["iam:PassRole"],
      "Resource": "arn:aws:iam::account-id:role/flow-log-role-name"
    }
  ]
}
```

CloudWatch Logs に発行するフローログの作成

VPCs、サブネット、またはネットワークインターフェイスのフローログを作成できます。これらのステップを IAM ユーザーとして実行する場合は、`iam:PassRole` アクションを使用するアクセス許可があることを確認してください。詳細については、「」を参照してくださいIAM ユーザーがロールを渡すためのアクセス許可 (p. 339)

Prerequisite

送信先ロググループを作成します。CloudWatch コンソールで [Log groups] (ロググループ) のページを開き、[Create log group] (ロググループの作成) を選択します。ロググループの名前を入力し、[Create] (作成) を選択します。

コンソールを使用してネットワークインターフェイスのフローログを作成するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. 1 つ以上のネットワークインターフェイスのチェックボックスにチェックを入れ、[Actions] (アクション)、[Create flow log] (フローログの作成) の順に選択します。
4. [Filter] (フィルター) で、ログに記録するトラフィックの種類を指定します。承認および拒否されたトラフィックを記録するには [All] (すべて)、拒否されたトラフィックだけをログ記録するには [Reject] (拒否)、承認されたトラフィックだけをログ記録するには [Accept] (承認) を選択します。
5. [Maximum aggregation interval] で、フローがキャプチャされ、1 つのフローログレコードに集約される最大期間を選択します。
6. [送信先] で、[Send to CloudWatch Logs (CloudWatch ログへの送信)] を選択します。
7. [Destination log group] (送信先ロググループ) で、作成した送信先ロググループの名前を選択します。
8. [IAM ロール] で、ログを CloudWatch Logs に発行できるアクセス許可があるロールの名前を指定します。
9. [Log record format] (ログレコードの形式) で、フローログレコードの形式を選択します。
   - デフォルトの形式を使用するには、[AWS default format] (AWS のデフォルトの形式) を選択します。
   - カスタム形式を使用するには、[Custom format] (カスタム形式) を選択し、[Log format] (ログ形式) からフィールドを選択します。

- デフォルトのフィールドを含むカスタムフローログを作成するには、まず [AWS default format] (AWS のデフォルトの形式) を選択して [Format preview] (形式のプレビュー) のフィールドをコピーし、[Custom format] (カスタム形式) を選択してテキストボックスにフィールドを貼り付けます。

10. (オプション) フローログにタグを適用するには、[Add new tag] (新規タグを追加) を選択します。
11. [フローログの作成] を選択します。

コンソールを使用して VPC またはサブネットのフローログを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Your VPCs] (VPC) または [Subnets] (サブネット) を選択します。
3. 1 つ以上の VPC またはサブネットのチェックボックスにチェックを入れ、[Actions] (アクション)、[Create flow log] (フローログの作成) の順に選択します。
4. [Filter] (フィルター) で、ログに記録するトラフィックの種類を指定します。承認および拒否されたトラフィックを記録するには [All] (すべて)、拒否されたトラフィックだけをログ記録するには [Reject] (拒否)、承認されたトラフィックだけをログ記録するには [Accept] (承認) を選択します。
5. [Maximum aggregation interval] で、フローがキャプチャされ、1 つのフローログレコードに集約される最大期間を選択します。
6. [送信先] で、[Send to CloudWatch Logs (CloudWatch ログへの送信)] を選択します。
7. [Destination log group] (送信先ロググループ) で、作成した送信先ロググループの名前を選択します。
8. [IAM ロール] で、ログを CloudWatch Logs に発行できるアクセス許可があるロールの名前を指定します。
9. [Log record format] (ログレコードの形式) で、フローログレコードの形式を選択します。
   - デフォルトの形式を使用するには、[AWS default format] (AWS のデフォルトの形式) を選択します。
   - カスタム形式を使用するには、[Custom format] (カスタム形式) を選択し、[Log format] (ログ形式) からフィールドを選択します。
   - デフォルトのフィールドを含むカスタムフローログを作成するには、まず [AWS default format] (AWS のデフォルトの形式) を選択して [Format preview] (形式のプレビュー) のフィールドをコピーし、[Custom format] (カスタム形式) を選択してテキストボックスにフィールドを貼り付けます。
10. (オプション) フローログにタグを適用するには、[Add new tag] (新規タグを追加) を選択します。
11. [フローログの作成] を選択します。

コマンドラインを使用してフローログを作成するには

以下のいずれかのコマンドを使用します。

- create-flow-logs (AWS CLI)
- New-EC2FlowLogs (AWS Tools for Windows PowerShell)
- CreateFlowLogs (Amazon EC2 Query API)

次の AWS CLI の例では、サブネット `subnet-1a2b3c4d` の許可されたすべてのトラフィックをキャプチャするフローログが作成されます。フローログは、IAM ロール `my-flow-logs` を使用し、アカウント 123456789101 内で、`publishFlowLogs` と呼ばれる CloudWatch Logs 内のロググループに配信されます。

```
aws ec2 create-flow-logs --resource-type Subnet --resource-ids subnet-1a2b3c4d --
traffic-type ACCEPT --log-group-name my-flow-logs --deliver-logs-permission-arn
 arn:aws:iam::123456789101:role/publishFlowLogs
```

CloudWatch Logs でのフローログレコードの処理

CloudWatch Logs で収集された他のログイベントのように、フローログレコードを操作できます。ログデータとメトリクスフィルタのモニタリングの詳細については、Amazon CloudWatch ユーザーガイドの「ログデータの検索およびフィルタリング」を参照してください。

例: フローログの CloudWatch メトリクスフィルタとアラームの作成

この例では、`eni-1a2b3c4d` のフローログがあります。1 時間以内の期間に TCP ポート 22 (SSH) 経由でインスタンスに接続しようとする試みが 10 個以上拒否された場合に、アラームを作成するとします。最初に、アラームを作成するトラフィックのパターンと一致するメトリクスフィルタを作成する必要があります。次に、メトリクスフィルタのアラームを作成できます。

拒否された SSH トラフィックのメトリクスフィルタを作成し、フィルタのアラームを作成するには

1. CloudWatch コンソール (https://console.aws.amazon.com/cloudwatch/) を開きます。
2. ナビゲーションペインで、[Logs (ログ)]、[Log groups (ロググループ)] の順に選択します。
3. ロググループのチェックボックスをオンにしてから、[Actions」、[Create metric filter] を選択します。
4. [フィルターパターン] で、次のように入力します。

```
[version, account, eni, source, destination, srcport, destport="22", protocol="6",
 packets, bytes, windowstart, windowend, action="REJECT", flowlogstatus]
```

5. [テストするログデータの選択] で、ネットワークインターフェイスのログストリームを選択します。(オプション) フィルタパターンと一致するログデータの行を表示するには、[テストパターン] を選択します。準備ができたら、[次へ] を選択します。
6. フィルター名、メトリクス名前空間、およびメトリック名を入力します。メトリクス値を 1 に設定します。完了したら、[次へ] を選択し、その後 [Create metric filter] を選択します。
7. ナビゲーションペインで、[アラーム]、[すべてのアラーム] の順に選択します。
8. [アラームの作成] を選択します。
9. 作成したメトリクスフィルターの名前空間を選択します。

   新しいメトリクスがコンソールに表示されるまでに数分かかる場合があります。
10. 作成したメトリクス名を選択し、その後 [Select metric] を選択します。
11. アラームを以下のように設定して、[Next] (次へ) をクリックします。

    - [統計] で、[合計] を選択します。これにより、指定された期間のデータポイントの総数をキャプチャしていることを確認できます。
    - [期間] で、[1 時間] を選択します。
    - [Whenever] で、[Greater/Equal] を選択し、しきい値は「10」と入力します。
    - [追加設定]、[Datapoints to alarm] はデフォルトの「1」のままにしておきます。
12. [Notification] で、既存の SNS トピックを選択するか、[Create new topic] を選択して新しいトピックを作成します。[Next (次へ)] を選択します。
13. 次のページで、アラームの名前と説明を入力し、[次へ] を選択します。
14. アラームの設定が終わったら、[Create alarm] を選択します。

フローログを Amazon S3 に発行する

フローログはフローログデータを Amazon S3 に発行できます。Amazon S3 に発行した場合、フローログデータは、指定する既存の Amazon S3 バケットに発行されます。モニタリングされるすべてのネットワークインターフェイスのフローログレコードが、バケットに保存された一連のログファイルオブジェクトに発行されます。フローログが VPC のデータを取得する場合、フローログは、選択された VPC ですべてのネットワークインターフェイスのフローログレコードを発行します。

フローログを Amazon S3 に発行すると、提供されたログに対するデータの取り込み料金とアーカイブ料金が適用されます。詳細については、「Amazon CloudWatch の料金」を参照してください。

フローログに使用する Amazon S3 バケットの作成方法については、Amazon Simple Storage Service ユーザーガイドの「バケットの作成」を参照してください。

複数のアカウントログの詳細については、「AWS ソリューションライブラリの中央ロギング」を参照してください。

目次

フローログファイル

VPC Flow Logs は、フローログレコードを収集し、ログファイルに統合して、5 分間隔でログファイルを Amazon S3 バケットに発行します。各ログファイルには、前の 5 分間に記録された IP トラフィックのフローログレコードが含まれています。

ログファイルの最大ファイルサイズは 75 MB です。ログファイルが 5 分以内にファイルサイズの上限に達した場合、フローログはフローログレコードの追加を停止します。次に、フローログを Amazon S3 バケットに発行してから、新しいログファイルを作成します。

Amazon S3 では、フローログファイルの [最終更新日時] フィールドに、ファイルが Amazon S3 バケットにアップロードされた日時が表示されます。これは、ファイル名のタイムスタンプより後で、Amazon S3 バケットにファイルをアップロードするのにかかった時間によって異なります。

ログファイル形式

ログファイルに指定できる形式は次のとおりです。各ファイルは 1 つの Gzip ファイルに圧縮されます。

- [Text] - プレーンテキスト。これがデフォルトの形式です。
- [Parquet] - Apache Parquet は柱状データ形式です。Parquet 形式のデータに対するクエリは、プレーンテキストのデータに対するクエリに比べて 10 ~ 100 倍高速です。Gzip 圧縮を使用した Parquet 形式のデータは、Gzip 圧縮を使用したプレーンテキストよりもストレージスペースが 20% 少なくなります。

ログファイルオプション

オプションで、次のオプションを指定できます。

- [Hive-compatible S3 prefixes] - Hive 互換ツールにパーティションをインポートする代わりに、Hive 互換プレフィックスを有効にします。クエリを実行する前に、[MSCK REPAIR TABLE] コマンドを使用します。
- [Hourly partitions] - 大量のログがあり、通常は特定の時間にクエリをターゲットにしている場合、ログを時間単位で分割することで、より高速な結果が得られ、クエリコストを節約できます。

ログファイル S3 バケット構造

ログファイルでは、フローログの ID、リージョン、作成日、および送信先オプションに基づくフォルダ構造を使用して、指定された Amazon S3 バケットに保存されます。

デフォルトでは、ファイルは次の場所に配信されます。

```
bucket-and-optional-prefix/AWSLogs/account_id/vpcflowlogs/region/year/month/day/
```

Hive 互換の S3 プレフィックスを有効にすると、ファイルは次の場所に配信されます。

```
bucket-and-optional-prefix/AWSLogs/aws-account-id=account_id/service=vpcflowlogs/aws-
region=region/year=year/month=month/day=day/
```

時間単位のパーティションを有効にすると、ファイルは次の場所に配信されます。

```
bucket-and-optional-prefix/AWSLogs/account_id/vpcflowlogs/region/year/month/day/hour/
```

Hive 互換パーティションを有効にして 1 時間あたりのフローログをパーティション化すると、ファイルは次の場所に配信されます。

```
bucket-and-optional-prefix/AWSLogs/aws-account-id=account_id/service=vpcflowlogs/aws-
region=region/year=year/month=month/day=day/hour=hour/
```

ログファイル名

ログファイルのファイル名は、フローログ ID、リージョン、および作成日時に基づきます。ファイル名は、次の形式です。

```
aws_account_id_vpcflowlogs_region_flow_log_id_YYYYMMDDTHHmmZ_hash.log.gz
```

以下は、us-east-1 リージョンで June 20, 2018 の 16:20 UTC に、リソースに対して AWS アカウント「123456789012」で作成されたフローログのログファイルの例です。ファイルには、終了時刻が 16:20:00 から 16:24:59 の間のフローログレコードが含まれます。

```
123456789012_vpcflowlogs_us-east-1_fl-1234abcd_20180620T1620Z_fe123456.log.gz
```

フローログを Amazon S3 にパブリッシュする IAM プリンシパルの IAM ポリシー

フローログを Amazon S3 バケットに発行するには、アカウントの IAM プリンシパル (例: IAM ユーザー) に十分なアクセス許可が付与されている必要があります。これには、フローログを作成および公開するための特定の `logs:` アクションを操作するアクセス許可が含まれます。IAM ポリシーには以下のアクセス許可が含まれています。

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:DeleteLogDelivery"
        ],
      "Resource": "*"
    }
  ]
}
```

フローログのための Amazon S3 バケットのアクセス許可

デフォルトでは、Amazon S3 バケットとそれに含まれているオブジェクトはプライベートです。バケット所有者のみが、そのバケットとそれに含まれているオブジェクトにアクセスできます。ただし、バケット所有者は、アクセスポリシーを記述することで他のリソースおよびユーザーにアクセス権限を付与することができます。

次のバケットポリシーは、フローログにログを発行するアクセス許可を付与します。バケットに次のアクセス許可を持つポリシーがすでに存在する場合、ポリシーはそのまま保持されます。個々の AWS アカウントの ARN ではなく、ログ配信サービスプリンシパルに、これらのアクセス権限を付与することをお勧めします。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket_name/optional_folder/AWSLogs/account_id/*",
            "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-control"}}
        },
        {
            "Sid": "AWSLogDeliveryCheck",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": ["s3:GetBucketAcl", "s3:ListBucket"],
            "Resource": "arn:aws:s3:::bucket_name"
        }
    ]
}
```

フローログを作成するユーザーがバケットを所有し、バケットの `PutBucketPolicy` アクセス許可を持ち、バケットに十分なログ配信アクセス許可を持つポリシーがない場合、前述のポリシーが自動的にバケットにアタッチされます。このポリシーは、バケットにアタッチされている既存のポリシーを上書きします。

フローログを作成しているユーザーがバケットを所有していないか、バケットに対する `GetBucketPolicy` および `PutBucketPolicy` アクセス権限がない場合、フローログの作成は失敗します。この場合、バケット所有者はバケットに手動で上記のポリシーを追加して、フローログ作成者の AWS アカウント ID を指定する必要があります。詳細については、Amazon Simple Storage Service ユーザーガイドの「S3 バケットポリシーを追加する方法」を参照してください。バケットが複数のアカウントからフローログを受け取る場合は、各アカウントの `Resource` ポリシーステートメントに `AWSLogDeliveryWrite` エレメントエントリを追加します。例えば、次のバケットポリシーでは、AWS

アカウント「123123123123」および「456456456456」に、log-bucket という名前のバケットの flow-logs という名前のフォルダに、フローログの発行を許可します。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": [
             "arn:aws:s3:::log-bucket/flow-logs/AWSLogs/123123123123/*",
             "arn:aws:s3:::log-bucket/flow-logs/AWSLogs/456456456456/*"
             ],
            "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-control"}}
        },
        {
            "Sid": "AWSLogDeliveryCheck",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": ["s3:GetBucketAcl", "s3:ListBucket"],
            "Resource": "arn:aws:s3:::log-bucket"
        }
    ]
}
```

SSE-KMS に使用する必須のキーポリシー

Amazon S3 バケット内のデータを保護するには、Amazon S3 マネージドキーを使用したサーバー側の暗号化 (SSE-S3)、または に格納された KMS キーを使用したサーバー側の暗号化 (SSE-KMS) のいずれかを有効にします。詳細については、Amazon S3 ユーザーガイドの「サーバー側の暗号化を使用したデータの保護」をご参照ください。

SSE-KMS では、AWS マネージドキーまたはカスタマーマネージドキーのいずれかを使用できます。AWS マネージドキーでは、クロスアカウント配信を使用できません。フローログはログ配信アカウントから配信されるため、クロスアカウント配信のアクセス権を付与する必要があります。S3 バケットへのクロスアカウントアクセス権を付与するには、カスタマーマネージドキーを使用し、バケット暗号化を有効にするときに、カスタマーマネージドキーの Amazon リソースネーム (ARN) を指定します。詳細については、Amazon S3 ユーザーガイドの「AWS KMS によるサーバー側の暗号化の指定」をご参照ください。

カスタマーマネージドキーで SSE-KMS を使用する場合、VPC フローログが S3 バケットに書き込めるように、キーのキーポリシー (S3 バケットのバケットポリシーではありません) に以下を追加する必要があります。

```
{
    "Sid": "Allow VPC Flow Logs to use the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "delivery.logs.amazonaws.com"
        ]
    },
   "Action": [
       "kms:Encrypt",
       "kms:Decrypt",
       "kms:ReEncrypt*",
       "kms:GenerateDataKey*",
       "kms:DescribeKey"
    ],
    "Resource": "*"
```

```
}
```

Amazon S3 ログファイルのアクセス許可

Amazon S3 は、必須のバケットポリシーに加えて、アクセスコントロールリスト (ACL) を使用して、フローログによって作成されたログファイルへのアクセスを管理します。デフォルトでは、バケット所有者が各ログファイルで `FULL_CONTROL` 権限を持ちます。ログ配信の所有者 (バケット所有者とは異なる場合) は、アクセス権限を持ちません。ログ配信アカウントには、`READ` および `WRITE` アクセス権限があります。詳細については、Amazon Simple Storage Service ユーザーガイドの「アクセスコントロールリスト (ACL) の概要」を参照してください。

Amazon S3 に発行するフローログの作成

Amazon S3 バケットを作成して設定した後は、ネットワークインターフェイス、サブネット、または VPC のフローログを作成できます。

コンソールを使用してネットワークインターフェイスのフローログを作成するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. 1 つ以上のネットワークインターフェイスのチェックボックスをオンにします。
4. [Actions]、[Create flow log] を選択します。
5. フローログ設定を構成します。詳細については、「フローログ設定を構成するには (p. 346)」を参照してください。

コンソールを使用してサブネットのフローログを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[Subnets] を選択します。
3. 1 つ以上のサブネットのチェックボックスをオンにします。
4. [Actions]、[Create flow log] を選択します。
5. フローログ設定を構成します。詳細については、「フローログ設定を構成するには (p. 346)」を参照してください。

コンソールを使用して VPC のフローログを作成するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. 画面左枠のナビゲーションペインで、[Your VPCs] を選択します。
3. 1 つ以上の VPC のチェックボックスをオンにします。
4. [Actions]、[Create flow log] を選択します。
5. フローログ設定を構成します。詳細については、「フローログ設定を構成するには (p. 346)」を参照してください。

コンソールを使用してフローログ設定を構成するには

1. [フィルタ] で、記録する IP トラフィックデータのタイプを指定します。
   - [Accepted] - 受け入れられたトラフィックのみをログに記録します。
   - [Rejected] - 拒否されたトラフィックのみをログに記録します。

- [All] - 承認されたトラフィックと拒否されたトラフィックをログに記録します。

2. [Maximum aggregation interval] で、フローがキャプチャされ、1 つのフローログレコードに集約される最大期間を選択します。
3. [送信先] で、[S3 バケットへの送信] を選択します。
4. [S3 バケット ARN] で、既存の Amazon S3 バケットの Amazon リソースネーム (ARN) を指定します。オプションで、サブフォルダを含めることができます。例えば、`my-logs` というバケットで `my-bucket` というサブフォルダを指定するには、次の ARN を使用します。

   ```
   arn:aws::s3:::my-bucket/my-logs/
   ```

   `AWSLogs` は予約語であるため、バケットでサブフォルダ名として使用することはできません。

   バケットを所有している場合は、リソースポリシーが自動的に作成され、バケットにアタッチされます。詳細については、「フローログのための Amazon S3 バケットのアクセス許可 (p. 344)」を参照してください。
5. [Log record format] で、フローログレコードの形式を指定します。

   - デフォルトのフローログレコード形式を使用するには、[AWS default format] (AWS のデフォルトの形式)を選択します。
   - カスタム形式を作成するには、[Custom format] を選択します。[Log format] で、フローログレコードに含めるフィールドを選択します。
   - デフォルトの書式フィールドを含むカスタムフローログを作成するには、まず [AWS default format] (AWS のデフォルトの形式) を選択して [Format preview] (形式のプレビュー) のフィールドをコピーし、[Custom format] (カスタム形式) を選択してテキストボックスにフィールドを貼り付けます。
6. [Log file format] で、ログファイルの形式を指定します。

   - [Text] - プレーンテキスト。これがデフォルトの形式です。
   - [Parquet] - Apache Parquet は柱状データ形式です。Parquet 形式のデータに対するクエリは、プレーンテキストのデータに対するクエリに比べて 10 ~ 100 倍高速です。Gzip 圧縮を使用した Parquet 形式のデータは、Gzip 圧縮を使用したプレーンテキストよりもストレージスペースが 20% 少なくなります。
7. (オプション) Hive 互換の S3 プレフィックスを使用するには、[Hive-compatible S3 prefix]、[Enable] を選択します。
8. (オプション) 1 時間あたりのフローログを分割するには、[Every 1 hour (60 mins)] を選択します。
9. (オプション) フローログにタグを追加するには、[Add new tag] を選択し、タグのキーと値を指定します。
10. [フローログの作成] を選択します。

コマンドラインツールを使用して Amazon S3 に発行されるフローログを作成するには

以下のいずれかのコマンドを使用します。

- create-flow-logs (AWS CLI)
- New-EC2FlowLogs (AWS Tools for Windows PowerShell)
- CreateFlowLogs (Amazon EC2 Query API)

次の AWS CLI の例では、VPC `vpc-00112233344556677` のすべてのトラフィックをキャプチャするフローログを作成し、フローログを `flow-log-bucket` と呼ばれる Amazon S3 バケットに配信します。`--log-format` パラメータにより、フローログレコードのカスタム形式が指定されます。

```
aws ec2 create-flow-logs --resource-type VPC --resource-ids vpc-00112233344556677 --
traffic-type ALL --log-destination-type s3 --log-destination arn:aws:s3:::flow-log-
bucket/my-custom-flow-logs/ --log-format '${version} ${vpc-id} ${subnet-id} ${instance-
```

```
id} ${srcaddr} ${dstaddr} ${srcport} ${dstport} ${protocol} ${tcp-flags} ${type} ${pkt-
srcaddr} ${pkt-dstaddr}'
```

Amazon S3 でのフローログレコードの処理

ログファイルは圧縮されます。Amazon S3 コンソールを使用してログファイルを開くと、ファイルは解凍され、フローログレコードが表示されます。ファイルをダウンロードする場合、フローログレコードを表示するには解凍する必要があります。

Amazon Athena を使用し、ログファイルのフローログレコードに対してクエリを実行することもできます。Amazon Athena はインタラクティブなクエリサービスで、Amazon S3 内のデータを標準 SQL を使用して簡単に分析できるようになります。詳細については、Amazon Athena ユーザーガイドの「Amazon VPC フローログのクエリ」を参照してください。

フローログの使用

Amazon EC2、Amazon VPC、CloudWatch、および Amazon S3 コンソールを使用して、フローログを操作できます。

タスク

フローログの使用の管理

デフォルトでは、IAM ユーザーにはフローログを使用するためのアクセス許可がありません。フローログを作成、説明、削除するアクセス許可をユーザーに付与する IAM ユーザーポリシーを作成できます。詳細については、Amazon EC2 API リファレンスの「IAM ユーザーに対する Amazon EC2 リソースに対するアクセス許可の付与」を参照してください。

フローログを作成、説明、削除する完全なアクセス許可をユーザーに付与するポリシー例を次に示します。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteFlowLogs",
        "ec2:CreateFlowLogs",
        "ec2:DescribeFlowLogs"
      ],
      "Resource": "*"
    }
  ]
}
```

発行先が CloudWatch Logs であるか Amazon S3 であるかにより、追加の IAM ロールとアクセス許可の設定が必要になります。詳細については、「CloudWatch Logs へのフローログの発行 (p. 337)」および「フローログを Amazon S3 に発行する (p. 342)」を参照してください。

フローログの作成

VPCs、サブネット、またはネットワークインターフェイスのフローログを作成できます。フローログでは CloudWatch Logs または Amazon S3 にデータを発行できます。

詳細については、「CloudWatch Logs に発行するフローログの作成 (p. 339)」および「Amazon S3 に発行するフローログの作成 (p. 346)」を参照してください。

フローログを表示する

Amazon EC2 および Amazon VPC コンソールでフローログに関する情報を表示するには、特定のリソースの [フローログ] タブを表示します。リソースを選択すると、そのリソースのすべてのフローログが表示されます。表示される情報には、フローログの ID、フローログの設定、およびフローログのステータスに関する情報が含まれます。

ネットワークインターフェイスのフローログに関する情報を表示するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択し、[フローログ] を選択します。フローログに関する情報がタブに表示されます。[送信先タイプ] 列は、フローログを発行する送信先を示します。

VPC またはサブネットのフローログに関する情報を表示するには

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、[VPC] または [サブネット] を選択します。
3. VPC またはサブネットを選択し、[フローログ] を選択します。フローログに関する情報がタブに表示されます。[送信先タイプ] 列は、フローログを発行する送信先を示します。

フローログのタグを追加または削除する

Amazon EC2 および Amazon VPC コンソールで、フローログのタグを追加または削除できます。

ネットワークインターフェイスのフローログのタグを追加または削除するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[Network Interfaces] を選択します。
3. ネットワークインターフェイスを選択し、[フローログ] を選択します。
4. 必要なフローログの [Manage tags (タグの管理)] を選択します。
5. 新しいタグを追加するには、[Create Tag] を選択します。タグを削除するには、削除アイコンを選択します (x)。
6. [Save] を選択します。

VPC またはサブネットのフローログのタグを追加または削除するには

1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで、[VPC] または [サブネット] を選択します。
3. VPC またはサブネットを選択し、[フローログ] を選択します。

4. フローログを選択し、[Actions (アクション)]、[Add/Edit Tags (タグの追加/編集)] の順に選択します。
5. 新しいタグを追加するには、[Create Tag] を選択します。タグを削除するには、削除アイコンを選択します (x)。
6. [Save] を選択します。

フローログレコードを表示する

選択した送信先タイプに応じて、CloudWatch Logs コンソールまたは Amazon S3 コンソールを使用して、フローログレコードを表示できます。フローログを作成してからコンソールに表示されるまでに、数分かかる場合があります。

CloudWatch Logs に対して発行されたフローログレコードを表示するには

1. CloudWatch コンソール (https://console.aws.amazon.com/cloudwatch/) を開きます。
2. ナビゲーションペインで、[ログ] を選択し、フローログを含むロググループを選択します。各ネットワークインターフェイス用のログストリームのリストが表示されます。
3. フローログレコードを表示するネットワークインターフェイスの ID を含むログストリームを選択します。詳細については、「」を参照してくださいフローログレコード (p. 327)

Amazon S3 に対して発行されたフローログレコードを表示するには

1. https://console.aws.amazon.com/s3/ で Amazon S3 コンソールを開きます。
2. [バケット名] で、フローログを発行するバケットを選択します。
3. [名前] で、ログファイルの横にあるチェックボックスを選択します。オブジェクトの概要パネルで、[ダウンロード] を選択します。

フローログレコードの検索

CloudWatch Logs コンソールを使用して、CloudWatch Logs に発行されたフローログレコードを検索できます。メトリクスフィルタを使用すると、フローログレコードをフィルタリングできます。フローログレコードはスペースで区切られます。

CloudWatch Logs コンソールを使用してフローログレコードを検索するには

1. CloudWatch コンソール (https://console.aws.amazon.com/cloudwatch/) を開きます。
2. ナビゲーションペインで、[ロググループ] を選択し、フローログを含むロググループを選択します。各ネットワークインターフェイス用のログストリームのリストが表示されます。
3. 検索するネットワークインターフェイスがわかっている場合は、個々のログストリームを選択します。または、[ロググループの検索] を選択して、ロググループ全体を検索します。ロググループに多数のネットワークインターフェイスがある場合、または選択した時間範囲によっては、この処理に時間がかかる場合があります。
4. [イベントをフィルター] で、次の文字列を入力します。これは、フローログレコードで デフォルトの形式 (p. 327)が使用されていることを前提としています。

```
[version, accountid, interfaceid, srcaddr, dstaddr, srcport, dstport, protocol,
 packets, bytes, start, end, action, logstatus]
```

5. 必要に応じてフィールドの値を指定して、フィルタを変更します。次の例では、特定の送信元 IP アドレスでフィルタリングします。

```
[version, accountid, interfaceid, srcaddr = 10.0.0.1, dstaddr, srcport, dstport,
 protocol, packets, bytes, start, end, action, logstatus]
```

```
[version, accountid, interfaceid, srcaddr = 10.0.2.*, dstaddr, srcport, dstport,
 protocol, packets, bytes, start, end, action, logstatus]
```

次の例では、送信先ポート、バイト数、およびトラフィックが拒否されたかどうかでフィルタリングします。

```
[version, accountid, interfaceid, srcaddr, dstaddr, srcport, dstport = 80 || dstport =
 8080, protocol, packets, bytes, start, end, action, logstatus]
[version, accountid, interfaceid, srcaddr, dstaddr, srcport, dstport = 80 || dstport =
 8080, protocol, packets, bytes >= 400, start, end, action = REJECT, logstatus]
```

フローログの削除

Amazon EC2 と Amazon VPC コンソールを使用して、フローログを削除できます。

これらの手順では、リソースのフローログサービスが無効になります。フローログを削除しても、既存のログストリームは CloudWatch Logs から削除されず、ログファイルは Amazon S3 から削除されません。既存のフローログデータは、それぞれのサービスのコンソールを使用して削除する必要があります。さらに、Amazon S3 に公開するフローログを削除しても、バケットポリシーとログファイルのアクセスコントロールリスト (ACL) は削除されません。

ネットワークインターフェイスのフローログを削除するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで [Network Interfaces] を選択してから、ネットワークインターフェイスを選択します。
3. [フローログ] を選択し、削除するフローログの削除ボタン (X) を選択します。
4. 確認ダイアログボックスで、[Yes, Delete] を選択します。

VPC またはサブネットのフローログを削除するには

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 [VPC] または [サブネット] を選択してから、リソースを選択します。
3. [フローログ] を選択し、削除するフローログの削除ボタン (X) を選択します。
4. 確認ダイアログボックスで、[Yes, Delete] を選択します。

API と CLI の概要

このページで説明しているタスクは、コマンドラインまたは API を使用して実行できます。コマンドラインインターフェイスの詳細および利用できる API アクションの一覧については、「Amazon VPC にアクセスする (p. 1)」を参照してください。

フローログの作成

- create-flow-logs (AWS CLI)
- New-EC2FlowLog (AWS Tools for Windows PowerShell)
- CreateFlowLogs (Amazon EC2 Query API)

フローログの説明

- describe-flow-logs (AWS CLI)

- Get-EC2FlowLog (AWS Tools for Windows PowerShell)
- DescribeFlowLogs (Amazon EC2 Query API)

フローログレコード (ログイベント) の表示

- get-log-events (AWS CLI)
- Get-CWLLogEvent (AWS Tools for Windows PowerShell)
- GetLogEvents (CloudWatch API)

フローログの削除

- delete-flow-logs (AWS CLI)
- Remove-EC2FlowLog (AWS Tools for Windows PowerShell)
- DeleteFlowLogs (Amazon EC2 Query API)

Amazon Athena を使用したフローログのクエリ

Amazon Athena は、標準の SQL を使用して、フローログなどの Amazon S3 内のデータを分析できる対話型のクエリサービスです。VPC フローログで Athena を使用すると、VPC を通過するトラフィックに関する実用的なインサイトをすばやく得ることができます。例えば、仮想プライベートクラウド (VPC) 内のリソースからトップトーカーを特定したり、最も TCP 接続を拒否された IP アドレスを特定したりできます。

必要な AWS リソースと事前定義されたクエリを作成する CloudFormation テンプレートを生成することで、VPC フローログと Athena との統合を合理化および自動化できます。これにより、VPC を通過するトラフィックに関するインサイトを得ることができます。

CloudFormation テンプレートは、次のリソースを作成します。

- Athena データベース。データベース名は vpcflowlogsathenadatabase<flow-logs-subscription-id> です。
- Athena のワークグループ。ワークグループ名は、<flow-log-subscription-id><partition-load-frequency><start-date><end-date>workgroup です。
- フローログレコードに対応するパーティション化された Athena テーブル。テーブル名は、<flow-log-subscription-id><partition-load-frequency><start-date><end-date> です。
- Athena の名前付きクエリのセット。詳細については、「」を参照してください事前に定義されたクエリ (p. 354)
- 指定したスケジュール (毎日、毎週、または毎月) でテーブルに新しいパーティションをロードする Lambda 関数。
- Lambda 関数を実行するためのアクセス権限を付与する IAM ロール。

Requirements

- AWS Lambda と Amazon Athena をサポートするリージョンを選択する必要があります。
- Amazon S3 バケットは、選択したリージョンに存在する必要があります。

Pricing

クエリの実行には、標準の Amazon Athena 料金が発生します 。(パーティションのロード頻度を指定するが、開始日と終了日を指定しない場合) 定期的なスケジュールで新しいパーティションをロードする Lambda 関数には、標準の AWS Lambda 料金が発生します。

タスク

コンソールを使用した CloudFormation テンプレートの生成

最初のフローログが S3 バケットに配信された後、CloudFormation テンプレートを生成し、そのテンプレートを使用してスタックを作成することで、Athena と統合できます。

コンソールを使用してテンプレートを生成するには

1. 次のいずれかを行ってください。
   - Amazon VPC コンソールを開きます。ナビゲーションペインで [お客様の VPC] をクリックして、VPCを選択します。
   - Amazon VPC コンソールを開きます。ナビゲーションペインで [サブネット] をクリックして、サブネットを選択します。
   - Amazon EC2 コンソールを開きます。ナビゲーションペインで [ネットワークインターフェース] クリックして、ネットワークインターフェイスを選択します。
2. [フローログ] タブで、Amazon S3 に発行するフローログを選択し、[アクション]、[Athena 統合の生成] の順に選択します。
3. パーティションのロード頻度を指定します。[なし] を選択した場合は、過去の日付を使用して、パーティションの開始日と終了日を指定する必要があります。[毎日]、[毎週]、または [毎月] を選択した場合は、パーティションの開始日と終了日はオプションになります。開始日と終了日を指定しない場合、CloudFormation テンプレートは、定期的なスケジュールで新しいパーティションをロードする Lambda 関数を作成します。
4. 生成されたテンプレート用の S3 バケット、およびクエリ結果用の S3 バケットを選択または作成します。
5. [Athena 統合を生成] を選択します
6. （オプション）成功メッセージで、CloudFormation テンプレートに指定したバケットに移動するリンクを選択し、テンプレートをカスタマイズします。
7. 成功のメッセージで、[Create CloudFormation stack] (CloudFormation スタックを作成) を選択して、AWS CloudFormation コンソールで [Create Stack] (スタックの作成) ウィザードを開きます。生成された CloudFormation テンプレートの URL は、[テンプレート] セクションで指定されます。ウィザードを完了して、テンプレートで指定されているリソースを作成します。

AWS CLI を使用した CloudFormation テンプレートの生成

最初のフローログが S3 バケットに配信された後、CloudFormation テンプレートを生成して使用して Athena と統合できます。

次の get-flow-logs-integration-template コマンドを使用して、CloudFormation テンプレートを生成します。

```
aws ec2 get-flow-logs-integration-template --cli-input-json file://config.json
```

次は、`config.json` ファイルの例です。

```
{
    "FlowLogId": "fl-12345678901234567",
    "ConfigDeliveryS3DestinationArn": "arn:aws:s3:::my-flow-logs-athena-integration/
templates/",
    "IntegrateServices": {
        "AthenaIntegrations": [
            {
                "IntegrationResultS3DestinationArn": "arn:aws:s3:::my-flow-logs-analysis/
athena-query-results/",
                "PartitionLoadFrequency": "monthly",
                "PartitionStartDate": "2021-01-01T00:00:00",
                "PartitionEndDate": "2021-12-31T00:00:00"
            }
        ]
    }
}
```

生成された CloudFormation テンプレートを使用してスタックを作成するには、次の create-stack コマンドを使用します。

```
aws cloudformation create-stack --stack-name my-vpc-flow-logs --template-body file://my-
cloudformation-template.json
```

事前定義されたクエリを実行する

生成された CloudFormation テンプレートには、AWS ネットワーク内のトラフィックに関する有意義なインサイトをすばやく得るために実行できる一連の定義済みクエリが用意されています。スタックを作成し、すべてのリソースが正しく作成されたことを確認したら、定義済みクエリの 1 つを実行できます。

コンソールを使用して定義済みのクエリを実行するには

1. Athena コンソールを開きます。[ワークグループ] パネルで、CloudFormation テンプレートによって作成されたワークグループを選択します。
2. 定義済みクエリ (p. 354)の 1 つを選択し、必要に応じてパラメータを変更してから、クエリを実行します。
3. Amazon S3 コンソールを開きます。クエリ結果に指定したバケットに移動し、クエリの結果を表示します。

事前に定義されたクエリ

生成された CloudFormation テンプレートによって提供される Athena 名前付きクエリを次に示します。

- vpcFlowLogsAcceptedTraffic — セキュリティグループとネットワーク ACL に基づいて許可された TCP 接続。
- vpcFlowLogsAdminPortTraffic — 管理用 ウェブアプリポートに記録されたトラフィック。
- vpcFlowLogsIPv4Traffic — 記録された IPv4 トラフィックの合計バイト数。
- vpcFlowLogsIPv6Traffic — 記録された IPv6 トラフィックの合計バイト数。
- vpcFlowLogsRejectedTCPTraffic — セキュリティグループまたはネットワーク ACL に基づいて拒否された TCP 接続。
- vpcFlowLogsRejectedTraffic — セキュリティグループまたはネットワーク ACL に基づいて拒否されたトラフィック。
- vpcFlowLogsShrdpTraffic — SSH および RDP トラフィック。
- vpcFlowLogStopTalkers — 記録されたトラフィックが最も多い50個のIPアドレス。

- vpcFlowLogStopTalkersPacketLevel — 記録されたトラフィックが最も多くある 50 個のパケットレベルの IP アドレス。
- vpcFlowLogStopTalkingInstances — 記録されたトラフィックが最も多い50個のインスタンスの ID。
- vpcFlowLogStopTalkingSubnets — 記録されたトラフィックが最も多くある 50 個のサブネットの ID。
- vpcFlowLogStopTcpTraffic — 送信元 IP アドレスに対して記録されたすべての TCP トラフィック。
- vpcFlowLogstotalBytestRansFerred — 記録されたバイト数が最も多い送信元と送信先 IP アドレスの 50 個のペア。
- vpcFlowLogstotalBytestRansFerredPacketLevel — 記録されたバイト数が最も多いパケットレベルの送信元および送信先 IP アドレスの 50 個のペア。
- vpcFlowLogStrafficFrmsrcaddr — 特定の送信元 IP アドレスについて記録されたトラフィック。
- vpcFlowLogStadfficToDr — 特定の送信先 IP アドレスについて記録されたトラフィック。

VPC フローログのトラブルシューティング

フローログを操作する際、発生する可能性のある問題を以下に示します。

問題点

- 不完全なフローログレコード (p. 355)
- フローログが有効でも、フローログレコードまたはロググループがない (p. 356)
- 「LogDestinationNotFoundException」または「Access Denied for LogDestination」エラー (p. 356)
- Amazon S3 バケットポリシーの制限の超過 (p. 356)

不完全なフローログレコード

Problem

フローログレコードが不完全であるか、公開されていません。

Cause

CloudWatch Logs ロググループへのフローログの配信に問題がある可能性があります。

Solution

Amazon EC2 コンソールまたは Amazon VPC コンソールで、関連するリソースの [フローログ] タブを選択します。詳細については、「」を参照してくださいフローログを表示する (p. 349) フローログの表で、エラーは [Status] 列に表示されます。または、describe-flow-logs コマンドを使用し、`DeliverLogsErrorMessage` フィールドに返された値を確認します。次のいずれかのエラーが表示される場合があります。

- `Rate limited`: このエラーは、CloudWatch Logs のスロットリングが適用されている場合に発生することがあります。ネットワークインターフェイスのフローログのレコード数が、特定の期間内に発行できるレコードの最大数より多い場合などが該当します。このエラーは、作成できる CloudWatch Logs ロググループの数がクォータに達した場合にも発生することがあります。詳細については、Amazon CloudWatch ユーザーガイドの「CloudWatch Service Quotas」を参照してください。
- `Access error`: このエラーは、次のいずれかの原因で発生することがあります。
  - フローログの IAM ロールに、CloudWatch Logs ロググループにフローログレコードを発行するための十分なアクセス許可がありません。
  - IAM ロールにフローログサービスとの信頼関係がない
  - 信頼関係によりフローログサービスがプリンシパルとして指定されていない

詳細については、「」を参照してくださいCloudWatch Logs へのフローログ発行のための IAM ロール (p. 337)

- `Unknown error`: 内部エラーがフローログサービスで発生しました。

フローログが有効でも、フローログレコードまたはロググループがない

Problem

フローログを作成し、Amazon VPC または Amazon EC2 コンソールにフローログが `Active` と表示されます。ただし、CloudWatch Logs のログストリームや、Amazon S3 バケットのログファイルは表示できない場合があります。

Cause

原因は、次のいずれかである可能性があります。

- フローログはまだ作成中です。場合によっては、対象のロググループのフローログを作成してから、データが表示されるまでに 10 分以上かかることがあります。
- ネットワークインターフェイスに対して記録されたトラフィックがまだありません。CloudWatch Logs のロググループは、トラフィックの記録時にのみ作成されます。

Solution

ロググループが作成されるか、トラフィックが記録されるまで数分待ちます。

「LogDestinationNotFoundException」または「Access Denied for LogDestination」エラー

Problem

フローログを作成しようとすると、`Access Denied for LogDestination` または `LogDestinationNotFoundException` エラーが発生します。

Cause

データを Amazon S3 バケットに発行するフローログを作成するときに、これらのエラーが発生する場合があります。このエラーは、指定された S3 バケットが見つからないか、バケットポリシーに問題があることを示します。

Solution

次のいずれかを行ってください。

- 既存の S3 バケットの ARN を指定したこと、および ARN が正しい形式であることを確認します。
- S3 バケットを所有していない場合は、バケットポリシー (p. 344) にログを発行するための十分なアクセス許可があることを検証します。バケットポリシーで、アカウント ID とバケット名を検証します。

Amazon S3 バケットポリシーの制限の超過

Problem

フローログを作成しようとすると、`LogDestinationPermissionIssueException` エラーが発生します。

Cause

Amazon S3 バケットポリシーのサイズは 20 KB に制限されています。

Amazon S3 バケットに発行するフローログを作成するたびに、指定されたバケットの ARN (フォルダパスを含む) がバケットのポリシーの `Resource` 要素に自動的に追加されます。

同じバケットに発行する複数のフローログを作成すると、バケットポリシーの制限を超える可能性があります。

Solution

次のいずれかを行ってください。

- 不要になったフローログエントリを削除して、バケットのポリシーをクリーンアップします。
- 個々のフローログエントリを以下で置き換えて、バケット全体にアクセス権限を付与します。

```
arn:aws:s3:::bucket_name/*
```

バケット全体にアクセス権限を付与した場合、新しいフローログのサブスクリプションによってバケットポリシーに新しいアクセス権限が追加されることはありません。

VPN 接続

以下の VPN 接続オプションを使用すると、Amazon VPC をリモートのネットワークおよびユーザーに接続できます。

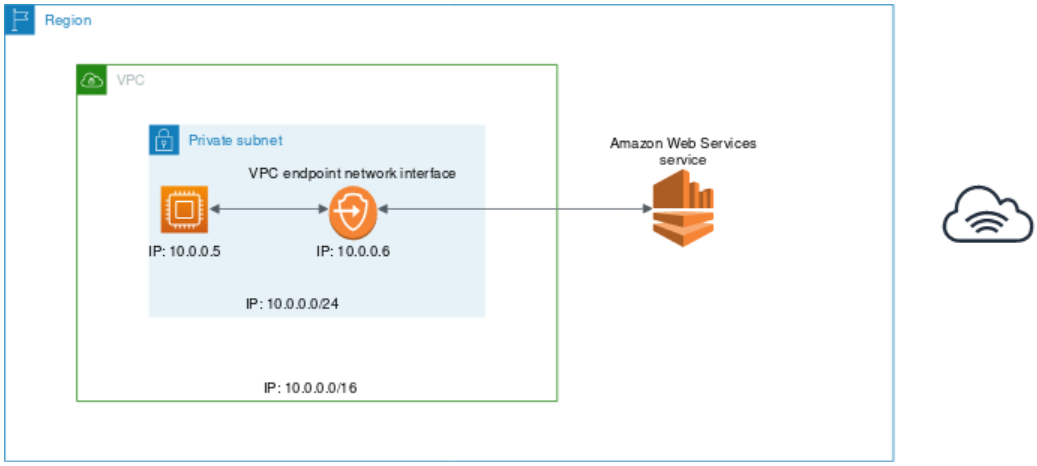
VPN 接続オプション	説明
AWS Site-to-Site VPN	VPC とリモートネットワーク間で、IPsec および VPN 接続を作成できます。Site-to-Site VPN 接続の AWS 側では、仮想プライベートゲートウェイまたはトランジットゲートウェイによって、自動フェイルオーバーのための 2 つの VPN エンドポイント (トンネル) が提供されます。Site-to-Site VPN 接続のリモート側でカスタマーゲートウェイデバイスを設定します。詳細については、AWS Site-to-Site VPN ユーザーガイドを参照してください。
AWS Client VPN	AWS Client VPN は、AWS リソースまたはオンプレミスネットワークに安全にアクセスできるようにする、クライアントベースのマネージド VPN サービスです。AWS Client VPN の場合は、ユーザーが接続して安全な TLS VPN セッションを確立できるエンドポイントを設定します。そうすることにより、クライアントは OpenVPN ベースの VPN クライアントを使用して、どこからでも AWS またはオンプレミスのリソースにアクセスできるようになります。詳細については、AWS Client VPN 管理ガイド を参照してください。
AWSVPN CloudHub	リモートネットワークが複数ある (たとえば、複数の支社がある) 場合は、仮想プライベートゲートウェイを通じて複数の AWS Site-to-Site VPN 接続を作成すると、それらのネットワーク間で通信できるようになります。詳細については、AWS Site-to-Site VPN ユーザーガイド の「VPN CloudHub を使用した安全なサイト間通信の提供」を参照してください。
サードパーティー製ソフトウェア VPN アプライアンス	サードパーティー製ソフトウェア VPN アプライアンスを実行する VPC の Amazon EC2 インスタンスを使用して、リモートネットワークへの VPN 接続を作成できます。AWS は、サードパーティー製ソフトウェア VPN アプライアンスを提供および維持しません。ただし、パートナーやオープンソースコミュニティが提供する様々な製品を選択することができます。 AWS Marketplace でサードパーティー製ソフトウェア VPN アプライアンスを検索します。

また、AWS Direct Connect を使用して、リモートのネットワークから VPC への専用のプライベート接続を作成できます。この接続を AWS Site-to-Site VPN 接続と組み合わせると、IPsec で暗号化された接続を作成できます。詳細については、AWS Direct Connectユーザーガイドの「AWS Direct Connect とは」を参照してください。

AWS PrivateLink および VPC エンドポイント

AWS PrivateLink は、VPC とサービスの間のトラフィックをインターネットに公開することなく、AWS またはオンプレミスでホストされているサービスと仮想プライベートクラウド (VPC)との間のプライベート接続を確立します。

AWS PrivateLink を使用するには、サービスの VPC エンドポイントを VPC 内に作成します。サポートされるサービスにより要求される VPC エンドポイントのタイプを作成します。これによって Elastic Network Interface がサブネットに作成され、そのプライベート IP アドレスがサービスへのトラフィックのエントリポイントとなります。以下の図表は、 AWS PrivateLink をサポートする AWS サービスに VPC を安全に接続するための基本的なアーキテクチャです。



AWS PrivateLink を使用する独自の VPC エンドポイントサービスを作成し、このサービスにアクセスすることを他の AWS のお客様に許可できます。

詳細については、 AWS PrivateLink のユーザーガイドを参照してください。

AWS Network Firewall

AWS Network Firewall を使用して、VPC の境界でネットワークトラフィックをフィルタリングできます。Network Firewall は、ステートフルでマネージド型のネットワークファイアウォールならびに侵入検知および防止サービスです。詳細については、「AWS Network Firewall デベロッパーガイド」を参照してください。

次の AWS リソースを使用して Network Firewall を実装します。

Network Firewall のリソース	説明
ファイアウォール	ファイアウォールは、ファイアウォールポリシーのネットワークトラフィックフィルタリング動作を、保護対象とする VPC に接続します。ファイアウォール設定には、ファイアウォールエンドポイントが配置されるアベイラビリティーゾーンおよびサブネットの仕様が含まれます。また、AWS ファイアウォールのリソースにおけるファイアウォールのログ設定やタグ付けなどの高レベルの設定も定義します。 詳細については、「AWS Network Firewall のファイアウォール」を参照してください。
ファイアウォールポリシー	ファイアウォールポリシーは、ファイアウォールのモニタリングおよび保護動作を定義します。動作の詳細は、ポリシーに追加するルールグループ、および一部のポリシーのデフォルト設定で定義されます。ファイアウォールポリシーを使用するには、1 つ以上のファイアウォールに関連付けます。 詳細については、「AWS Network Firewall のファイアウォールポリシー」を参照してください。
ルールグループ	ルールグループは、ネットワークトラフィックを検査および処理するための再利用可能な条件のセットです。ポリシー設定の一部として、ファイアウォールポリシーに 1 つ以上のルールグループを追加します。ステートレスルールグループを定義して、各ネットワークパケットを個別に検査できます。ステートレスルールグループは、Amazon VPC ネットワークアクセスコントロールリスト (ACL) と動作および使用態様が似ています。また、ステートフルルールグループを定義して、トラフィックフローのコンテキストでパケットを検査することもできます。ステートフルルールグループは、Amazon VPC セキュリティグループと動作と使用態様が似ています。 詳細については、「AWS Network Firewall のルールグループ」を参照してください。

AWS Firewall Managerを使用して、 AWS Organizationsのアカウントおよびアプリケーション全体で Network Firewall リソースを一元的に構成および管理することもできます。Firewall Manager で 1 つのアカウントを使用して、複数のアカウントのファイアウォールを管理できます。詳細については、AWS WAF、AWS Firewall Manager、AWS Shield Advanced デベロッパーガイドの「AWS Firewall Manager」を参照してください。

Route 53 Resolver DNS Firewall

DNS Firewall では、VPC に関連付けるルールグループにドメイン名のフィルタリングルールを定義します。許可またはブロックするドメイン名のリストを指定できます。また、ブロックする DNS クエリのレスポンスをカスタマイズできます。詳細については、Route 53 リゾルバ DNS ファイアウォールのドキュメント を参照してください。

次の AWS リソースを使用して、DNS ファイアウォールを実装します。

DNS ファイアウォールリソース	説明
DNS ファイアウォールルールグループ	DNS ファイアウォールルールグループは、DNS クエリをフィルタリングするための DNS ファイアウォールルールの再利用可能な名前付きコレクションです。ルールグループにフィルタリングルールを設定し、そのルールグループを Amazon VPC の 1 つ以上の VPC に関連付けます。ルールグループを VPC に関連付けると、VPC の DNS Firewall フィルタリングが有効になります。その後、関連付けられているルールグループを持つ VPC の DNS クエリを Resolver が受信すると、そのクエリは DNS Firewall に送信され、フィルタリングが行われます。 ルールグループ内の各ルールは、ドメインがリスト内のドメイン仕様に一致する DNS クエリに対して実行するドメインリスト 1 つとアクションを指定します。一致するクエリについて許可、ブロック、アラートを行うことができます。ブロックしたクエリのカスタムレスポンスも定義できます。 詳細については、「 Route 53 Resolver DNS Firewall」の「ルールグループとルール 」を参照してください。
ドメインリスト	ドメインリストは、ルールグループ内の DNS Firewall ルールで使用する、再利用可能なドメイン仕様のセットです。 詳細については、「 Route 53 Resolver DNS Firewall」の「ドメインリスト 」を参照してください。

AWS Firewall Manager を使用して、AWS Organizations のアカウントおよび組織全体で DNS ファイアウォールリソースを一元的に設定および管理することもできます。Firewall Manager で 1 つのアカウントを使用して、複数のアカウントのファイアウォールを管理できます。詳細については、AWS WAF、AWS Firewall Manager、AWS Shield Advanced デベロッパーガイドの「AWS Firewall Manager」を参照してください。

Amazon VPC クォータ

以下のテーブルは、AWS アカウントに対してリージョン別に適用される Amazon VPC リソースのクォータ (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。これらのクォータの一部については、Amazon EC2 コンソールの [制限] ページを使用して現在のクォータを表示できます。

リソースごとに適用されるクォータの引き上げをリクエストすると、引き上げられたクォータはそのリージョン内のすべてのリソースに適用されます。

VPC とサブネット

名前	デフォルト	調整可能	コメント
リージョンあたりの VPC の数	5	はい	このクォータを引き上げると、リージョンあたりのインターネットゲートウェイのクォータが同じ数だけ増加します。 この制限を引き上げて、リージョンあたり 100 個の VPC を持たせることができます。
VPC 当たりのサブネットの数	200	はい	
VPC 当たりの IPv4 CIDR ブロック	5	はい (最大 50)	このプライマリ CIDR ブロックとすべてのセカンダリ CIDR ブロックは、このクォータに対してカウントされます。
VPC 当たりの IPv6 CIDR ブロック	1	いいえ	

DNS

各 EC2 インスタンスは Route 53 Resolver (具体的には 10.0.0.2 などの .2 アドレス、および 169.254.169.253) へパケット数をネットワークインターフェイスあたり 1024 パケット/秒を送信できます。このクォータを増やすことはできません。Route 53 Resolver でサポートされる 1 秒あたりの DNS クエリの数は、クエリのタイプ、レスポンスのサイズ、および使用中のプロトコルにより異なります。スケーラブルな DNS アーキテクチャの詳細および推奨については、「アクティブディレクトリを使用した AWS ハイブリッド DNS 技術ガイド」を参照してください。

Elastic IP アドレス (IPv4)

名前	デフォルト	調整可能	コメント
リージョン当たりの Elastic IP アドレスの数	5	はい	このクォータは、個々の AWS アカウント VPC および共有 VPC に適用されます。

Gateways

名前	デフォルト	調整可能	コメント
リージョンあたりの Egress-only インターネットゲートウェイの数	5	はい	このクォータを引き上げるには、リージョンごとの VPC のクォータを引き上げます。 一度に VPC にアタッチできる Egress-Only インターネットゲートウェイは 1 つだけです。
リージョンあたりのインターネットゲートウェイの数	5	はい	このクォータを引き上げるには、リージョンごとの VPC のクォータを引き上げます。 一度に VPC にアタッチできるインターネットゲートウェイは 1 つだけです。
アベイラビリティーゾーン当たりの NAT ゲートウェイの数	5	はい	NAT ゲートウェイは、`pending`、`active`、`deleting` のいずれかの状態でクォータにカウントされます。
VPC あたりのキャリアゲートウェイ数	1	いいえ	

カスタマーマネージドプレフィックスリスト

名前	デフォルト	コメント
リージョンあたりのプレフィックスリスト数	100	
プレフィックスリストあたりのバージョン数	1,000	プレフィックスリストに 1,000 個の保存されたバージョンがあり、新しいバージョンを追加する場合、新しいバージョンを追加できるように古いバージョンが削除されます。
プレフィックスリストあたりの最大エントリ数	1,000	リソース内でプレフィックスリストを参照する場合、プレフィックスリストのエントリの最大数は、リソースのエントリの数のクォータに対してカウントされます。例えば、エントリ数が 20 個のプレフィックスリストを作成し、セキュリティグループルール内でそのプレフィックスリストを参照する場合、セキュリティグループの 20 個のルールとしてカウントされます。
リソースタイプごとのプレフィックスリストへの参照	5,000	このクォータは、プレフィックスリストを参照できるリソースタイプごとに適用されます。例えば、すべてのセキュリティグループにわたってプレフィックスリストへの参照を 5,000 個と、すべてのサブネットルートテーブルにわたってプレフィックスリストへの参照を 5,000 個作成することができます。プレフィックスリストを他の AWS アカウントと共有する場合、プレフィックスリス

名前	デフォルト	コメント
		トへの他のアカウントの参照は、このクォータに対してカウントされます。

ネットワーク ACL

名前	デフォルト	調整可能	コメント
VPC 当たりのネットワーク ACL の数	200	はい	1 つのネットワーク ACL を VPC の 1 つ以上のサブネットに関連付けることができます。
ネットワーク ACL 当たりのルールの数	20	はい	これは、単一のネットワーク ACL の一方向クォータです。このクォータは、IPv4 ルールと IPv6 ルールに個別に適用されます。例えば、IPv4 トラフィックと IPv6 トラフィックそれぞれに 20 の進入ルールを含めることができます。このクォータには、デフォルトで拒否されるルールが含まれます (ルール番号は、IPv4 では 32767、IPv6 では 32768、または Amazon VPC コンソールではアスタリスク * です)。 このクォータは最大 40 まで引き上げることができます。ただし、追加のルールを処理するためのワークロードが増えるため、ネットワークのパフォーマンスに影響することがあります。

ネットワークインターフェイス

名前	デフォルト	調整可能	コメント
インスタンス当たりのネットワークインターフェイス	インスタンスタイプによって異なる	いいえ	詳細については、「各インスタンスタイプのネットワークインターフェイス」を参照してください。
リージョンあたりのネットワークインターフェイス	5,000	はい	このクォータは、個々の AWS アカウント VPC および共有 VPC に適用されます。

ルートテーブル

名前	デフォルト	調整可能	コメント
VPC 当たりのルートテーブルの数	200	はい	メインルートテーブルは、このクォータに対してカウントされます。
ルートテーブル当たりのルートの数 (伝播されないルート)	50	はい	このクォータは最大 1,000 まで引き上げ可能です。ただし、ネットワークパフォー

名前	デフォルト	調整可能	コメント
			マンスに影響する場合があります。このクォータは、IPv4 ルートと IPv6 ルートに対して個別に適用されます。 125 を超えるルートがある場合は、パフォーマンスを高めるため、呼び出しをページ分割してルートテーブルについて説明することをお勧めします。
ルートテーブル当たりの、BGP でアドバタイズされるルートの数 (伝播されるルート)	100	いいえ	追加のプレフィックスが必要な場合は、デフォルトルートをアドバタイズします。

セキュリティグループ

名前	デフォルト	調整可能	コメント
リージョンあたりの VPC セキュリティグループの数	2,500	はい	このクォータは、個々の AWS アカウント VPC および共有 VPC に適用されます。 このクォータを引き上げてリージョンのセキュリティグループを 5,000 以上にする場合は、パフォーマンスを高めるため、呼び出しをページ分割してセキュリティグループについて記述することをお勧めします。
セキュリティグループ当たりのインバウンドルールまたはアウトバウンドルールの数	60	はい	セキュリティグループあたり 60 個のインバウンドルールと 60 個のアウトバウンドルール (合計 120 個のルール) を指定できます。このクォータは、IPv4 ルートと IPv6 ルートに対して個別に適用されます。例えば、セキュリティグループで、IPv4 トラフィックと IPv6 トラフィックにそれぞれ 60 のインバウンドルールを含めることができます。 クォータの変更は、インバウンドルールとアウトバウンドルールの両方に適用されます。このクォータにネットワークインターフェイスあたりのセキュリティグループのクォータを乗算した値が 1,000 を超えることはできません。例えば、クォータを 100 個に引き上げると、ネットワークインターフェイスあたりのセキュリティグループ数のクォータが 10 に減少します。
ネットワークインターフェイス当たりのセキュリティグループ	5	はい (最大 16)	このクォータは、IPv4 ルールと IPv6 ルールに対して個別に適用されます。ネットワークインターフェイスあたりのセキュリティグループのクォータと、セキュリティグループあたりのルールのクォータを乗算した値が、1,000 を超えることはできません。例えば、このクォータを 10 に引き上げる

名前	デフォルト	調整可能	コメント
			と、セキュリティグループあたりのルール数のクォータは 100 に減少します。

VPC ピアリング接続

名前	デフォルト	調整可能	コメント
VPC 当たりのアクティブな VPC ピアリング接続	50	はい (最大 125)	このクォータを引き上げる場合は、それに応じてルートテーブルごとのエントリー数を増やす必要があります。
未処理の VPC ピアリング接続リクエスト	25	はい	これは、アカウント行った未処理の VPC ピアリング接続リクエストの数です。
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)	いいえ	

VPC エンドポイント

名前	デフォルト	調整可能	コメント
リージョンあたりのゲートウェイ VPC エンドポイントの数	20	はい	1 VPC あたりのゲートウェイエンドポイント数を 255 以上にすることはできません。
VPC あたりのインターフェイスおよび Gateway Load Balancer エンドポイント	50	はい	これは、VPC 内のインターフェイスエンドポイントおよび Gateway Load Balancer エンドポイントの最大数の合計クォータです。このクォータを引き上げるには、 お問い合わせくださいAWS Support
VPC エンドポイントポリシーのサイズ	20,480 文字	いいえ	このクォータには空白が含まれます。

以下の最大転送単位 (MTU) ルールは、VPC エンドポイントを通過するトラフィックに適用されます。

- ネットワーク接続の最大送信単位 (MTU) とは、VPC エンドポイントを通じて渡すことができる最大許容パケットサイズ (バイト単位) です。MTU が大きいほど、より多くのデータを単一のパケットで渡すことができます。VPC エンドポイントは、8500 バイトの MTU をサポートします。
- VPC エンドポイントに到達したサイズが 8500 バイトを超えるパケットはドロップされます。
- VPC エンドポイントは、FRAG_NEEDEDICMP パケットを生成しないため、パス MTU 検出 (PMTUD) はサポートされません。
- VPC エンドポイントは、すべてのパケットに対して最大セグメントサイズ (MSS) クランプを適用します。詳細については、「RFC879」を参照してください。

VPC 共有

すべての標準 VPC のクォータは共有 VPC に適用されます。

これらのクォータを引き上げる際は、AWS Support にお問い合わせください。AWS では、`DescribeSecurityGroups` および `DescribeSubnets` の API コールをページ分割してから引き上げをリクエストされることをお勧めしています。

名前	デフォルト	調整可能	コメント
VPC ごとの参加者アカウント	100	はい	これは、VPC 内のサブネットを共有できる個々の参加者アカウントの数です。これは VPC あたりのクォータで、VPC で共有されているすべてのサブネットに適用されます。このクォータを引き上げるには、 にお問い合わせくださいAWS Support VPC 所有者は、参加者のリソースにアタッチされているネットワークインターフェイスとセキュリティグループを表示できます。
アカウントと共有できるサブネット	100	はい	これは、AWS アカウントと共有できるサブネットの最大数です。

Amazon EC2 API スロットリング

Amazon EC2 スロットリングの詳細については、Amazon EC2 API リファレンスの「API リクエストのスロットリング」を参照してください。

その他のクォータリソース

詳細については、以下を参照してください:

- 「Amazon VPC Transit Gateways」の「トランジットゲートウェイのクォータ」
- 「AWS Client VPN 管理者ガイド」の「AWS Client VPN クォータ」
- AWS Site-to-Site VPN ユーザーガイド の Site-to-Site VPN のクォータ
- AWS Direct Connectユーザーガイド の AWS Direct Connectクォータ

ドキュメント履歴

次の表に、Amazon VPC ユーザーガイドおよび Amazon VPC ピアリングガイドの各リリースにおける重要な変更点を示します。

update-history-change	update-history-description	update-history-date
VPC Flow Logs 配信オプションを Amazon S3 に記録する (p. 368)	Apache Parquet ログファイル形式、時間単位のパーティション、および Hive 互換の S3 プレフィックスを指定できます。	2021 年 10 月 13 日
Amazon EC2 グローバルビュー	Amazon EC2 グローバルビューを使用すると、複数のAWSリージョンの VPC、サブネット、インスタンス、セキュリティグループ、およびボリュームを 1 つのコンソールで表示します。	2021 年 9 月 1 日
より具体的なルート (p. 368)	ローカルルートよりも具体的なルートを追加できます。より具体的なルートを使用して、VPC 内のサブネット間のトラフィック（East-West トラフィック）をミドルボックスアプライアンスにリダイレクトできます。VPC 内のサブネットの IPv4 または IPv6 CIDR ブロック全体に一致するように、ルートの送信先を設定できます。	2021 年 8 月 30 日
セキュリティグループルールのリソース ID とタグ付けについてのサポート (p. 368)	リソース ID により、セキュリティグループルールを参照することができます。また、セキュリティグループにはタグも追加できます。	2021 年 7 月 7 日
プライベート NAT ゲートウェイ (p. 368)	VPC 間または VPC とオンプレミスネットワーク間の送信専用プライベート通信にプライベート NAT ゲートウェイを使用できます。	2021 年 6 月 10 日
Amazon S3 インターフェイスエンドポイント	Amazon S3 インターフェイスエンドポイントを作成できます。	2021 年 2 月 2 日
Gateway Load Balancer エンドポイント	VPC 内に Gateway Load Balancer エンドポイントを作成して、Gateway Load Balancer を使用して設定した VPC エンドポイントサービスにトラフィックをルーティングできます。	2020 年 11 月 10 日
キャリアゲートウェイ	キャリアゲートウェイを作成して、特定の場所にあるキャリアネットワークからのインバウンド	2020 年 8 月 6 日

	トラフィックを許可し、キャリアネットワークおよびインターネットへのアウトバウンドトラフィックを許可することができます。	
作成時のタグ付け (p. 368)	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
作成時のタグ付け (p. 368)	タグを追加できるのは、VPC、DHCP オプション、インターネットゲートウェイ、Egress-Only ゲートウェイ、ネットワーク ACL、およびセキュリティグループを作成する場合です。	2020 年 6 月 30 日
マネージドプレフィックスリスト	プレフィックスリスト内の CIDR ブロックのセットを作成および管理できます。	2020 年 6 月 29 日
フローログの強化	新しいフローログフィールドが使用でき、CloudWatch Logs に発行するフローログのカスタム形式を指定できます。	2020 年 5 月 4 日
フローログのタグ付けサポート	フローログにタグを追加できます。	2020 年 3 月 16 日
NAT ゲートウェイ作成時のタグ	タグは、NAT ゲートウェイの作成時に追加できます。	2020 年 3 月 9 日
VPC エンドポイントとエンドポイントサービスの条件キー	EC2 条件キーを使用して、VPC エンドポイントおよびエンドポイントサービスへのアクセスを制御できます。	2020 年 3 月 6 日
VPC エンドポイントおよび VPC エンドポイントサービス作成のタグ	VPC エンドポイントまたは VPC エンドポイントサービスを作成するときに、タグを追加できます。	2020 年 2 月 5 日
フローログの最大集約間隔	フローがキャプチャされ、フローログレコードに集約される最大期間を指定できます。	2020 年 2 月 4 日
ネットワーク境界グループ設定	Amazon Virtual Private Cloud Console から VPC のネットワーク境界グループを設定できます。	2020 年 1 月 22 日
プライベート DNS 名	プライベート DNS 名を使用して、VPC 内から AWS PrivateLink ベースのサービスにプライベートにアクセスできます。	2020 年 1 月 6 日
ゲートウェイルートテーブル	ルートテーブルをゲートウェイに関連付けて、インバウンド VPC トラフィックを VPC 内の特定のネットワークインターフェイスにルーティングできます。	2019 年 12 月 3 日

フローログの強化	フローログのカスタム形式を指定し、フローログレコードで返すフィールドを選択できます。	2019 年 9 月 11 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
AWS Site-to-Site VPN	AWS マネージド VPN は AWS Site-to-Site VPN と呼ばれるようになりました。	2018 年 12 月 18 日
VPC 共有	同じ VPC 内にあるサブネットを同じ AWS 組織内の複数のアカウントと共有できます。	2018 年 11 月 27 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 エンドポイントサービス	VPC で独自の AWS PrivateLink サービスを作成し、インターフェイス VPC エンドポイントを介して、このサービスに他の AWS アカウントやユーザーが接続することを許可できます。	2017 年 11 月 28 日
デフォルトサブネットの作成	アベイラビリティーゾーンにデフォルトサブネットがない場合は、これを作成できます。	2017 年 11 月 9 日
のサービスのAWSインターフェイス VPC エンドポイント	インターフェイスエンドポイントを作成して AWS の一部のサービスにプライベートに接続できます。インターフェイスエンドポイントは、サービスへのトラフィックのエントリポイントとなるプライベート IP アドレスを持つネットワークインターフェイスです。	2017 年 11 月 8 日
NAT ゲートウェイのタグ付けのサポート	NAT ゲートウェイにタグを付けることができます。	2017 年 9 月 7 日
NAT ゲートウェイの Amazon CloudWatch メトリクス	NAT ゲートウェイの CloudWatch メトリクスを表示できます。	2017 年 9 月 7 日
セキュリティグループルールの説明	説明をセキュリティグループに追加できます。	2017 年 8 月 31 日
VPC のセカンダリ IPv4 CIDR ブロック	VPC に複数の IPv4 CIDR ブロックを追加できます。	2017 年 8 月 29 日
DynamoDB の VPC エンドポイント	VPC エンドポイントを使用して、VPC から Amazon DynamoDB にアクセスできます。	2017 年 8 月 16 日
Elastic IP アドレスの復元	Elastic IP アドレスを解放した場合、復元できる場合があります。	2017 年 8 月 11 日

デフォルト VPC の作成	新しいデフォルト VPC を作成するには、既存のデフォルト VPC を削除します。	2017 年 27 月 7 日
IPv6 サポート	VPC CIDR ブロックを IPv6 と関連付け、IPv6 アドレスを VPC 内のリソースに割り当てることができます。	2016 年 12 月 1 日
非RFC 1918 IP アドレス範囲の DNS 解決サポート (p. 368)	Amazon DNS サーバーは、プライベート DNS ホスト名をすべてのアドレス空間のプライベート IP アドレスに解決できます。	2016 年 10 月 24 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通信できるように、ピアリング接続を変更できます。	2016 年 4 月 26 日
NAT ゲートウェイ	パブリックサブネットで NAT ゲートウェイを作成し、プライベートサブネットのインスタンスからインターネットや他の AWS サービスへのアウトバウンドトラフィックを開始することができます。	2015 年 12 月 17 日
VPC フローログ	フローログを作成して、VPC のネットワークインターフェイスとの間で行き来する IP トラフィックに関する情報をキャプチャできます。	2015 年 6 月 10 日
VPC エンドポイント ()	エンドポイントにより、インターネット、VPN 接続、NAT インスタンス、または AWS を経由せずに、VPC と他の AWS Direct Connect サービスとをプライベートに接続できます。	2015 年 5 月 11 日

ClassicLink	ClassicLink を使用すると、EC2-Classic インスタンスを自アカウントの VPC にリンクできます。これによって、VPC のセキュリティグループを EC2-Classic インスタンスに関連付け、プライベート IP アドレスを使用して EC2-Classic インスタンスと VPC 内のインスタンスが通信できるようになります。	2015 年 1 月 7 日
プライベートホストゾーンの使用	Route 53 のプライベートホストゾーンで定義したカスタムの DNS ドメイン名を使用して、VPC のリソースにアクセスできます。	2014 年 11 月 5 日
サブネットのパブリック IP アドレス属性の変更	サブネットのパブリック IP アドレス属性を変更して、そのサブネットで起動するインスタンスがパブリック IP アドレスを受け取るかどうかを示すことができます。	2014 年 6 月 21 日
VPC ピアリング	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
パブリック IP アドレスの割り当て	起動時にパブリック IP アドレスをインスタンスに割り当てられます。	2013 年 8 月 20 日
DNS ホスト名の有効化と DNS 解決の無効化	VPC のデフォルトを変更したり、DNS 解決を無効にしたり、DNS ホスト名を有効にしたりできます。	2013 年 3 月 11 日
VPC Everywhere (p. 368)	5 つの AWS リージョンの VPC、複数のアベイラビリティーゾーンの VPC、AWS アカウントごとの複数の VPC、および VPC ごとの複数の VPN 接続に対するサポートが追加されました。	2011 年 8 月 3 日
Dedicated Instances (p. 368)	ハードウェア専有インスタンスとは、単一のお客様専用のハードウェアを実行する VPC 内で起動される Amazon EC2 インスタンスのことです。	2011 年 3 月 27 日