aws

ユーザーガイド

Amazon Virtual Private Cloud



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Virtual Private Cloud: ユーザーガイド

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

Amazon VPC とは? ... 1
機能 ... 1
Amazon VPC の開始方法 ... 3
Amazon EC2 の使用 ... 3
Amazon VPC の料金 ... 3
Amazon VPC の仕組み ... 6
VPC とサブネット ... 7
デフォルトの VPC とデフォルト以外の VPC ... 7
ルートテーブル ... 8
インターネットへのアクセス ... 8
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 ... 9
VPC とネットワークの接続 ... 10
AWS プライベートグローバルネットワーク ... 10
VPC のプランニング ... 12
AWS アカウント へのサインアップ ... 12
AWS アカウント へのサインアップ ... 12
アクセス許可の確認 ... 12
IP アドレスの範囲を決定する ... 12
アベイラビリティーゾーンの選択 ... 13
インターネット接続の計画 ... 13
VPC の作成 ... 14
アプリケーションをデプロイします ... 14
IP アドレス指定 ... 16
プライベート IPv4 アドレス ... 17
パブリック IPv4 アドレス ... 18
IPv6 アドレス ... 19
パブリック IPv6 アドレス ... 20
プライベート IPv6 アドレス ... 21
独自の IP アドレスを使用する ... 22
Amazon VPC IP Address Manager を使用する ... 23
VPC CIDR ブロック ... 23
IPv4 VPC CIDR ブロック ... 23
VPC の IPv4 CIDR ブロックを管理する ... 25
IPv4 CIDR ブロック関連付けの制限 ... 27

IPv6 VPC CIDR ブロック 29
サブネット CIDR ブロック 30
IPv4 のサブネットのサイズ設定 31
IPv6 のサブネットのサイズ設定 31
IPv4 と IPv6 を比較する 32
マネージドプレフィックスリスト 34
プレフィックスリストの概念とルール 35
プレフィックスリストの Identity and Access Management 36
カスタマーマネージドプレフィックスリスト 37
AWS マネージドプレフィックスリスト 47
プレフィックスリストを使用して AWS インフラストラクチャ管理を最適化する 50
AWS IP アドレスの範囲 52
ダウンロード 53
送信コントロール 53
位置情報フィード 54
アドレス範囲の検索 54
構文 61
通知のサブスクライブ 66
VPC の IPv6 サポート 68
VPC の IPv6 サポートを追加する 69
デュアルスタック VPC の例 74
AWS の IPv6 サポート 76
IPv6 をサポートするサービス 76
追加の IPv6 サポート 90
詳細情報 91
仮想プライベートクラウド 92
VPC の基本 93
VPC の IP アドレスの範囲 93
VPC の図 93
VPC リソース 94
VPC の設定オプション 95
デフォルト VPC 97
デフォルト VPC のコンポーネント 97
デフォルトサブネット 100
デフォルトの VPC とデフォルトのサブネットを使って作業する 101
VPC を作成する 105

VPC と他の VPC リソースを作成する 105
VPC のみを作成する 107
AWS CLI を使用して VPC を作成する 109
VPC 内のリソースを視覚化する 114
CIDR ブロックの追加または削除 116
DHCP オプションセット 118
DHCP とは 118
DHCP オプションセットの概念 119
DHCP オプションセットの使用 123
DNS 属性 127
Amazon DNS について理解する 128
EC2 インスタンスの DNS ホスト名を表示する 133
VPC の DNS 属性の表示と更新 134
ネットワークアドレスの使用状況 135
NAU クォータの超過 136
NAU の計算方法 137
NAU の例 138
VPC サブネットを共有する 140
共有サブネットの前提条件 141
共有サブネットの操作 141
所有者と参加者の請求と計測 144
所有者および参加者の責任と権限 145
AWS リソースと共有 VPC サブネット 148
VPC を別のゾーンに拡張する 149
AWS Local Zones 内のサブネット 150
AWS Wavelength のサブネット 155
AWS Outposts のサブネット 158
VPC の削除 160
コンソールを使用して削除する 160
CLIを使用して削除する 161
コンソールアクションから IaC を生成する 162
サブネット 164
サブネットの基本 164
サブネット IP アドレス範囲 164
サブネットタイプ 165
サブネットの図表 166

サブネットのルーティング ... 166
サブネットの設定 ... 166
サブネットのセキュリティ ... 167
サブネットの作成 ... 168
サブネットからの IPv6 CIDR ブロックを追加または削除する ... 170
サブネットの IP アドレス指定属性を変更する ... 170
サブネット CIDR 予約 ... 172
コンソールを使用してサブネット CIDR 予約を操作する ... 173
AWS CLI を使用してサブネット CIDR 予約を操作する ... 173
ルートテーブル ... 174
ルートテーブルの概念 ... 175
サブネットのルートテーブル ... 177
ゲートウェイルートテーブル ... 184
ルーティングの優先度 ... 187
ルーティングオプションの例 ... 190
ルートテーブルとルートを作成する ... 205
サブネットルートテーブルを管理する ... 207
メインルートテーブルの置換 ... 211
ルートテーブルをゲートウェイに関連付ける ... 212
ローカルルートのターゲットを置換または復元する ... 213
高度なルーティング ... 214
到達可能性に関する問題のトラブルシューティング ... 263
ミドルボックスルーティングウィザード ... 263
ミドルボックスルーティングウィザードの前提条件 ... 264
VPC 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 ... 264
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 ... 267
ミドルボックスシナリオ ... 268
サブネットを削除する ... 278
VPC に接続する ... 279
インターネットゲートウェイ ... 281
インターネットゲートウェイの基本 ... 281
インターネットゲートウェイを作成する ... 284
インターネットゲートウェイを削除する ... 287
Egress-Only インターネットゲートウェイ ... 288
Egress-Only インターネットゲートウェイの基本 ... 289
サブネットへの Egress-Only インターネットアクセスの追加 ... 290

NAT デバイス ... 293
NAT ゲートウェイ ... 295
NAT インスタンス ... 349
NAT デバイスの比較 ... 362
Elastic IP アドレス ... 365
Elastic IP アドレスの概念とルール ... 366
Elastic IP アドレスの使用を開始する ... 368
AWS Transit Gateway ... 377
AWS Virtual Private Network ... 378
VPC ピアリング接続 ... 379
モニタリング ... 381
VPC Flow Logs ... 382
フローログの基礎 ... 383
フローログレコード ... 386
フローログレコードの例 ... 405
フローログの制限事項 ... 416
料金 ... 421
フローログの使用 ... 421
CloudWatch Logs への発行 ... 425
Amazon S3 に発行する ... 433
Amazon Data Firehose への発行 ... 442
Athena を使用したクエリ ... 451
VPC フローログの Identity and Access Management ... 455
のトラブルシューティング ... 464
CloudWatch のメトリクス ... 468
NAU メトリクスとディメンション ... 468
NAU の監視を有効または無効にする ... 471
NAU CloudWatch アラームの例 ... 472
請求および使用状況レポート ... 473
IP アドレス管理 ... 474
VPC エンドポイント ... 474
Transit Gateway ... 475
ネットワーク分析 ... 476
トラフィックのミラーリング ... 476
VPC Lattice ... 476
クロスアカウント/リージョンのリソース ... 477

VPC ネットワークを説明する 478
地理的場所 479
サブネット 480
ネットワーク接続 481
セキュリティコントロール 486
トラフィック管理 488
関連リソース 491
セキュリティ 492
データ保護 493
ネットワーク間のトラフィックのプライバシー 494
転送時の VPC 暗号化を強制する 494
暗号化コントロールのモード 495
トラフィックフローの暗号化ステータスのモニタリング 496
VPC 暗号化コントロールの除外 497
実装ワークフロー 498
VPC 暗号化コントロールの状態 498
AWS サービスのサポートと互換性 499
料金 3
AWS CLI コマンドリファレンス 504
その他のリソース 505
Identity and access management 505
対象者 505
ID で認証する 506
ポリシーを使用してアクセスを管理する 507
Amazon VPC で IAM を使用する方法 509
ポリシーの例 513
のトラブルシューティング 527
AWS 管理ポリシー 528
サービスにリンクされたロールの使用 532
インフラストラクチャセキュリティ 537
ネットワークの隔離 538
ネットワークトラフィックの制御 538
セキュリティグループとネットワーク ACL を比較する 539
セキュリティグループ 541
セキュリティグループの基本 542
セキュリティグループの例 544

セキュリティグループのルール ... 545
デフォルトのセキュリティグループ ... 551
セキュリティグループの作成 ... 553
セキュリティグループのルールを設定する ... 555
セキュリティグループを削除する ... 557
セキュリティグループを複数の VPCに関連付ける ... 558
AWS Organizations とセキュリティグループを共有する ... 561
ネットワーク ACL ... 567
ネットワーク ACL の基本 ... 568
ネットワーク ACL ルール ... 570
デフォルトのネットワーク ACL ... 571
カスタムネットワーク ACL ... 573
パス MTU 検出 ... 579
ネットワーク ACL の作成 ... 580
ネットワーク ACL の関連付けを管理する ... 583
ネットワーク ACL を削除する ... 586
例: サブネットのインスタンスへのアクセス制御 ... 587
レジリエンス ... 591
コンプライアンス検証 ... 592
VPC とサブネットへのパブリックアクセスをブロックする ... 593
VPC BPA の基本 ... 594
VPC BPA の影響を評価し、VPC BPA をモニタリングする ... 600
高度な例 ... 605
ベストプラクティス ... 660
他のサービスで使用する ... 662
AWS PrivateLink ... 663
AWS Network Firewall ... 664
Route 53 Resolver DNS Firewall ... 666
Reachability Analyzer ... 667
例 ... 669
テスト環境 ... 670
概要: ... 670
1. VPC を作成する ... 673
2. アプリケーションをデプロイします ... 674
3. 設定をテストする ... 674
4. クリーンアップ ... 674

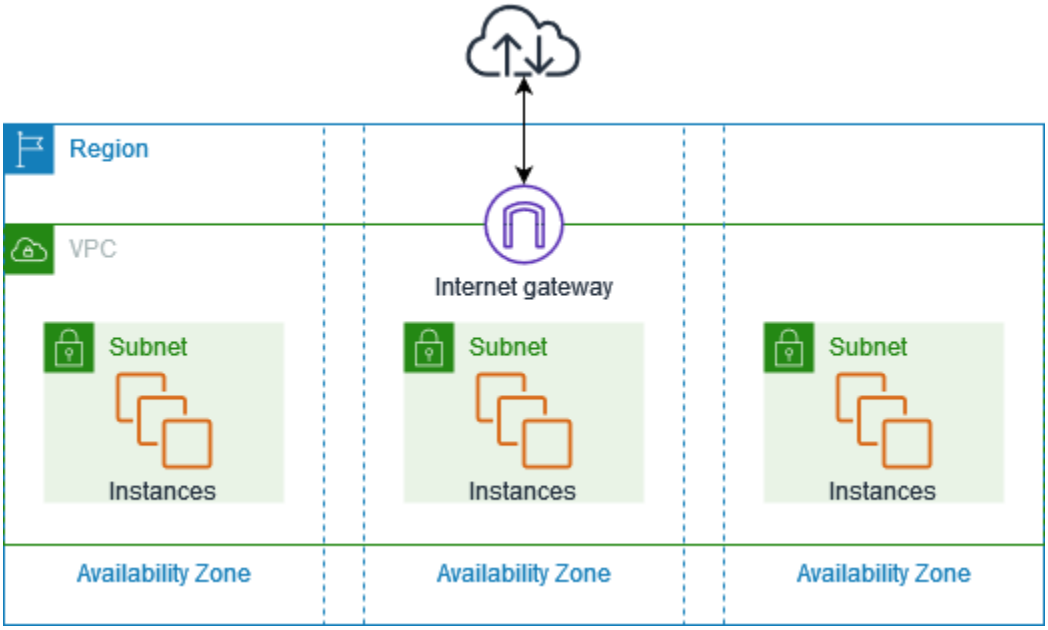
ウェブサーバーとデータベースサーバー ... 674
概要: ... 675
1. VPC を作成する ... 679
2. アプリケーションをデプロイします ... 680
3. 設定をテストする ... 680
4. クリーンアップ ... 681
プライベートサーバー ... 681
概要 ... 681
1. VPC を作成する ... 684
2. アプリケーションをデプロイします ... 685
3. 設定をテストする ... 686
4. クリーンアップ ... 686
チュートリアル ... 687
AWS CLI を使用した Amazon VPC の開始方法 ... 687
前提条件 ... 688
「VPC を作成する」 ... 688
サブネットを作成する ... 689
インターネット接続を設定する ... 691
NAT ゲートウェイを作成する ... 692
サブネット設定を構成する ... 693
セキュリティグループを作成する ... 694
VPC 設定を検証する ... 695
EC2 インスタンスをデプロイする ... 696
トラブルシューティング ... 229
リソースをクリーンアップする ... 699
本番環境への移行 ... 701
次のステップ ... 701
AWS CLI を使用してプライベートサブネットおよび NAT ゲートウェイを持つ VPC を作成する ... 702
前提条件 ... 688
VPC およびサブネットを作成する ... 704
インターネット接続を作成および設定する ... 706
NAT ゲートウェイを作成する ... 708
Amazon S3 の VPC エンドポイントを作成する ... 709
セキュリティグループの設定 ... 710
EC2 インスタンスの起動テンプレートを作成する ... 711

ロードバランサーおよびターゲットグループを作成する 712
Auto Scaling グループを作成する 714
設定をテストする 714
リソースをクリーンアップする 699
次のステップ 701
クォータ 718
VPC とサブネット 718
DNS 719
Elastic IP アドレス 719
ゲートウェイ 720
カスタマーマネージドプレフィックスリスト 720
ネットワーク ACL 722
ネットワークインターフェイス 722
ルートテーブル 723
ルートサーバー数 724
セキュリティグループ 727
VPC サブネット共有 728
ネットワークアドレスの使用状況 729
Amazon EC2 API スロットリング 729
その他のクォータリソース 730
ドキュメント履歴 731

Amazon VPC とは?

Amazon Virtual Private Cloud (Amazon VPC) を使用すると、論理的に隔離されている定義済みの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、AWS 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。

次の図表は、VPC の例を示しています。VPC には、リージョンの各アベイラビリティーゾーンに 1 つのサブネット、各サブネットに EC2 インスタンス、VPC 内のリソースとインターネットとの通信を可能にするインターネットゲートウェイがあります。



詳細については、Amazon Virtual Private Cloud (Amazon VPC) を参照してください。

機能

次の機能は、アプリケーションに必要な接続性を実現するよう VPC を設定するのに役立ちます。

仮想プライベートクラウド (VPC)

VPC は、お客様自身のデータセンターで運用されている従来のネットワークによく似た仮想ネットワークです。VPC の作成後、サブネットを追加できます。

サブネット

サブネットは、VPC の IP アドレスの範囲です。サブネットは、1 つのアベイラビリティーゾーンに存在する必要があります。サブネットを追加した後、VPC で AWS リソースをデプロイできます。

IP アドレス指定

IPv4 と IPv6 の両方の IP アドレスを VPC およびサブネットに割り当てることができます。また、AWS で パブリック IPv4 アドレスおよび IPv6 GUA アドレスを使用して、EC2 インスタンス、NAT ゲートウェイ、Network Load Balancer などの VPC 内のリソースに割り当てることもできます。

ルーティング

ルートテーブルを使用して、サブネットやゲートウェイからのネットワークトラフィックの転送先を指定します。

ゲートウェイとエンドポイント

ゲートウェイは、VPC を別のネットワークに接続します。例えば、インターネットゲートウェイを使用して、VPC をインターネットに接続できます。VPC エンドポイントを使用すると、インターネットゲートウェイや NAT デバイスを使用せずに、プライベートで AWS のサービス に接続できます。

ピアリング接続

VPC ピアリング接続を使用すると、2 つの VPC 内のリソース間でトラフィックをルーティングできます。

トラフィックのミラーリング

ネットワークインターフェイスからネットワークトラフィックをコピーし、それらをセキュリティおよびモニタリングのアプライアンスに送信することで、ディープパケットインスペクションを行えます。

Transit Gateway

中央のハブとして機能する Transit Gateway を使用すると、VPC、VPN 接続、Direct Connect 接続間のトラフィックをルーティングできます。

VPC フローログ

フローログ は、VPC のネットワークインターフェイスに出入りする IP トラフィックに関する情報をキャプチャします。

VPN 接続

AWS Virtual Private Network (Site-to-Site VPN) を使用して、VPC をオンプレミスネットワークに接続できます。

Amazon VPC の開始方法

AWS アカウント の各 AWS リージョン に、デフォルトの VPC が含まれています。デフォルトの VPC は、EC2 インスタンスの起動と接続をすぐに開始できるように設定されています。詳細については、「VPC のプランニング」を参照してください。

必要なサブネット、IP アドレス、ゲートウェイ、ルーティングを使用して、追加の VPC を選択できます。詳細については、「the section called “VPC を作成する”」を参照してください。

Amazon EC2 の使用

次のインターフェイスのいずれかを使用して、VPC を作成および管理できます。

- AWS マネジメントコンソール — 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。(詳しくは、AWS Command Line Interface 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、AWSSDK 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」の「Amazon VPC アクション」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。ただし、NAT ゲートウェイ、IP Address Manager、トラフィックミラーリング、Reachability Analyzer、Network Access Analyzer 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。

仮想プライベートクラウド (VPC) で起動するほぼすべてのリソースでは、接続用の IP アドレスが提供されます。VPC 内のほとんどのリソースでは、プライベート IPv4 アドレスを使用します。ただし、IPv4 経由でインターネットに直接アクセスする必要があるリソースでは、パブリック IPv4 アドレスを使用します。

Amazon VPC では、VPC の事前設定を必要とすることなく、Elastic Load Balancing、Amazon RDS、および Amazon EMR などのマネージドサービスを起動できます。これらは、アカウントのデフォルト VPC (存在する場合) を使用して起動されます。マネージドサービスによってアカウントにプロビジョニングされたパブリック IPv4 アドレスには、料金が発生します。これらの料金は、AWS Cost and Usage Report の Amazon VPC サービスに関連付けられます。

パブリック IPv4 アドレスの料金

パブリック IPv4 アドレスは、インターネットからルーティング可能な IPv4 アドレスです。インターネットから IPv4 経由でリソースに直接アクセスするには、パブリック IPv4 アドレスが必要です。

既存または新規の AWS 無料利用枠のお客様は、EC2 サービスでのパブリック IPv4 アドレスの使用が 750 時間無料になります。EC2 サービスを AWS 無料利用枠で使用していない場合は、パブリック IPv4 アドレスの料金が請求されます。具体的な料金情報については、「Amazon VPC の料金」の [パブリック IPv4 アドレス] タブを参照してください。

プライベート IPv4 アドレス (RFC 1918) の利用には料金がかかりません。共有 VPC に対するパブリック IPv4 アドレスの課金方法の詳細については、「Billing and metering for the owner and participants」を参照してください。

パブリック IPv4 アドレスには次のタイプがあります。

- Elastic IP アドレス (EIP): Amazon が提供する静的なパブリック IPv4 アドレスであり、EC2 インスタンス、Elastic Network Interface、または AWS リソースと関連付けることができます。
- EC2 パブリック IPv4 アドレス: Amazon が EC2 インスタンスに割り当てるパブリック IPv4 アドレスです (デフォルトのサブネットで EC2 インスタンスが起動された場合、またはパブリック IPv4 アドレスを自動的に割り当てるように設定されたサブネットでインスタンスが起動された場合)。
- BYOIPv4 アドレス: Bring-Your-Own-IP (BYOIP) 機能により AWS に持ち込む IPv4 アドレス範囲のパブリック IPv4 アドレスです。
- サービスマネージド IPv4 アドレス: AWS サービスにより自動的に AWS リソースにプロビジョニングされて管理されるパブリック IPv4 アドレスです。例えば、Amazon ECS、Amazon RDS、Amazon WorkSpaces のパブリック IPv4 アドレスなどです。

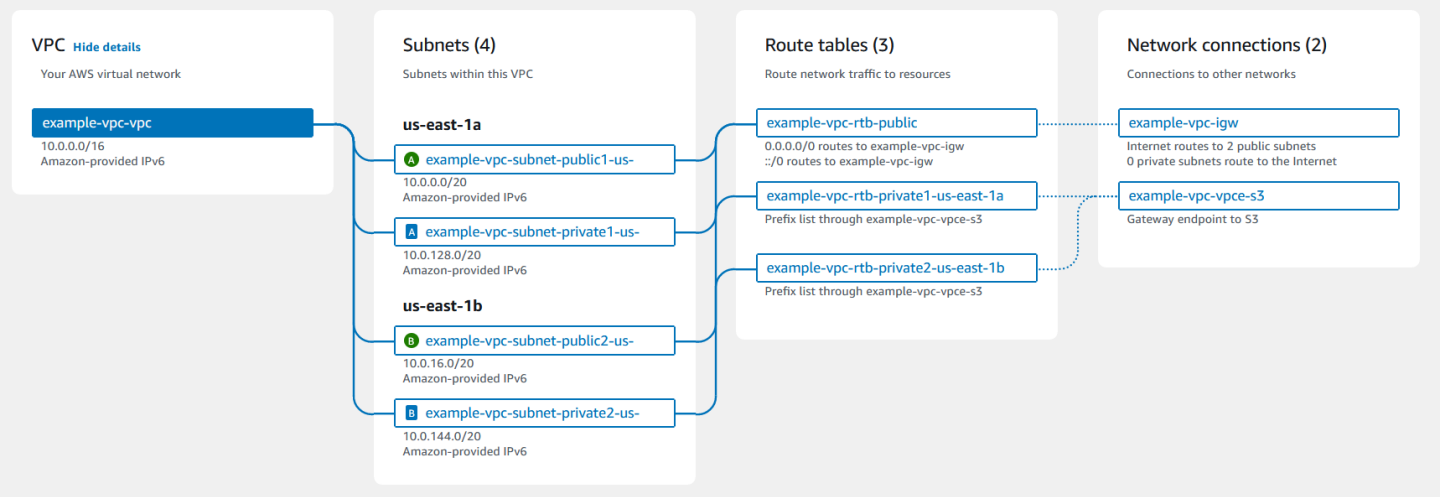
パブリック IPv4 アドレスを使用できる最も一般的な AWS サービスを以下に示します。

- Amazon WorkSpaces Applications
- AWS Client VPN
- AWS Database Migration Service
- Amazon EC2
- Amazon Elastic Container Service
- Amazon EKS
- Amazon EMR
- Amazon GameLift Servers
- AWS Global Accelerator
- AWS Mainframe Modernization
- Amazon Managed Streaming for Apache Kafka
- Amazon MQ
- Amazon RDS
- Amazon Redshift
- AWS Site-to-Site VPN
- Amazon VPC NAT ゲートウェイ
- Amazon WorkSpaces
- エラスティックロードバランシング

Amazon VPC の仕組み

Amazon Virtual Private Cloud (Amazon VPC) を使用すると、論理的に隔離されている定義済みの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、AWS 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。

以下は、AWS マネジメントコンソール を使用した VPC の作成時に表示される [プレビュー] ペインの VPC とそのリソースを視覚的に表現したものです。既存の VPC の場合、この図には [リソースマップ] タブからアクセスできます。この例では、VPC と他のネットワークリソースの作成を選択した際に [VPC を作成] ページで最初に選択されるリソースを示しています。この VPC は、1 つの IPv4 CIDR および Amazon が提供した 1 つの IPv6 CIDR、2 つのアベイラビリティーゾーンのサブネット、3 つのルートテーブル、1 つのインターネットゲートウェイ、1 つのゲートウェイエンドポイントで構成されています。この図では、インターネットゲートウェイを選択したことで、対応するルートテーブルが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、パブリックサブネットからのトラフィックがインターネットにルーティングされていることを示しています。



概念

- VPC とサブネット
- デフォルトの VPC とデフォルト以外の VPC
- ルートテーブル
- インターネットへのアクセス
- 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
- VPC とネットワークの接続

- AWS プライベートグローバルネットワーク

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 の IP アドレス範囲を指定して、サブネットを追加し、ゲートウェイを追加して、セキュリティグループを関連付けます。

サブネットは、VPC の IP アドレスの範囲です。Amazon EC2 インスタンスなどの AWS リソースを VPC サブネット内部で起動します。サブネットをインターネット、他の VPC、および独自のデータセンターに接続し、ルートテーブルを使用してサブネット間でトラフィックをルーティングできます。

詳細情報

- IP アドレス指定
- 仮想プライベートクラウド
- サブネット

デフォルトの VPC 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、各リージョンにデフォルトの VPC があります。デフォルトの VPC は設定済みで、すぐに使用できます。例えば、この VPC には、リージョンの各アベイラビリティーゾーンのデフォルトサブネット、アタッチされたインターネットゲートウェイ、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送信するメインルートテーブルのルート、およびパブリック IP アドレスを持つインスタンスに DNS ホスト名を自動的に割り当てて、Amazon が提供する DNS サーバーを介して DNS 解決を有効にする DNS 設定が含まれています (「VPC の DNS 属性」を参照してください)。そのため、デフォルトサブネットで起動された EC2 インスタンスは、自動的にインターネットにアクセスできます。リージョンにデフォルトの VPC があり、そのリージョンでの EC2 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、デフォルトサブネットの 1 つを選択して、そのサブネットでインスタンスを起動します。

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

- the section called “デフォルト VPC”
- the section called “VPC を作成する”

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

- ルートテーブルを設定する

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を使用して VPC のインターネットアクセスを有効にする
- Egress-Only インターネットゲートウェイを使用してアウトバウンド IPv6 トラフィックを有効にする
- NAT デバイスを使用してインターネットまたは他のネットワークに接続する

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 を自社の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 Transit Gateway 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。

詳細情報

- AWS Site-to-Site VPN ユーザーガイド

- Amazon VPC Transit Gateway

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、Transit Gateway 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。Transit Gateway 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、Direct Connect ゲートウェイ、および Transit Gateway ピア接続が含まれます。

詳細情報

- Amazon VPC Peering Guide
- Amazon VPC Transit Gateway

AWS プライベートグローバルネットワーク

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。

AWS ネットワークから AWS ネットワークに送信されるパケットは、AWS 中国リージョンおよび AWS 欧州ソブリンクラウドリージョンとの送受信されるトラフィックを除き、AWS グローバルネットワークに残ります。これは、送信先がプライベート IP アドレスであるかパブリック IP アドレスであるかに関係なく該当します。例えば、2 つの VPC の EC2 インスタンスがパブリック IP アドレスを使用して通信する場合、トラフィックはプライベートグローバルネットワークに残ります。送信先は、同じアベイラビリティーゾーン、同じリージョン内の別のアベイラビリティーゾーン、または別のリージョンにある場合があります。

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。

VPC のプランニング

VPC の作成と接続の準備をするには、次のタスクを完了します。完了後、AWS にアプリケーションをデプロイする準備ができます。

タスク

- AWS アカウント へのサインアップ
- アクセス許可の確認
- IP アドレスの範囲を決定する
- アベイラビリティーゾーンの選択
- インターネット接続の計画
- VPC の作成
- アプリケーションをデプロイします

AWS アカウント へのサインアップ

AWS アカウント へのサインアップ

AWS を使い始めるには、AWS アカウントが必要です。AWS アカウントの作成の詳細については、「AWS アカウント管理 リファレンスガイド」の「AWS アカウントの開始方法」を参照してください。

アクセス許可の確認

Amazon VPC を使用するには、事前に必須のアクセス権限が必要です。詳細については、「Amazon VPC の Identity and Access Management」および「Amazon VPC ポリシーの例」を参照してください。

IP アドレスの範囲を決定する

IP アドレスを使用して、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行います。VPC とサブネットの作成時に IP アドレス範囲を選択できます。EC2 インスタンスなどのサブネットにリソースをデプロイすると、サブネットの IP アドレス範囲からの IP アドレスを受信します。詳細については、「IP アドレス指定」を参照してください。

VPC のサイズを選択する際には、AWS アカウント と VPC 全体で必要な IP アドレスの数を検討してください。VPC の IP アドレス範囲がネットワーク自体の IP アドレス範囲と重複しないようにしてください。複数の VPC 間の接続が必要な場合は、IP アドレスが重複していないことを確認する必要があります。

IP Address Manager (IPAM) を使用すると、アプリケーションの IP アドレスの計画、追跡、監視を実行しやすくなります。詳細については、「IP アドレスマネージャーガイド」を参照してください。

アベイラビリティーゾーンの選択

AWS リージョンは、アベイラビリティーゾーンと呼ばれるデータセンターをクラスター化する物理的な場所です。アベイラビリティーゾーンは、独立した電源、冷却、物理セキュリティ、冗長電源、ネットワーク、接続性を備えています。リージョン内のアベイラビリティーゾーンは物理的に距離を持たせており、高帯域幅、低レイテンシーのネットワーキングで相互接続されています。アプリケーションを複数のアベイラビリティーゾーンで実行するように設計することで、耐障害性をさらに高めることができます。

本番環境

本番環境では、少なくとも 2 つのアベイラビリティーゾーンを選択し、AWS リソースをアクティブなアベイラビリティーゾーンそれぞれに均等にデプロイすることをお勧めします。

開発またはテスト環境

開発環境やテスト環境では、1 つのアベイラビリティーゾーンにのみリソースをデプロイするとコストを削減できます。

インターネット接続の計画

接続要件に基づいて、各 VPC をサブネットに分割することを計画します。例えば、次のようになります。

- インターネット上のクライアントからトラフィックを受信するウェブサーバーがある場合は、各アベイラビリティーゾーンにこれらのサーバーのサブネットを作成します。
- また、VPC 内の他のサーバーからのみトラフィックを受信するサーバーがある場合は、これらのサーバー用に、アベイラビリティーゾーンごとの個別のサブネットを作成します。
- ネットワークへの VPN 接続を介してのみトラフィックを受信するサーバーがある場合は、これらのサーバー用に、アベイラビリティーゾーンごとの個別のサブネットを作成します。

アプリケーションがインターネットからトラフィックを受信する場合、VPC にはインターネットゲートウェイが必要です。インターネットゲートウェイを VPC にアタッチしても、インターネットから自動的にインスタンスにアクセスできるようにはなりません。インターネットゲートウェイをアタッチすることに加え、サブネットルートテーブルをインターネットゲートウェイへのルートと共に更新する必要があります。また、インスタンスにパブリック IP アドレスが設定され、アプリケーションに必要な、特定のポートとプロトコルを介したインターネット経由のトラフィックを許可する、関連するセキュリティグループが設定されていることを、確認する必要があります。

代替策として、インターネットに接続されるロードバランサーにインスタンスを登録することもできます。ロードバランサーはクライアントからトラフィックを受信し、1 つ以上のアベイラビリティーゾーンにある登録済みのインスタンス全体にトラフィックを分配します。詳細については、「Elastic Load Balancing」を参照してください。インターネットの未承諾インバウンド接続を許可しない状態で、プライベートサブネット内のインスタンスにインターネット接続を許可する (アップデートのダウンロードなどの場合) には、アクティブなアベイラビリティーゾーンそれぞれにパブリック NAT ゲートウェイを追加し、ルートテーブルを更新して、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ようにします。詳細については、「the section called “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”」を参照してください。

VPC の作成

必要な VPC とサブネットの数、VPC とサブネットに割り当てる CIDR ブロック、VPC をインターネットに接続する方法が決定したら、VPC を作成する準備は完了です。AWS マネジメントコンソール を使用して VPC を作成し、設定にパブリックサブネットを含めると、サブネットのルートテーブルが作成され、インターネットへの直接アクセスに必要なルートが追加されます。詳細については、「the section called “VPC を作成する”」を参照してください。

アプリケーションをデプロイします

VPC を作成したら、アプリケーションをデプロイできます。

本番環境

本番環境では、以下のサービスのいずれかを使用して、複数のアベイラビリティーゾーンにサーバーをデプロイし、アプリケーションに必要なサーバー数を最小限に抑えるようにスケーリングを設定して、サーバーをロードバランサーに登録しトラフィックをサーバー全体に均等に分配することができます。

- Amazon EC2 Auto Scaling

- EC2 Fleet
- Amazon Elastic Container Service (Amazon ECS)

開発またはテスト環境

開発環境またはテスト環境では、単一の EC2 インスタンスを起動することを選択できます。詳細については、「Amazon EC2 ユーザーガイド」の「Amazon EC2 の開始方法」を参照してください。

VPC とサブネットの IP アドレス指定

IP アドレスは、VPC 内のリソースの相互通信と、インターネット経由でのリソースとの通信を可能にします。

Classless Inter-Domain Routing (CIDR) 表記は、IP アドレスとそのネットワークマスクを表す方法です。これらのアドレスの形式は次のとおりです。

- 個々の IPv4 アドレスは 32 ビットで、0～255 の最大 3 桁の 10 進数の 4 つのグループで構成されます。例えば、10.0.1.0 です。
- IPv4 CIDR ブロックには、IPv4 アドレス、それに続くスラッシュ、および 0 から 32 までの数値が含まれます。例えば、10.0.0.0/16 は、10.0.0.0 から 10.0.255.255 までの 65,536 個の IPv4 アドレスを表します。
- 個々の IPv6 のアドレスは、128 ビットで、4 桁の 16 進数の 8 つのセグメントで構成されます。例えば、2001:0db8:85a3:0000:0000:8a2e:0370:7334 です。先頭のゼロをセグメントに含める必要はありません。連続する全ゼロセグメントを、アドレス内で二重コロン (::) に 1 回置き換えることもできます。したがって、サンプルアドレスは 2001:db8:85a3::8a2e:370:7334 として圧縮できます。
- IPv6 CIDR ブロックには、全ゼロセグメントで終わる IPv6 アドレスがあり、全ゼロセグメントは二重コロンに置き換えられ、スラッシュと 0 から 128 の数値が続きます。例えば、2001:db8:1234:1a00::/56 は、2001:db8:1234:1a00:0000:0000:0000:0000 から 2001:db8:1234:1aff:ffff:ffff:ffff:ffff までの 2^72 個の IPv6 アドレスを表します。

詳細については、「CIDR とは?」を参照してください。

内容

- プライベート IPv4 アドレス
- パブリック IPv4 アドレス
- IPv6 アドレス
- 独自の IP アドレスを使用する
- Amazon VPC IP Address Manager を使用する
- VPC CIDR ブロック
- サブネット CIDR ブロック
- IPv4 と IPv6 を比較する

- マネージドプレフィックスリストを使用したネットワーク CIDR ブロックの統合と管理
- AWS IP アドレスの範囲
- VPC の IPv6 サポート
- IPv6 をサポートする AWS サービス

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 アドレスとも呼ばれます) は、インターネット経由では到達できないアドレスで、VPC 内のインスタンス間の通信に使用できます。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プライマリネットワークインターフェイス (例えば 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 ホスト名も割り当てられます。ホスト名には、リソースベースと IP ベースの 2 つのタイプがあります。詳細については、EC2 インスタンスの名前付けを参照してください。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、「Amazon EC2 ユーザーガイド」の「Elastic Network Interface」を参照してください。

VPC 内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されて再起動されたときもネットワークインターフェイスとの関連付けを維持し、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、「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。

プライベート IP アドレスとは、VPC の IPv4 CIDR 範囲内にある IP アドレスを指します。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。インターネットアクセスはゲートウェイ経由で設定する必要があります。例えば、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 Direct Connect などです。

サブネットの IPv4 アドレス範囲がインターネットにアドバタイズされることはありません。

パブリック IPv4 アドレス

すべてのサブネットには、サブネット内に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 アドレスとも呼ばれます) を自動的に受信するかどうかを決定する属性があります。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) を通じ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。

> **Note**
>
> AWS は、実行中のインスタンスに関連付けられているパブリック IPv4 アドレスや Elastic IP アドレスを含めたすべてのパブリック IPv4 アドレスの料金を請求します。詳細については「Amazon VPC の料金」ページにあるパブリック IPv4 アドレスタブを参照してください。

インスタンスがパブリック IP アドレスを取得するかどうかは、以下を実行することで制御できます。

- 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、「サブネットの IP アドレス指定属性を変更する」を参照してください。
- 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。この設定は、サブネットのパブリック IP アドレス指定属性を上書きします。
- パブリック IP アドレスは、ネットワークインターフェースに関連付けられた IP アドレスを管理することで、起動後にインスタンスから割り当て解除できます。詳細については、「Amazon EC2 ユーザーガイド」の「IP アドレスを管理する」を参照してください。

パブリック IP アドレスは Amazon のパブリック IP アドレスプールから割り当てられ、お客様のアカウントには関連付けられていません。パブリック IP アドレスとインスタンスの関連付けを解除すると、アドレスが解放されてプールに戻され、そのアドレスを使用することができなくなります。場合によっては、AWS がお客様のインスタンスからパブリック IP アドレスを解放する、または新しいインスタンスに割り当てる場合があります。詳細については、「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。

必要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永続的なパブリック 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、「Elastic IP アドレスを VPC 内のリソースに関連付ける」を参照してください。

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、「VPC の DNS 属性」を参照してください。

Amazon VPC IP Address Manager (IPAM) を利用している場合は、AWS からパブリック IPv4 アドレスの連続したブロックを取得し、それを使用してシーケンシャル Elastic IP アドレスを AWS リソースに割り当てることができます。連続した IPv4 アドレスブロックを使用すると、セキュリティアクセスコントロールリストの管理オーバーヘッドを大幅に削減し、AWS でスケールする企業のために IP アドレスの割り当てと追跡を簡素化できます。詳細については、「Amazon VPC IPAM ユーザーガイド」の「IPAM プールからシーケンシャル Elastic IP アドレスを割り当てる」を参照してください。

IPv6 アドレス

インターネットが拡大し続けるにつれて、IP アドレスの必要性も高まります。IP アドレスの最も一般的な形式は IPv4 です。IPv6 は IP アドレスの新しい形式で、IPv4 よりも大きなアドレス空間を提供します。IPv6 により、IPv4 アドレス枯渇の問題が解決され、より多くのデバイスをインターネットに接続できるようになります。移行は段階的に行われますが、IPv6 の導入が増えるにつれて、ネットワークを簡素化し、IPv6 の高度な機能を活用して接続性、パフォーマンス、セキュリティを向上させることができます。

Amazon EC2、Amazon S3、Amazon CloudFront などの多くの AWS サービスは、デュアルスタック (IPv4 および IPv6) または IPv6 専用のサポートを提供しており、リソースに IPv6 アドレスを割り当てて IPv6 プロトコル経由でアクセスできます。これにより、IPv6 を導入するお客様のネットワーク設定と管理が簡素化されます。

他のサービスでは、デュアルスタックおよび IPv6 専用の限定的または部分的なサポートを提供しています。

IPv6 をサポートしているサービスの詳細については、「IPv6 をサポートする AWS サービス」を参照してください。

なお、一部の IPv6 アドレスは、インターネットエンジニアリングタスクフォースによって予約されています。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。

> **Note**
>
> AWS ではパブリック IPv6 アドレス指定とプライベート IPv6 アドレス指定の両方が利用できます。AWS では、AWS からインターネットでアドバタイズされる IP アドレスはパブリック IP アドレスとして定義されますが、プライベート IP アドレスは AWS からインターネットでアドバタイズされず、アドバタイズすることもできません。

内容

- パブリック IPv6 アドレス
- プライベート IPv6 アドレス

パブリック IPv6 アドレス

Amazon が提供する IPv6 アドレスは常にインターネットでアドバタイズされます。これ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。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 を使用することで、EC2 インスタンスなどのリソースに IPv6 アドレスを使用してアクセスできるかどうかを制御できます。

以下に、パブリック IPv6 アドレスをワークロードに使用するための準備方法をいくつか紹介します。

- Amazon VPC IP Address Manager で IPAM を作成し、Amazon 所有のパブリック IPv6 アドレス範囲を IPAM アドレスプールにプロビジョニングします。詳細については、「Amazon VPC IPAM ユーザーガイド」の「IPv6 プールを作成する」を参照してください。
- IPAM があり、パブリック IPv6 アドレス範囲を所有している場合は、パブリック IPv6 アドレス範囲の一部またはすべてを IPAM に持ち込み、パブリック IPv6 アドレス範囲を IPAM アドレスプールにプロビジョニングします。詳細については、「Amazon VPC IPAM ユーザーガイド」の「チュートリアル: IP アドレスを IPAM に移行する」を参照してください。
- IPAM はないが、パブリック IPv6 アドレス範囲を所有している場合は、パブリック IPv6 アドレス範囲の一部またはすべてを AWS に持ち込みます。詳細については、「Amazon EC2 ユーザーガイド」の「Amazon EC2 に自分の IP アドレスを持ち込む (BYOIP)」を参照してください。

パブリック IPv6 アドレスを使用する準備ができたら、インスタンスにパブリック IPv6 アドレスを割り当てたり (「Amazon EC2 ユーザーガイド」の「EC2 インスタンスの IPv6 アドレスを管理する」を参照)、パブリック IPv6 CIDR ブロックを VPC に割り当てたり (「CIDR ブロックを追加するまたは VPC から削除する」を参照)、IPv6 CIDR ブロックをサブネットに関連付けたり (「サブネットの IP アドレス指定属性を変更する」を参照) することができます。

プライベート IPv6 アドレス

プライベート IPv6 アドレスは、AWS からインターネットでアドバタイズされず、そしてアドバタイズできない IPv6 アドレスです。

プライベートネットワークで IPv6 をサポートし、このアドレスからインターネットにトラフィックをルーティングするつもりがない場合は、プライベート IPv6 アドレスを使用できます。プライベート IPv6 アドレスを持つリソースからインターネットに接続する場合、それは可能ですが、そのためにはパブリック IPv6 アドレスを持つ別のサブネットのリソースを介してトラフィックをルーティングする必要があります。

プライベート IPv6 アドレスには 2 つのタイプがあります。

- IPv6 ULA 範囲: RFC4193 で定義されている IPv6 アドレス。このアドレス範囲は常に「fc」または「fd」で始まり、簡単に識別できます。有効な IPv6 ULA スペースは、Amazon の予約範囲 fd00::/16 と重複しない fd00::/8 より下のいずれかです。
- IPv6 GUA 範囲: RFC3587 で定義されている IPv6 アドレス。IPv6 GUA 範囲をプライベート IPv6 アドレスとして使用するオプションはデフォルトで無効になっているため、使用する前に有効にする必要があります。詳細については、「Amazon VPC IPAM ユーザーガイド」の「プライベート IPv6 GUA CIDR のプロビジョニングを有効にする」を参照してください。

次の点に注意してください。

- プライベート IPv6 アドレスは、Amazon VPC IP Address Manager (IPAM) 経由でのみ使用できます。IPAM は、IPv6 ULA および GUA アドレスを持つリソースを検出し、プールをモニタリングして、IPv6 ULA および GUA アドレス空間が重複していないかを確認します。
- プライベート IPv6 GUA 範囲を使用するときは、独自に所有する IPv6 GUA 範囲を使用する必要があります。
- プライベート IPv6 アドレスは、AWS によってインターネットでアドバタイズされることはなく、そしてアドバタイズすることができません。AWS では、VPC にインターネットゲートウェイまたは Egress Only インターネットゲートウェイがある場合でも、プライベート IPv6 範囲から

のパブリックインターネットへの直接送信を許可しません。プライベート IPv6 アドレスは、インターネットゲートウェイエッジで自動的にドロップされ、パブリックにルーティングされません。

- AWS では、最初の 4 つのサブネットプライベート IPv6 アドレスと最後のサブネットプライベート IPv6 アドレスが予約されます。
- プライベート IPv6 ULA の有効範囲は、fd80::/9 以降の /9 ~ /60 です。
- VPC にプライベート IPv6 GUA 範囲が割り当てられている場合、同じ VPC 内のプライベート IPv6 GUA スペースと重複するパブリック IPv6 GUA スペースを使用することはできません。
- プライベート IPv6 ULA および GUA アドレス範囲を持つリソース間の通信がサポートされています (Direct Connect、VPC ピアリング、Transit Gateway、および VPN 接続間の通信など)。
- プライベート IPv6 アドレスは、IPv6 のみ、およびデュアルスタックの VPC サブネット、Elastic Load Balancer、AWS Global Accelerator エンドポイントで使用できます。
- プライベート IPv6 アドレスは無料です。

ワークロードにプライベート IPv6 アドレスを使用するための準備方法の一部を以下に示します。

- Amazon VPC IP Address Manager で IPAM を作成し、IPAM アドレスプールにプライベート IPv6 ULA 範囲をプロビジョニングします。詳細については、「Amazon VPC IPAM ユーザーガイド」の「IPv6 プールを作成する」を参照してください。
- Amazon VPC IP Address Manager で IPAM を作成し、IPAM アドレスプールにプライベート IPv6 GUA 範囲をプロビジョニングします。IPv6 GUA 範囲をプライベート IPv6 アドレスとして使用するオプションはデフォルトで無効になっているため、使用する前に IPAM で有効にする必要があります。詳細については、「Amazon VPC IPAM ユーザーガイド」の「プライベート IPv6 GUA CIDR のプロビジョニングを有効にする」を参照してください。

プライベート IPv6 アドレスを使用する準備ができたら、IPAM プールから VPC にプライベート IPv6 CIDR ブロックを割り当てたり (「CIDR ブロックを追加するまたは VPC から削除する」を参照)、IPv6 CIDR ブロックをサブネットに関連付けたり (「サブネットの IP アドレス指定属性を変更する」を参照) することができます。

独自の 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。アドレス範囲は引き続き所有できますが、AWS はデフォルトで、この範囲をインターネット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

範囲が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 アドレスプールからの 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、「Amazon EC2 ユーザーガイド」の「自分の IP アドレスを持ち込む (BYOIP)」を参照してください。

Amazon VPC IP Address Manager を使用する

Amazon VPC IP Address Manager (IPAM)は、AWS ワークロードの IP アドレスを計画、追跡、監視しやすくする VPC 機能です。IPAM を使用すると、特定のビジネスルールを使用して IP アドレス CIDR を VPC に割り当てることができます。

詳細については、「Amazon VPC IPAM ユーザーガイド」の「IPAM とは」を参照してください。

VPC CIDR ブロック

仮想プライベートクラウド (VPC) の IP アドレスは、Classless Inter-Domain Routing (CIDR) 表記で表されます。VPC には、関連付けられた IPv4 CIDR ブロックがある必要があります。オプションで、追加の IPv4 CIDR ブロックと、1 つ以上の 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、「VPC とサブネットの IP アドレス指定」を参照してください。

内容

- IPv4 VPC CIDR ブロック
- VPC の IPv4 CIDR ブロックを管理する
- IPv4 CIDR ブロック関連付けの制限
- IPv6 VPC CIDR ブロック

IPv4 VPC CIDR ブロック

VPC を作成するときに、VPC に IPv4 CIDR ブロックを指定する必要があります。許容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) の間のサイズです。VPC を作成したら、VPC と追加の IPv4 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、「CIDR ブロックを追加するまたは VPC から削除する」を参照してください。

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)	10.0.0.0/16
172.16.0.0 ~ 172.31.255.255 (172.16/12 プレフィックス)	172.31.0.0/16
192.168.0.0 ~ 192.168.255.255 (192.168/16 プレフィックス)	192.168.0.0/20

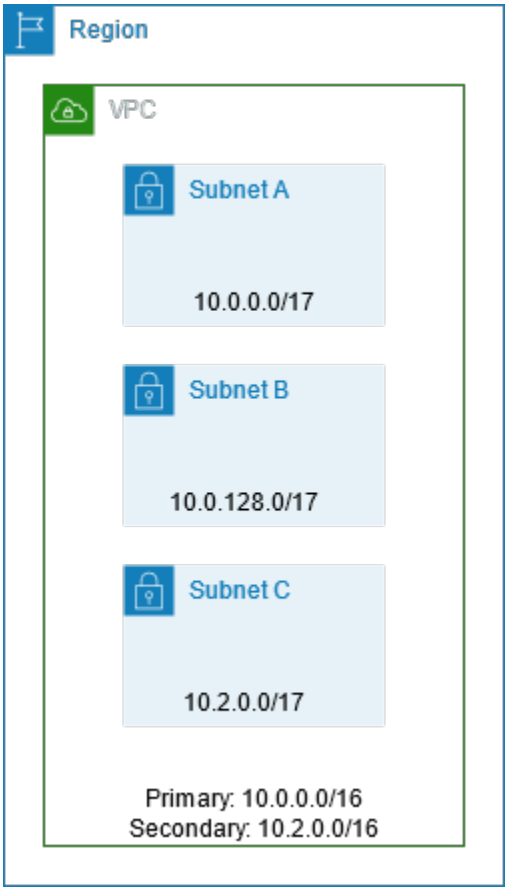
考慮事項

- VPC に次の CIDR ブロックを指定することはできません。
  - 0.0.0.0/8
  - 127.0.0.0/8 (内部ホストループバックアドレス範囲)
  - 169.254.0.0/16 (リンクローカルアドレス範囲)
  - 224.0.0.0/4 (マルチキャストアドレス範囲)
- AWS サービスでの使用向けに VPC を作成する場合は、サービスドキュメントを参照して、その設定に特定の要件があるかどうかを確認します。
- 一部の AWS サービスでは、172.17.0.0/16 CIDR 範囲が使用されています。IP アドレス範囲がネットワークで既に使用されている場合、サービスで IP アドレスの競合が発生する可能性があります。例えば、AWS Cloud9 と Amazon SageMaker AI は 172.17.0.0/16 を使用します。競合が発生しないように、VPC を作成するときはこの範囲を使用しないでください。詳細については、「AWS Cloud9 ユーザーガイド」の「VPC の IP アドレスを Docker が使用しているため、EC2 環境に接続できません」を参照してください。
- RFC 1918 に指定されているプライベート IPv4 アドレスの範囲に含まれない、パブリックルーティングが可能な CIDR ブロックで VPC を作成できます。ただし、このドキュメントで使用される「プライベート IP アドレス」とは、VPC の CIDR 範囲に含まれる IPv4 アドレスのことです。
- コマンドラインツールや Amazon EC2 API を使用して VPC を作成する場合、CIDR ブロックは正規形式に自動変更されます。例えば、CIDR ブロックに 100.68.0.18/18 を指定した場合、100.68.0.0/18 の CIDR ブロックが作成されます。

VPC の IPv4 CIDR ブロックを管理する

セカンダリ IPv4 CIDR ブロックを VPC に関連付けることができます。VPC に CIDR ブロックを関連付けると、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 (ルーティング先は CIDR ブロックで、ターゲットは `local`)。

次の例では、VPC にプライマリ CIDR ブロックとセカンダリ CIDR ブロックの両方があります。Subnet A および Subnet B の CIDR ブロックは、プライマリ VPC CIDR ブロックからのものです。Subnet C の CIDR ブロックは、セカンダリ VPC CIDR ブロックからのものです。



次のルートテーブルは、VPC のローカルルートを示しています。

ルーティング先	ターゲット
10.0.0.0/16	ローカル
10.2.0.0/16	ローカル

CIDR ブロックを VPC に追加するときは、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、「IPv4 CIDR ブロック関連付けの制限」を参照してください。
- 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、「Amazon VPC クォータ」を参照してください。
- 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、`10.0.0.0/24` のルーティング先が仮想プライベートゲートウェイになっている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可能です。
- 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ときは、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。
- 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、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` 状態になると利用可能になります。

VPC に関連付けた CIDRブロックの関連付けを解除できます。ただし、VPC の作成当初に使用した CIDR ブロック (プライマリ 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 のチェックボックスをオンにして [VPC] を選択し、[CIDR] タブを選択します。AWS CLI を使用してプライマリ CIDR を表示するには、次の describe-vpcs コマンドを使用します。プライマリ CIDR は最上位の `CidrBlock` `element` で返されます。

```
aws ec2 describe-vpcs --vpc-id vpc-1a2b3c4d --query Vpcs[*].CidrBlock --output text
```

以下は出力の例です。

```
10.0.0.0/16
```

IPv4 CIDR ブロック関連付けの制限

次の表には、既存の VPC CIDR ブロックの許可および制限された VPC CIDR ブロック関連付けの概要が記載されています。制限の理由は、一部の AWS サービスでは、AWS サービス側で競合しない CIDR ブロックを必要とするクロス VPC およびクロスアカウント機能を利用しているためです。

既存の IPv4 アドレス範囲	制限される関連付け	許可される関連付け
10.0.0.0/8	他の RFC 1918* 範囲 (172.16.0.0/12 と 192.168.0.0/16) からの CIDR ブロック。 VPC に関連付けられたいずれかの CIDR ブロックが 10.0.0.0/15 の範囲 (10.0.0.0 ~ 10.1.255.255) である場合、10.0.0.0/16 の範囲 (10.0.0.0 ~ 10.0.255.255) から CIDR ブロックを追加することはできません。	制限されていない /16 ネットマスクから /28 ネットマスクの間の 10.0.0.0/8 範囲にある他の CIDR ブロック。 /16 ネットマスクと /28 ネットマスクの間で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 範囲にある /16 ネットマスクから /28 ネットマスクの間の CIDR ブロック。

既存の IPv4 アドレス範囲	制限される関連付け	許可される関連付け
	198.19.0.0/16 範囲からの CIDR ブロック。	
169.254.0.0/16	「リンクローカル」ブロックの CIDR ブロックは、RFC 5735 で説明されているように、予約されているため、VPC に割り当てることはできません。	
172.16.0.0/12	他の RFC 1918* 範囲 (10.0.0.0/8 と 192.168.0.0/16) からの CIDR ブロック。 172.31.0.0/16 範囲からの CIDR ブロック。 198.19.0.0/16 範囲からの CIDR ブロック。	制限されていない /16 ネットマスクから /28 ネットマスクの間の 172.16.0.0/12 範囲にある他の CIDR ブロック。 /16 ネットマスクと /28 ネットマスクの間で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 範囲にある /16 ネットマスクから /28 ネットマスクの間の CIDR ブロック。
192.168.0.0/16	他の RFC 1918* 範囲 (10.0.0.0/8 と 172.16.0.0/12) からの CIDR ブロック。 198.19.0.0/16 範囲からの CIDR ブロック。	/16 ネットマスクと /28 ネットマスクの間の 192.168.0.0/16 範囲にある他の CIDR ブロック。 /16 ネットマスクと /28 ネットマスクの間のパブリックにルーティングが可能な IPv4 CIDR ブロック (RFC 1918 以外)、または /16 ネットマスクから /28 ネットマスクの間の 100.64.0.0/10 範囲からの CIDR ブロック。

既存の IPv4 アドレス範囲	制限される関連付け	許可される関連付け
198.19.0.0/16	RFC 1918* 範囲からの CIDR ブロック。	/16 ネットマスクと /28 ネットマスクの間で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 範囲にある /16 ネットマスクから /28 ネットマスクの間の CIDR ブロック。
パブリックルーティングが可能な CIDR ブロック (非 RFC 1918)、または 100.64.0.0/10 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 範囲からの CIDR ブロック。	/16 ネットマスクと /28 ネットマスクの間のパブリックルーティングが可能なその他すべての IPv4 CIDR ブロック (非 RFC 1918)、または 100.64.0.0/10 範囲 からの /16 ネットマスクから /28 ネットマスクの間の CIDR ブロック。 また、いずれかの RFC 1918 範囲に CIDR を関連付けることもできますが、これを行うには、まず VPC の作成時にその CIDR を追加し、次に RFC 1918 以外の CIDR を追加する必要があります。

* RFC 1918 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。

IPv6 VPC CIDR ブロック

新しい VPC を作成する場合、単一の IPv6 CIDR ブロックを関連付けるか、または /44 から /60 の、/4 刻みで最大 5 つの IPv6 CIDR ブロックを関連付けることができます。Amazon の IPv6 アドレスプールからの IPv6 CIDR ブロックをリクエストできます。詳細については、「CIDR ブロックを追加するまたは VPC から削除する」を参照してください。

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。詳細については、「the section called “IPv6 のサブネットのサイズ設定”」を参照してください。

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。Amazon は IPv6 CIDR ブロック `2001:db8:1234:1a00::/56` を VPC に割り当てます。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て、この範囲からの IPv6 CIDR ブロック (`2001:db8:1234:1a00::/64` など) を関連付けることができます。

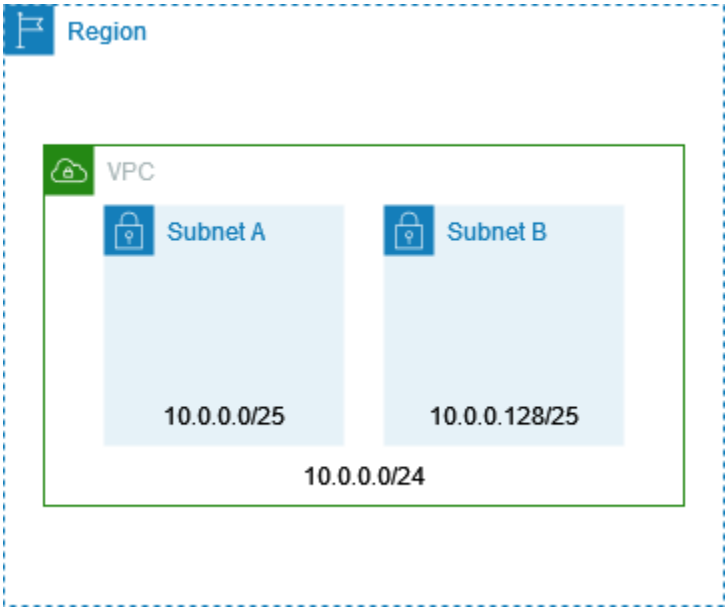
IPv6 CIDR ブロックと VPC の関連付けを解除できます。VPC から IPv6 CIDR ブロックの関連付けを解除すると、その後で IPv6 CIDR ブロックと VPC を再度関連付けるときに同じ CIDR を受け取ることは見込めません。

サブネット CIDR ブロック

サブネットの IP アドレスは、Classless Inter-Domain Routing (CIDR) 表記で表されます。サブネットの CIDR ブロックは、VPC の CIDR ブロック (VPC に 1 つのサブネットを作成するため)、または VPC の CIDR ブロックのサブネット (VPC に複数のサブネットを作成するため) と同じものにすることができます。

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 および IPv6 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 IPv4 および IPv6 CIDR ブロックを特定することもできます。

IPv4 のサブネットのサイズ設定

サブネットで許可される IPv4 CIDR ブロックサイズは、/28 ネットマスクから /16 ネットマスクの間です。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、EC2 インスタンスなどのリソー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: AWS が VPC ルーター用に予約しています。
- 10.0.0.2: AWS が予約しています。DNS 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。(詳しくは、「Amazon DNS サーバー」を参照してください。)
- 10.0.0.3: 将来の利用のために AWS が予約しています。
- 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 を指定した場合、100.68.0.0/18 の CIDR ブロックが作成されます。

BYOIP を用いて AWS に IPv4 アドレス範囲を使用する場合、最初のアドレス (ネットワークアドレス) と最後のアドレス (ブロードキャストアドレス) を含む、その範囲内のすべての IP アドレスを使用できます。

IPv6 のサブネットのサイズ設定

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。可能な IPv6 のネットマスク長は、/44 から /64 の /4 刻みです。

インターネット上には、IPv6 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。「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。

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、EC2 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`: VPC ルーター用に AWS で予約されています。
- `2001:db8:1234:1a00::2`
- `2001:db8:1234:1a00::3`
- `2001:db8:1234:1a00:ffff:ffff:ffff:ffff`

上記の例の VPC ルーター用に AWS で予約された IP アドレスに加えて、次の IPv6 アドレスがデフォルト VPC ルーター用に予約されています。

- EUI-64 を使用して生成された FE80::/10 範囲のリンクローカル IPv6 アドレス。リンクローカルアドレスの詳細については、「Link-local address」(リンクローカルアドレス) を参照してください。
- リンクローカル IPv6 アドレス `FE80:ec2::1`。

IPv6 経由で VPC ルーターと通信する必要がある場合は、ニーズに最適なアドレスと通信するようにアプリケーションを設定できます。

IPv4 と IPv6 を比較する

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。

デュアルスタック設定 (IPv4 と IPv6) および IPv6 専用設定をサポートしている AWS サービスの一覧は、「IPv6 をサポートするサービス」を参照してください。

特徴	IPv4	IPv6
VPC のサイズ	/16 から /28 までの最大 5 個の CIDR。このクォータは調整可能です。	/44 から /60 の、/4 刻みで最大 5 個の CIDR。このクォータは調整可能です。
サブネットのサイズ	/16 から /28。	/44 から /64 の、/4 刻み。
アドレスの選択	VPC の IPv4 CIDR ブロックを選択するか、Amazon VPC IP Address Manager (IPAM) から CIDR ブロックを割り当てることができます。詳細については、「Amazon VPC IPAM ユーザーガイド」の「IPAM とは」を参照してください。	AWS で独自の IPv6 CIDR ブロックを VPC に使用するか、Amazon が提供する IPv6 CIDR ブロックを選択するか、Amazon VPC IP Address Manager (IPAM) から CIDR ブロックを割り当てることができます。詳細については、「Amazon VPC IPAM ユーザーガイド」の「IPAM とは」を参照してください。
インターネットアクセス	インターネットゲートウェイが必要です。	インターネットゲートウェイが必要です。エグレス専用のインターネットゲートウェイを使用する送信専用の通信をサポートします。
Elastic IP アドレス	サポート対象。EC2 インスタンスに、永続的な静的パブリック IPv4 アドレスを付与します。	サポート外。EIP は、インスタンスの再起動時に、インスタンスのパブリック IPv4 アドレスを維持します。IPv6 アドレスはデフォルトでは静的です。
NAT ゲートウェイ	サポート対象。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 ゲートウェイを使用してインターネットに接続するか、プライベート NAT ゲートウェイを使用して他の VPC のリソースに接続することができます。	サポート対象。NAT64 の NAT ゲートウェイを使用すると、IPv6 専用サブネットのインスタンスは、VPC 内、VPC 間、オンプレミスネットワーク内で、または、インターネット経由で、IPv4 専用リソースと通信することが可能になります。

特徴	IPv4	IPv6
DNS 名	インスタンスは、Amazon が提供する IPBN または RBN ベースの DNS 名を受け取ります。DNS 名は、インスタンスに対して選択された DNS レコードに解決されます。	インスタンスは、Amazon が提供する IPBN または RBN ベースの DNS 名を受け取ります。DNS 名は、インスタンスに対して選択された DNS レコードに解決されます。

マネージドプレフィックスリストを使用したネットワーク CIDR ブロックの統合と管理

マネージド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。Resource Access Manager (RAM) を使用して、他の AWS アカウントでマネージドプレフィックスリストを使用することもできます。

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の Identity and Access Management
- カスタマー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

- プレフィックスリストを使用して AWS インフラストラクチャ管理を最適化する

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、最大エントリ数が 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、「プレフィックスリストのルーティング優先度」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」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は、GovCloud (米国) リージョンと中国リージョンを含む、すべての商用 AWS リージョンで利用できます。

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。
- 異なる AWS マネージドプレフィックスリストを使用すると、ウェイトが異なります。詳細については、「AWS マネージドプレフィックスリストのウェイト」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、 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。IAM ポリシーを作成し、ユーザーにプレフィックスリストの操作を許可するロールにアタッチ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を参照するには、サービス認可リファレンスの「Amazon EC2 のアクション、リソース、および条件キー」を参照してください。

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。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:GetManagedPrefixListAssociations",
                "ec2:GetManagedPrefixListEntries",
                "ec2:ModifyManagedPrefixList",
                "ec2:RestoreManagedPrefixListVersion"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012: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」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

カスタマーマネージドプレフィックスリストを使用することで、プレフィックスと呼ばれる、独自の IP アドレス範囲のセットをAWS 内で定義し維持することができます。こうした IP アドレスをさまざまなリソースにハードコーディングするのではなく、一元化されたプレフィックスリストを作成して必要に応じて参照することができます。それにより、IP アドレスの管理を簡素化できるだけでなく、AWS ランドスケープ全体で一貫性を維持し再利用を促すことができます。

カスタマーマネージドプレフィックスリストの特筆すべき特徴の 1 つが他の AWS アカウントと共有できる点です。プレフィックスリストへのアクセスを許可することで、自分が定義した IP アドレス範囲を他のチームや組織が各自のリソース内で使用できるようになります。この協働的なアプローチによって、IP アドレスの管理が共有化され同期された、より一体感のある効率的なクラウド体験を促せます。

以下のセクションでは、IP アドレス範囲の作成、管理、共有に関する詳細な手順を含め、カスタマーマネージドプレフィックスリストを使用する際の実際的側面の数々を詳しく解説していきます。

> **Note**
>
> Amazon VPC IPAM を使用してプレフィックスリスト管理を自動化し、定義したルールに基づいて CIDR を自動的に同期できます。これにより、インフラストラクチャが変更されたときの手動更新が不要になります。詳細については、「Amazon VPC IPAM ユーザーガイド」の「IPAM でプレフィックスリストの更新を自動化する」を参照してください。

タスク

- カスタマーマネージドプレフィックスリストの操作

カスタマーマネージドプレフィックスリストの操作

このセクションでは、カスタマーマネージドプレフィックスリストの使用方法について説明します。

内容

- 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表示する
- 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の関連付け (参照) の表示
- プレフィックスリストの変更
- プレフィックスリストのサイズ変更
- プレフィックスリストの以前のバージョンを復元する
- プレフィックスリストを削除する
- カスタマーマネージド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。

制限

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。

制限

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。エントリの最大数を修正する場合は、プレフィックスリストのサイズ変更 を参照してください。

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。

考慮事項

- 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. マネージドプレフィックスリストが IPAM プレフィックスリストリゾルバーターゲットとして設定されている場合、[IPAM プレフィックスリストリゾルバーの同期] オプションが表示されます。

   IPAM プレフィックスリストリゾルバーとの同期を有効にするか無効にするかを選択します。有効にすると、プレフィックスリスト CIDR は、関連付けられたリゾルバーの CIDR 選択ルールに基づいて自動的に更新されます。無効にすると、プレフィックスリスト CIDR は自動的に更新されません。この機能の詳細については、「Amazon VPC IPAM ユーザーガイド」の「IPAM でプレフィックスリストの更新を自動化する」を参照してください。
6. [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。
7. [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サイズ変更

プレフィックスリストのサイズを変更し、プレフィックスリストの最大エントリ数を 1,000 個まで変更できます。カスタマーマネージドプレフィックスリストのクォータの詳細については、「カスタマーマネージドプレフィックスリスト」を参照してください。

コンソールを使用してプレフィックスリストのサイズを変更するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[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)、[Resize prefix list] (プレフィックスリストのサイズ変更) の順に選択します。

4. [New max entries] (新しい最大エントリ) に値を入力します。
5. [サイズ変更] を選択します。

AWS CLI を使用してプレフィックスリストのサイズを変更するには

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。

制限

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。

カスタマーマネージドプレフィックスリストを共有する

AWS Resource Access Manager (AWS RAM) により、カスタマーマネージド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と共有することができます。

- AWS Organizations の組織内または組織外の特定の AWS アカウント
- AWS Organizations の組織内の組織単位
- AWS Organizations の組織全体

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。

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

> **Important**
> 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。

内容

- 共有プレフィックスリストのアクセス許可
- 共有プレフィックスリストの操作

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。

共有プレフィックスリストの操作

AWS プレフィックスリストは、さまざまな AWS サービスで使用される IP アドレス範囲を管理し参照するための便利な方法です。AWS マネージドプレフィックスリストに加えて、ユーザーは独自のカスタマーマネージドプレフィックスリストを作成し、他の AWS アカウントと共有することもできます。

プレフィックスリストの共有は、複雑なネットワーク要件を持つ組織や、複数の AWS ワークロードで IP アドレスの使用を調整する必要がある組織に、特に有用です。プレフィックスリストを共有することで、一貫性のある IP アドレス管理を遂行し、共同作業者のネットワーク設定を簡素化することができます。

このセクションでは、プレフィックスリストを共有する方法と、アカウントに共有されたプレフィックスリストを識別して使用する方法について説明します。

内容

- プレフィックスリストを共有する
- 共有プレフィックスリストの共有解除
- 共有プレフィックスリストの特定
- 共有プレフィックスリストへの参照の特定

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。

Important

- プレフィックスリストを共有するには、リストを所有する必要があります。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。

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。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。

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は、AWS サービスの IP アドレス範囲一式です。これらのプレフィックスリストは Amazon Web Services が管理しており、さまざまな AWS サービスで使用される IP アドレスを参照する手段を提供します。これは、セキュリティグループまた VPC 内の他のネットワークレベルのコントロールを設定する際に特に有用です。

プレフィックスリストは、S3 や DynamoDB、その他数多くの AWS サービスに対応しています。マネージドプレフィックスリストを使用することで、ネットワーク構成が最新状態であり、依存する AWS サービスで使用される IP アドレスが適切に考慮されていることを確認できます。それにより、ネットワークタスクを簡素化し、IP アドレスのリストを手動で管理する際の管理オーバーヘッドを削減することができます。

マネージドプレフィックスリストを使用することは、実用面で利点があるだけでなく、AWS セキュリティのベストプラクティスにも一致します。AWS が提供する信頼のおける IP アドレス情報に依存することで、設定ミスや予期せぬ接続問題によるリスクを最小限に抑えることができます。こうしたことは、厳格なコンプライアンス要件を持つミッションクリティカルなアプリケーションやワークロードにとって特に重要です。

内容

- 使用可能な 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のウェイト
- AWS マネージドプレフィックスリストの使用

使用可能な AWS マネージドプレフィックスリスト

次のサービスが AWS マネージドプレフィックスリストを提供します。

AWS のサービス	プレフィックスリスト名	Weight
Amazon CloudFront +	com.amazonaws.global.cloudfront.origin-facing (IPv4) com.amazonaws.global.ipv6.cloudfront.origin-facing (IPv6)	55
Amazon DynamoDB	com.amazonaws.*region*.dynamodb	1
Amazon EC2 Instance Connect	com.amazonaws.*region*.ec2-instance-connect	2
	com.amazonaws.*region*.ipv6.ec2-instance-connect	2
AWS Ground Station	com.amazonaws.global.groundstation	5
Amazon Route 53:	com.amazonaws.*region*.ipv6.route53-healthchecks	25
	com.amazonaws.*region*.route53-healthchecks	25
Amazon S3	com.amazonaws.*region*.s3	1
Amazon S3 Express One Zone	com.amazonaws.*region*.s3express	6
AWS Secrets Manager	com.amazonaws.*region*.secretsmanager-managed-external-secrets	20

AWS のサービス	プレフィックスリスト名	Weight
Amazon VPC Lattice	com.amazonaws.*region*.vpc-lattice	10
	com.amazonaws.*region*.ipv6.vpc-lattice	10

コンソールを使用して AWS マネージドプレフィックスリストを表示するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[マネージドプレフィックスリスト] を選択します。
3. 検索フィールドに [Owner ID: AWS] フィルターを追加します。

AWS CLI を使用して AWS マネージドプレフィックスリストを表示するには

describe-managed-prefix-lists コマンドを以下のように使用します。

```
aws ec2 describe-managed-prefix-lists --filters Name=owner-id,Values=AWS
```

AWS マネージドプレフィックスリストのウェイト

AWS マネージドプレフィックスリストのウェイトは、このプレフィックスリストがリソースで占めるエントリ数を参照します。

例えば、Amazon CloudFront マネージドプレフィックスリストのウェイトは 55 です。これが Amazon VPC クォータにどのように影響するかは次のとおりです。

- セキュリティグループ – デフォルトのクォータには 60 個のルールがあり、セキュリティグループのルールには残り 5 個しか追加できません。このクォータに対して、クォータの増加をリクエストすることができます。
- ルートテーブル – デフォルトのクォータのルートは 50 個のため、プレフィックスリストをルートテーブルに追加する前に、クォータの引き上げをリクエストする必要があります。

AWS マネージドプレフィックスリストの使用

AWS マネージドプレフィックスリストは AWS が作成と管理を行い、AWS アカウントを所有していれば誰でも使用できます。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。

顧客管理のプレフィックスリストと同様に、AWS マネージドプレフィックスリストは、セキュリティグループやルートテーブルなどの AWS リソースで使用できます。詳細については、「プレフィックスリストを使用して AWS インフラストラクチャ管理を最適化する」を参照してください。

プレフィックスリストを使用して AWS インフラストラクチャ管理を最適化する

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
- サブネットルートテーブル
- トランジットゲートウェイルートテーブル
- AWS Network Firewall ルールグループ
- Amazon Managed Grafana のネットワークアクセスコントロール
- AWS Outposts ラックローカルゲートウェイ

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。詳細については、「セキュリティグループ」を参照してください。

> **Important**
>
> プレフィックスリストを使用するために既存のルールを変更することはできません。プレフィックスリストを使用するには新しいルールを作成する必要があります。

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 [カスタム] を選択します。その後、次のフィールドの [プレフィックスリスト] 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、「ルートテーブルを設定する」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。

AWS Network Firewall ルールグループ

AWS Network Firewall 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。Suricata 互換のステートフルルールグループを AWS Network Firewall に作成すると、ルールグループからプレフィックスリストを参照できます。詳細については、AWS Network Firewall 開発者ガイドの「Amazon VPC プレフィックスリストの参照」および「ステートフルルールグループの作成」を参照してください。

Amazon Managed Grafana のネットワークアクセスコントロール

Amazon Managed Grafana ワークスペースに対するリクエストのインバウンドルールとして、1 つまたは複数のプレフィックスリストを指定できます。プレフィックスリストの参照方法を含めた Grafana ワークスペースのネットワークアクセスコントロールに関する詳細については、「Amazon Managed Grafana ユーザーガイド」の「ネットワークアクセスの管理」を参照してください。

AWS Outposts ラックローカルゲートウェイ

各 AWS Outposts ラックには、Outpost リソースをオンプレミスネットワークに接続できるローカルゲートウェイが用意されています。頻繁に使用する CIDR をプレフィックスリストにグループ化し、このリストをローカルゲートウェイルートテーブルのルートターゲットとして参照できます。詳細については、「ラック用 AWS Outposts ユーザーガイド」の「ローカルゲートウェイのルートテーブルルートを管理する」を参照してください。

AWS IP アドレスの範囲

AWS は、その現在の IP アドレス範囲を JSON 形式で公開します。この情報をもとに、AWS からのトラフィックを特定することができます。また、この情報を使用して、一部の AWS のサービスとのトラフィックを許可または拒否することもできます。

考慮事項

- 当社は、お客様がエグレスフィルタリングを実行するために使用することが多いサービスの IP アドレス範囲を公開します。サービスには IP アドレス範囲が公開されていないものもあります。
- サービスは IP アドレス範囲を使用して他のサービスと通信または顧客ネットワークと通信します。
- 独自の IP アドレスの持ち込み (BYOIP) を通じて AWS に持ち込む IP アドレス範囲は、`.json` ファイルには含まれていません。詳細については、「Amazon EC2 ユーザーガイド」の「AWS を通じてアドレス範囲をアドバタイズする」を参照してください。

一部のサービスでは、AWS マネージドプレフィックスリストを使用してアドレス範囲が公開されます。詳細については、「the section called “使用可能な AWS マネージドプレフィックスリスト”」を参照してください。

内容

- JSON ファイルをダウンロードする
- 送信コントロール
- 位置情報フィード
- AWS のサービスの IP アドレス範囲を検索する
- AWS IP アドレス範囲 JSON の構文
- AWS の IP アドレス範囲の通知

JSON ファイルをダウンロードする

現在のアドレス範囲を表示するには、ip-ranges.json をダウンロードします。過去のバージョンを保持するには、歴代バージョンの JSON ファイルを自分のコンピュータに保存します。ファイルを最後に保存してから変更があるかどうかを確認するには、現在のファイルの公開時刻を確認し、それを最後に保存したファイルの公開時刻と比較します。

JSON ファイルを現在のディレクトリに保存する curl コマンドの例を次に示します。

```
curl -O https://ip-ranges.amazonaws.com/ip-ranges.json
```

プログラムを使用してこのファイルにアクセスする場合、サーバーが提供した TLS 証明書を正しく確認した後にアプリケーションがファイルをダウンロードすることを保証する必要があります。

JSON ファイルの更新に関する通知を受け取るには、「AWS の IP アドレス範囲の通知」を参照してください。

送信コントロール

ある AWS サービスで作成したリソースが他の AWS サービスにのみアクセスできるようにするには、ip-ranges.json ファイルの IP アドレス範囲情報を使用して送信フィルタリングを実行できます。セキュリティグループのルールが AMAZON リストの CIDR ブロックへのアウトバウンドトラフィックを許可していることを確認してください。セキュリティグループにはクォータがあります。各リージョンの IP アドレス範囲の数によっては、リージョンごとに複数のセキュリティグループが必要になる場合があります。

> **Note**
>
> AWS サービスの中には、EC2 上に構築され、EC2 IP アドレススペースを使用するものもあります。EC2 IP アドレススペースへのトラフィックをブロックすると、EC2 以外のサービスへのトラフィックもブロックされます。

位置情報フィード

`ip-ranges.json` の IP アドレス範囲は AWS リージョン によって決まります。ただし、ローカルゾーンは親リージョンと同じ物理的な場所にはありません。geo-ip-feed.csv で公開された位置情報データには、ローカルゾーンについてのデータが含まれます。データは RFC 8805 に従います。

AWS のサービスの IP アドレス範囲を検索する

AWS が提供する AWS IP アドレス範囲の JSON ファイルは、さまざまな AWS サービスの IP アドレスを検索し、その情報を使用してネットワークセキュリティとアクセスコントロールを強化するための貴重なリソースとして利用できます。この JSON ファイル内に含まれる詳細なデータを解析することで、特定の AWS のサービス サービスやリージョンに関連付けられた IP アドレス範囲を正確に特定することができます。

例えば、IP アドレス範囲を使用して堅牢なネットワークセキュリティポリシーを設定し、特定の AWS リソースへのアクセスを許可または拒否する詳細なファイアウォールルールを設定することができます。この情報はさまざまな AWS Network Firewall タスクにも役立ちます。このレベルの制御は、アプリケーションとデータを保護するのに不可欠で、承認されたトラフィックのみが必要な AWS のサービスに到達できるようになります。さらに、この IP インテリジェンスを使用することは、適当な AWS エンドポイントと通信するようにアプリケーションを適切に設定し、全体的な信頼性とパフォーマンスを高めるのに役立ちます。

`ip-ranges.json` ファイルは、単なるファイアウォールルールを超えて、ネットワークインフラストラクチャに高度な送信フィルタリングを設定するのにも使用できます。さまざまな AWS のサービスの送信先 IP アドレス範囲を理解することで、ルーティングポリシーを設定したり、高度なネットワークセキュリティソリューションを使用したりして、意図した送信先に基づいてアウトバウンドトラフィックを選択的に許可または拒否するといったことができます。このエグレスコントロールは、データ漏洩や不正アクセスのリスクを軽減するために不可欠です。

重要なことは、`ip-ranges.json` ファイルは定期的に更新されるため、常に最も正確な最新の情報を入手するのに、最新のローカルコピーを維持することが不可欠だということです。ファイルの内

容を継続的に使用することで、AWS ベースのアプリケーションのネットワークアクセスとセキュリティを効率的に管理し、クラウドのセキュリティ体制全般を強化することができます。

次の例は、AWS IP アドレス範囲を目的のものだけにフィルタリングするのに役立ちます。Linux では、jq ツールをダウンロードおよび使用して、JSON ファイルのローカルコピーを解析できます。AWS Tools for Windows PowerShell には、この JSON ファイルの解析に使用できる コマンドレット Get-AWSPublicIpAddressRange が含まれています。詳細については、ブログ「AWS のパブリック IP アドレス範囲のクエリの実行」を参照してください。

JSON ファイルを取得するには、「the section called “ダウンロード”」を参照してください。JSON ファイルの構文の詳細については、「the section called “構文”」を参照してください。

例

- ファイル作成日を取得する
- 特定のリージョンの IP アドレスを取得する
- すべての IPv4 アドレスを取得します
- 特定のサービスのすべての IPv4 アドレスを取得します
- 特定のリージョンで、サービスのすべての IPv4 アドレスを取得します
- すべての IPv6 アドレスを取得します
- 特定のサービスのすべての IPv6 アドレスを取得する
- 特定のボーダーグループのすべての IP アドレスを取得する

ファイル作成日を取得する

次の例では、`ip-ranges.json` の作成日を取得しています。

jq

```
$ jq .createDate < ip-ranges.json

"2024-08-01-17-22-15"
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange -OutputPublicationDate
```

```
Thursday, August 1, 2024 9:22:35 PM
```

特定のリージョンの IP アドレスを取得する

次の例では、指定したリージョンの IP アドレスについて JSON ファイルをフィルタリングしています。

jq

```
$ jq '.prefixes[] | select(.region=="us-east-1")' < ip-ranges.json

{
  "ip_prefix": "23.20.0.0/14",
  "region": "us-east-1",
  "network_border_group": "us-east-1",
  "service": "AMAZON"
},
{
  "ip_prefix": "50.16.0.0/15",
  "region": "us-east-1",
  "network_border_group": "us-east-1",
  "service": "AMAZON"
},
{
  "ip_prefix": "50.19.0.0/16",
  "region": "us-east-1",
  "network_border_group": "us-east-1",
  "service": "AMAZON"
},
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange -Region us-east-1

IpPrefix        Region      NetworkBorderGroup         Service
--------        ------       -------                   -------
23.20.0.0/14    us-east-1    us-east-1                 AMAZON
50.16.0.0/15    us-east-1    us-east-1                 AMAZON
50.19.0.0/16    us-east-1    us-east-1                 AMAZON
...
```

すべての IPv4 アドレスを取得します

次の例では、IPv4 アドレスについて JSON ファイルをフィルタリングしています。

jq

```
$ jq -r '.prefixes | .[].ip_prefix' < ip-ranges.json

23.20.0.0/14
27.0.0.0/22
43.250.192.0/24
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange | where {$_.IpAddressFormat -eq "Ipv4"} | select
 IpPrefix

IpPrefix
--------
23.20.0.0/14
27.0.0.0/22
43.250.192.0/24
...
```

特定のサービスのすべての IPv4 アドレスを取得します

次の例では、指定したサービスの IPv4 アドレスについて JSON ファイルをフィルタリングしています。

jq

```
$ jq -r '.prefixes[] | select(.service=="GLOBALACCELERATOR") | .ip_prefix' < ip-
ranges.json

13.248.117.0/24
15.197.34.0/23
15.197.36.0/22
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange -ServiceKey GLOBALACCELERATOR | where
 {$_.IpAddressFormat -eq "Ipv4"} | select IpPrefix

IpPrefix
--------
13.248.117.0/24
15.197.34.0/23
15.197.36.0/22
...
```

特定のリージョンで、サービスのすべての IPv4 アドレスを取得します

次の例では、指定したリージョン内の指定したサービスの IPv4 アドレスについて JSON ファイルをフィルタリングしています。

jq

```
$ jq -r '.prefixes[] | select(.region=="us-east-1") |
 select(.service=="GLOBALACCELERATOR") | .ip_prefix' < ip-ranges.json

13.248.124.0/24
99.82.166.0/24
99.82.171.0/24
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange -Region us-east-1 -ServiceKey GLOBALACCELERATOR
 | where {$_.IpAddressFormat -eq "Ipv4"} | select IpPrefix

IpPrefix
--------
13.248.117.0/24
99.82.166.0/24
99.82.171.0/24
...
```

すべての IPv6 アドレスを取得します

次の例では、IPv6 アドレスについて JSON ファイルをフィルタリングしています。

jq

```
$ jq -r '.ipv6_prefixes | .[].ipv6_prefix' < ip-ranges.json

2a05:d07c:2000::/40
2a05:d000:8000::/40
2406:dafe:2000::/40
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange | where {$_.IpAddressFormat -eq "Ipv6"} | select
 IpPrefix

IpPrefix
--------
2a05:d07c:2000::/40
2a05:d000:8000::/40
2406:dafe:2000::/40
...
```

特定のサービスのすべての IPv6 アドレスを取得する

次の例では、指定したサービスの IPv6 アドレスについて JSON ファイルをフィルタリングしています。

jq

```
$ jq -r '.ipv6_prefixes[] | select(.service=="GLOBALACCELERATOR") | .ipv6_prefix' <
 ip-ranges.json

2600:1f01:4874::/47
2600:1f01:4802::/47
2600:1f01:4860::/47
2600:9000:a800::/40
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange -ServiceKey GLOBALACCELERATOR | where
 {$_.IpAddressFormat -eq "Ipv6"} | select IpPrefix

IpPrefix
--------
2600:1f01:4874::/47
2600:1f01:4802::/47
2600:1f01:4860::/47
2600:9000:a800::/40
...
```

特定のボーダーグループのすべての IP アドレスを取得する

次の例では、指定したボーダーグループのすべての IP アドレスについて JSON ファイルをフィルタリングしています。

jq

```
$ jq -r '.prefixes[] | select(.network_border_group=="us-west-2-lax-1")
 | .ip_prefix' < ip-ranges.json
70.224.192.0/18
52.95.230.0/24
15.253.0.0/16
...
```

PowerShell

```
PS C:\> Get-AWSPublicIpAddressRange | where {$_.NetworkBorderGroup -eq "us-west-2-
lax-1"} | select IpPrefix

IpPrefix
--------
70.224.192.0/18
52.95.230.0/24
15.253.0.0/16
...
```

AWS IP アドレス範囲 JSON の構文

AWS は、その現在の IP アドレス範囲を JSON 形式で公開します。JSON ファイルを取得するには、「the section called “ダウンロード”」を参照してください。JSON ファイルの構文は以下のとおりです。

```
{
  "syncToken": "0123456789",
  "createDate": "yyyy-mm-dd-hh-mm-ss",
  "prefixes": [
    {
      "ip_prefix": "cidr",
      "region": "region",
      "network_border_group": "network_border_group",
      "service": "subset"
    }
  ],
  "ipv6_prefixes": [
    {
      "ipv6_prefix": "cidr",
      "region": "region",
      "network_border_group": "network_border_group",
      "service": "subset"
    }
  ]
}
```

syncToken

UNIX エポック時刻形式での公開時刻。

型: 文字列

例: `"syncToken": "1416435608"`

createDate

発行日時 (UTC YY-MM-DD-hh-mm-ss 形式)。

型: 文字列

例: `"createDate": "2014-11-19-23-29-02"`

prefixes

IPv4 アドレス範囲の IP プレフィックス。

型: 配列

ipv6_prefixes

IPv6 アドレス範囲の IP プレフィックス。

型: 配列

ip_prefix

CIDR 表記でのパブリック IPv4 アドレス範囲。AWS はさらに特定の範囲でプレフィックスをアドバタイズする場合があります。たとえば、プレフィックス 96.127.0.0/17 が含まれるファイルでは、96.127.0.0/21、96.127.8.0/21、96.127.32.0/19、および 96.127.64.0/18 としてアドバタイズされる場合があります。

型: 文字列

例: `"ip_prefix": "198.51.100.2/24"`

ipv6_prefix

CIDR 表記でのパブリック IPv6 アドレス範囲。AWS はさらに特定の範囲でプレフィックスをアドバタイズする場合があります。

型: 文字列

例: `"ipv6_prefix": "2001:db8:1234::/64"`

network_border_group

AWS が IP アドレス、または GLOBAL をアドバタイズするアベイラビリティーゾーンまたはローカルゾーンの一意のセットである、ネットワーク境界グループの名前です。GLOBAL サービスのトラフィックは、AWS が IP アドレスをアドバタイズする複数の (最大ですべての) アベイラビリティーゾーンまたはローカルゾーンに引き付けられたり、そこから発信されたりする可能性があります。

タイプ: 文字列

例: `"network_border_group": "us-west-2-lax-1"`

リージョン

AWS リージョンまたは GLOBAL です。GLOBAL サービスのトラフィックは、複数の (最大ですべての) AWS リージョンに引き付けられたり、そこから発信されたりする可能性があります。

タイプ: 文字列

有効な値: af-south-1 | ap-east-1 | ap-east-2 | ap-northeast-1 | ap-northeast-2 | ap-northeast-3 | ap-south-1 | ap-south-2 | ap-southeast-1 | ap-southeast-2 | ap-southeast-3 | ap-southeast-4 | ap-southeast-5 | ap-southeast-6 | ap-southeast-7 | ca-central-1 | ca-west-1 | cn-north-1 | cn-northwest-1 | eu-central-1 | eu-central-2 | eu-north-1 | eu-south-1 | eu-south-2 | eu-west-1 | eu-west-2 | eu-west-3 | il-central-1 | mx-central-1 | me-central-1 | me-south-1 | sa-east-1 | us-east-1 | us-east-2 | us-gov-east-1 | us-gov-west-1 | us-west-1 | us-west-2 | GLOBAL

例: "region": "us-east-1"

サービス

IP アドレス範囲のサブセット。API_GATEWAY にリストされているアドレスは送信専用です。すべての IP アドレス範囲を取得する場合は、AMAZON を指定します (つまり、すべてのサブセットも AMAZON サブセットに含まれます)。ただし、一部の IP アドレス範囲は AMAZON サブセット内にしかありません (つまり、別のサブセットでは使用できません)。

タイプ: 文字列

有効な値: AMAZON | AMAZON_APPFLOW | AMAZON_CONNECT | API_GATEWAY | AURORA_DSQL | CHIME_MEETINGS | CHIME_VOICECONNECTOR | CLOUD9 | CLOUDFRONT | CLOUDFRONT_ORIGIN_FACING | CODEBUILD | DYNAMODB | EBS | EC2 | EC2_INSTANCE_CONNECT | GLOBALACCELERATOR | IVS_LOW_LATENCY | IVS_REALTIME | KINESIS_VIDEO_STREAMS | MEDIA_PACKAGE_V2 | ROUTE53 | ROUTE53_HEALTHCHECKS | ROUTE53_HEALTHCHECKS_PUBLISHING | ROUTE53_RESOLVER | S3 | WORKSPACES_GATEWAYS

例:"service": "AMAZON"

範囲の重複

任意のサービスコードから返される IP アドレスの範囲は、AMAZON サービスコードからも返されます。例えば、S3 サービスコードから返される IP アドレスのすべての範囲は、AMAZON サービスコードからも返されます。

サービス A がサービス B のリソースを使用する場合、サービス A とサービス B の両方のサービスコードから返される IP アドレス範囲があります。ただし、これらの IP アドレス範囲はサービス A によってのみ使用され、サービス B では使用できません。例えば、Amazon S3 は Amazon EC2 のリソースを使用するため、S3 と EC2 両方のサービスコードから返される IP アドレス範囲があります。ただし、これらの IP アドレス範囲は Amazon S3 でのみ使用されます。したがって、S3 サービスコードは、Amazon S3 でのみ使用されるすべての IP アドレス範囲を返します。Amazon EC2 でのみ使用される IP アドレス範囲を特定するには、S3 サービスコードではなく、EC2 サービスコードから返される IP アドレス範囲を探してください。

詳細情報

このセクションには、さまざまなサービスコードの追加情報へのリンクを掲載しています。

- AMAZON_APPFLOW – IP アドレス範囲
- AMAZON_CONNECT – ネットワークセットアップ
- CHIME_MEETINGS —メディアとシグナリング向けの設定
- CLOUDFRONT – CloudFront エッジサーバーの場所と IP アドレス範囲
- DYNAMODB – IP アドレス範囲
- EC2 – パブリック IPv4 アドレス
- EC2_INSTANCE_CONNECT —EC2 Instance Connect の前提条件
- GLOBALACCELERATOR – Global Accelerator エッジサーバーの場所と IP アドレス範囲
- ROUTE53 – Amazon Route 53 サーバーの IP アドレス範囲
- ROUTE53_HEALTHCHECKS – Amazon Route 53 サーバーの IP アドレス範囲
- ROUTE53_HEALTHCHECKS_PUBLISHING – Amazon Route 53 サーバーの IP アドレス範囲
- WORKSPACES_GATEWAYS – PCoIP ゲートウェイサーバー

リリースノート

次の表では ip-ranges.json の構文に対する更新について説明します。リージョンの開始ごとに、新しいリージョンコードが追加されています。

説明	リリース日
IVS_LOW_LATENCY　サービスコードが追加されました。	2025 年 7 月 29 日

説明	リリース日
AURORA_DSQL サービスコードが追加されました。	2025 年 5 月 21 日
IVS_REALTIME サービスコードが追加されました。	2024 年 6 月 11 日
MEDIA_PACKAGE_V2 サービスコードが追加されました。	2023 年 5 月 9 日
CLOUDFRONT_ORIGIN_FACING サービスコードが追加されました。	2021 年 10 月 12 日
ROUTE53_RESOLVER サービスコードが追加されました。	2021 年 6 月 24 日
EBS サービスコードが追加されました。	2021 年 5 月 12 日
KINESIS_VIDEO_STREAMS サービスコードが追加されました。	2020 年 11 月 19 日
サービスコードとして CHIME_MEETINGS と CHIME_VOICECONNECTOR を追加しました。	2020 年 6 月 19 日
AMAZON_APPFLOW サービスコードが追加されました。	2020 年 6 月 9 日
ネットワーク境界グループのサポートを追加します。	2020 年 4 月 7 日
WORKSPACES_GATEWAYS サービスコードが追加されました。	2020 年 3 月 30 日
ROUTE53_HEALTHCHECK_PUBLISHING サービスコードが追加されました。	2020 年 1 月 30 日
API_GATEWAY サービスコードが追加されました。	2019 年 9 月 26 日

説明	リリース日
EC2_INSTANCE_CONNECT　サービスコードが追加されました。	2019 年 6 月 26 日
DYNAMODB サービスコードが追加されました。	2019 年 4 月 25 日
GLOBALACCELERATOR　サービスコードが追加されました。	2018 年 12 月 20 日
AMAZON_CONNECT　サービスコードが追加されました。	2018 年 20 月 6 日
CLOUD9 サービスコードが追加されました。	2018 年 20 月 6 日
CODEBUILD　サービスコードが追加されました。	2018 年 4 月 19 日
S3 サービスコードが追加されました。	2017 年 2 月 28 日
IPv6 アドレス範囲のサポートが追加されました。	2016 年 8 月 22 日
初回リリース	2014 年 11 月 19 日

AWS の IP アドレス範囲の通知

AWS は、その現在の IP アドレス範囲を JSON 形式で公開します。AWS の IP アドレス範囲が変更されるときは、常に AmazonIpSpaceChanged という名前の Amazon SNS トピックのサブスクライバーに通知が送信されます。JSON ファイルの構文の詳細については、「the section called “構文”」を参照してください。

通知のペイロードには、次の形式の情報が含まれています。

```
{
  "create-time":"yyyy-mm-ddThh:mm:ss+00:00",
  "synctoken":"0123456789",
  "md5":"6a45316e8bc9463c9e926d5d37836d33",
  "url":"https://ip-ranges.amazonaws.com/ip-ranges.json"
```

```
}
```

create-time

作成日時。

通知は、誤った順序で配信される場合があります。したがって、正しい順序を保証するためにタイムスタンプを確認することをお勧めします。

synctoken

UNIX エポック時刻形式での公開時刻。

md5

`ip-ranges.json` ファイルの暗号ハッシュ値。この値を使用して、ダウンロードしたファイルが破損しているかどうかを確認できます。

url

`ip-ranges.json` ファイルの場所。詳細については、「the section called “ダウンロード”」を参照してください。

通知の受け取りは、次のようにサブスクライブすることができます。

AWS の IP アドレス範囲の通知をサブスクライブするには

1. Amazon SNS コンソール (https://console.aws.amazon.com/sns/v3/home) を開きます。
2. ナビゲーションバーで、必要に応じて、リージョンを [米国東部 (バージニア北部)] に変更します。サブスクライブする SNS 通知がこのリージョンで作成されているため、このリージョンを選択する必要があります。
3. ナビゲーションペインで [Subscriptions] を選択してください。
4. [Create subscription] を選択してください。
5. [Create subscription] ダイアログボックスで、次の操作を行います。
   a. トピックの ARN には、以下の Amazon リソースネーム (ARN) をコピーします。

      ```
      arn:aws:sns:us-east-1:806199016981:AmazonIpSpaceChanged
      ```

   b. プロトコルには、使用するプロトコルを選択します (たとえば、`Email`)。

c. エンドポイントには、通知を受け取るエンドポイントを入力します (たとえば、E メールアドレス)。

d. [Create subscription] を選択します。

6. 指定したエンドポイントに接続されて、登録を確認するように求められます。たとえば、E メールアドレスを指定した場合は、件名に `AWS Notification - Subscription Confirmation` と表示された E メールメッセージが届きます。指示に沿って操作し、登録を確認します。

通知はエンドポイントの可用性によって異なります。そのため、JSON ファイルを定期的に確認して、常に最新の範囲を入手した方がよいでしょう。Amazon SNS の信頼性について詳しくは、 を参照してくださいhttps://aws.amazon.com/sns/faqs/#Reliability

通知が不要になった場合は、次の手順で受信登録を解除します。

AWS の IP アドレス範囲の通知へのサブスクリプションを解除するには

1. Amazon SNS コンソール (https://console.aws.amazon.com/sns/v3/home) を開きます。
2. ナビゲーションペインで [Subscriptions] を選択します。
3. サブスクリプションのチェックボックスをオンにします。
4. [Actions]、[Delete subscriptions] の順に選択します。
5. 確認を求めるメッセージが表示されたら、[削除] を選択します。

詳細については、Amazon SNS、Amazon Simple Notification Service Developer Guide を参照してください。

VPC の IPv6 サポート

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、その VPC とリソースに IPv6 を追加できます。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。

考慮事項

- 現在、IPv4 専用サブネットから IPv6 専用サブネットへの移行パスはありません。
- この例は、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 VPC の作成方法については、「the section called “VPC を作成する”」を参照してください。
- IPv6 の使用を開始する前に、Amazon VPC に対する IPv6 アドレス指定の機能に関する「IPv4 と IPv6 を比較する」を参照したことを確認します。

内容

- VPC の IPv6 サポートを追加する
- デュアルスタック VPC 設定の例

VPC の IPv6 サポートを追加する

次の表は、VPC で IPv6 を有効にするためのプロセスの概要を示しています。

内容

- 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
- ステップ 2: ルートテーブルを更新する
- ステップ 3: セキュリティグループルールを更新する
- ステップ 4: IPv6 アドレスをインスタンスに割り当てる

Step	注意事項
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	Amazon が提供する IPv6 CIDR ブロック、または BYOIP の 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	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

Step	注意事項
	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	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: IPv6 アドレスをインスタンスに割り当てる	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 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 を選択します。
4. [アクション]、[CIDR の編集]、[新しい IPv6 CIDR の追加] の順に選択します。
5. 次のいずれかのオプションを選択し、[CIDR の選択] を選択します。

   - [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 ブロックを使用します。[ネットワークボーダーグループ] 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。
   - [IPAM 割り当て済み IPv6 CIDR ブロック] – IPAM プールから IPv6 CIDR ブロックを使用します。IPAM プールと IPv6 CIDR ブロックを選択します。

- [ユーザー所有の IPv6 CIDR] – (BYOIP) IPv6 アドレスプールから IPv6 CIDR ブロックを使用します。IPv6 アドレスプールおよび IPv6 CIDR ブロックを選択します。

6. [閉じる] を選択してください。

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、[サブネット] を選択してください。
3. サブネットを選択します。
4. [アクション]、[IPv6 CIDR の編集]、[IPv6 CIDR の追加] の順に選択します。
5. 必要に応じて CIDR ブロックを編集します (例えば、`00` を置き換えます)。
6. [保存] を選択します。
7. VPC 内の他のすべて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、「IPv6 VPC CIDR ブロック」を参照してください。

ステップ 2: ルートテーブルを更新する

IPv6 CIDR ブロックを VPC と関連付けると、VPC の各ルートテーブルにローカルルートが自動的に追加され、VPC 内で IPv6 トラフィックが有効になります。

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、NAT ゲートウェイは IPv6 をサポートしていないため、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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[サブネット] を選択してください。パブリックサブネットを選択します。[ルートテーブル] タブでルートテーブル ID を選択し、ルートテーブルの詳細ページを開きます。
3. ルートテーブルを選択します。[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。[宛先] で `::/0` を選択します。[ターゲット] でインターネットゲートウェイの ID を選択します。

5. [Save changes] (変更の保存) をクリックします。

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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[Egress Only インターネットゲートウェイ] を選択します。[Egress Only インターネットゲートウェイの作成] を選択します。VPC から [VPC] を選択し、[Egress Only インターネットゲートウェイの作成] を選択します。

   詳細については、「Egress-Only インターネットゲートウェイを使用してアウトバウンド IPv6 トラフィックを有効にする」を参照してください。
3. ナビゲーションペインで、[サブネット] を選択してください。プライベートサブネットを選択します。[ルートテーブル] タブでルートテーブル ID を選択し、ルートテーブルの詳細ページを開きます。
4. ルートテーブルを選択します。[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。[宛先] で `::/0` を選択します。[ターゲット] で Egress Only インターネットゲートウェイの ID を選択します。
6. [Save changes] (変更の保存) をクリックします。

> **Note**
>
> インターネットゲートウェイと Egress-Only インターネットゲートウェイの両方を同時にポイントする同じ送信先 (::/0) をルートテーブルに設定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を設定しているときに「ネクストホップがインターネットゲートウェイになっている既存の IPv6 ルートがあります」というエラーメッセージが表示された場合は、Egress-Only インターネットゲートウェイへのルートを追加する前に、インターネットゲートウェイへの既存の IPv6 ルートを削除する必要があります。

詳細については、「ルーティングオプションの例」を参照してください。

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 通信が含まれています。

インバウンドセキュリティグループルールを更新するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [セキュリティグループ] 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. [インバウンドルール] タブで、[インバウンドルールの編集] を選択します。
4. IPv4 トラフィックを許可するルールごとに、[ルールの追加] を選択し、対応する IPv6 トラフィックを許可するようにルールを設定します。例えば、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、[ソース] で [::/0] を選択します。
5. ルールの追加が完了したら、[ルールの保存] を選択します。

アウトバウンドセキュリティグループルールを更新する

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 VPC 用に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。

ネットワーク ACL ルールを更新する

IPv6 CIDR ブロックを VPC と関連付けると、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、デフォルトのネットワーク ACL を変更した場合、または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、「ルールの追加と削除」を参照してください。

ステップ 4: IPv6 アドレスをインスタンスに割り当てる

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。インスタンスタイプが IPv6 をサポートしていない場合は、IPv6 アドレスを割り当てる前に、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。使用するプロセスは、選択した新しいインスタンスタイプが現在のインスタンスタイプと互換性があるかどうかによって異なりま

す。詳細については、「Amazon EC2 ユーザーガイド」の「インスタンスタイプを変更する」を参照してください。IPv6 をサポートする新しい AMI からインスタンスを起動する必要がある場合は、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。

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 (例えば、eth0) に割り当てられます。詳しくは、「Amazon EC2 ユーザーガイド」の「インスタンスへの IPv6 アドレスの割り当て」を参照してください。

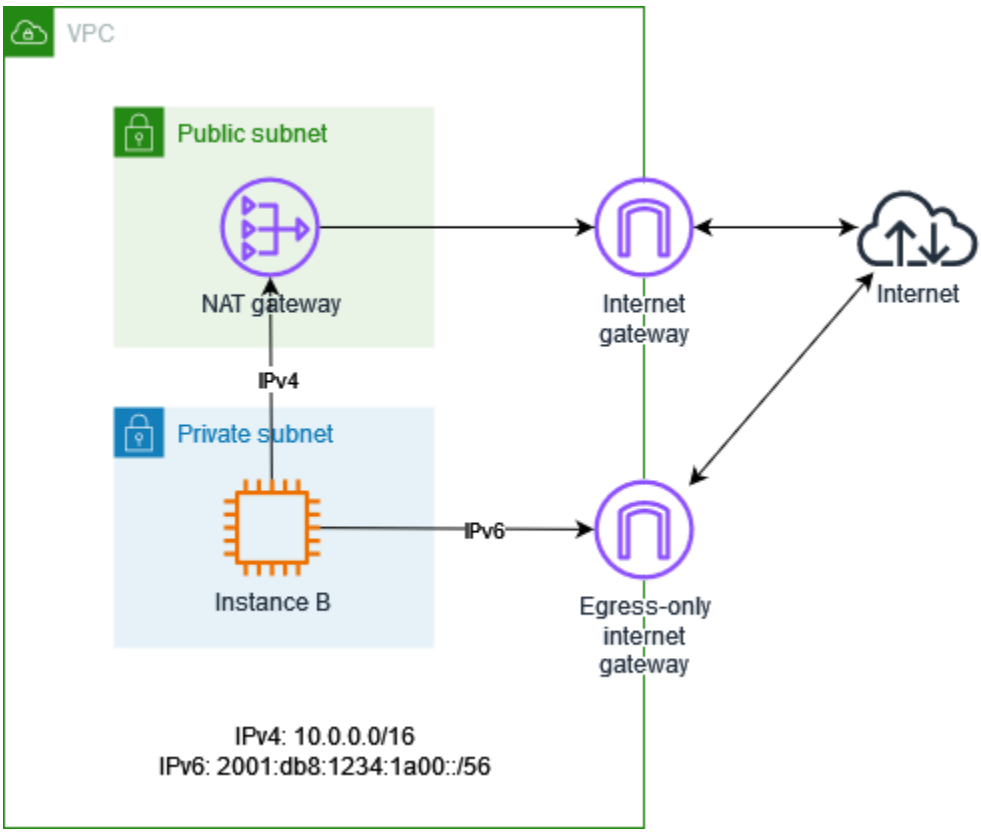
その IPv6 アドレスを使用してインスタンスに接続できます。詳細については、Amazon EC2 ユーザーガイドの「SSH クライアントを使用した Linux インスタンスへの接続」を参照してください。

現在のバージョンのオペレーティングシステム用の AMI 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 用に設定されます。インスタンスから IPv6 アドレスに ping を送信できない場合は、オペレーティングシステムのドキュメントを参照して IPv6 を設定してください。

デュアルスタック VPC 設定の例

デュアルスタック設定では、VPC 内のリソースとインターネット上のリソース間の通信に IPv4 アドレスと IPv6 アドレスの両方を使用できます。

次の図は、VPC のアーキテクチャを示しています。VPC には、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。VPC とサブネットには、IPv4 CIDR ブロックと IPv6 CIDR ブロックの両方があります。プライベートサブネットには、IPv4 アドレスと IPv6 アドレスの両方を持つ EC2 インスタンスがあります。インスタンスは、NAT ゲートウェイを使用してアウトバウンド IPv4 トラフィックをインターネットに送信し、Egress Only インターネットゲートウェイを使用してアウトバウンド IPv6 トラフィックをインターネットに送信できます。



パブリックサブネットのルートテーブル

以下は、パブリックサブネットのルートテーブルです。最初の 2 つのエントリはローカルルートです。3 番目のエントリは、すべての IPv4 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。4 番目のエントリは、パブリックサブネットで IPv6 アドレスを持つ EC2 インスタンスを起動する場合にのみ必要であることに注意してください。

ルーティング先	Target
VPC IPv4 CIDR	ローカル
VPC IPv6 CIDR	ローカル
0.0.0.0/0	*internet-gateway-id*
::/0	*internet-gateway-id*

プライベートサブネットのルートテーブル

以下は、プライベートサブネットのルートテーブルです。最初の 2 つのエントリはローカルルートです。3 番目のエントリは、すべての IPv4 トラフィックを NAT ゲートウェイに送信します。最後のエントリは、すべての IPv6 トラフィックを Egress Only インターネットゲートウェイに送信します。

ルーティング先	Target
VPC IPv4 CIDR	ローカル
VPC IPv6 CIDR	ローカル
0.0.0.0/0	*nat-gateway-id*
::/0	*egress-only-gateway-id*

IPv6 をサポートする AWS サービス

コンピューターとスマートデバイスは、IP アドレスを使用して、インターネットやその他のネットワークを介して相互に通信します。インターネットが拡大し続けるにつれて、IP アドレスの必要性も高まります。IP アドレスの最も一般的な形式は IPv4 です。IPv6 は IP アドレスの新しい形式で、IPv4 よりも大きなアドレス空間を提供します。

IPv6 の AWS のサービス サポートには、デュアルスタック (IPv4 と IPv6) または IPv6 のみの設定のサポートが含まれます。仮想プライベートクラウド (VPC) は AWS クラウド の 論理的に分離されたセクションであり、ユーザーが AWS リソースを起動できる場所です。VPC 内では、IPv4 のみ、デュアルスタック、または IPv6 のみのサブネットを作成できます。

AWS のサービス は、パブリックエンドポイントを介したアクセスもサポートします。一部の AWS のサービス は、AWS PrivateLink を利用したプライベートエンドポイントを使用したアクセスもサポートしています。AWS のサービス は、パブリックエンドポイントを通じて IPv6 をサポートしていなくても、プライベートエンドポイントを通じて IPv6 をサポートできます。IPv6 をサポートするエンドポイントは、AAAA レコードを使用して DNS クエリに応答できます。

IPv6 をサポートするサービス

次のテーブルは、デュアルスタックのサポート、IPv6 のみのサポート、および IPv6 をサポートする AWS のサービス を一覧しています。この表は、IPv6 の追加サポートがリリースされたときに更新

されます。サービスが IPv6 をサポートする方法についての情報は、そのサービス用のドキュメントを参照してください。

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
AWS Amplify	あり	いいえ	あり	
Amazon API Gateway	あり	いいえ	はい	はい
AWS App Mesh	はい	はい	はい	いいえ
AWS App Runner	はい	いいえ	はい	はい
AWS AppConfig	はい	いいえ	はい	はい
Application Auto Scaling	いいえ	なし	はい	はい
AWS Application Discovery Service	はい	いいえ	はい	あり
Application Recovery Controller (ARC)	あり	いいえ	はい	
Amazon WorkSpaces Applications	はい	いいえ	いいえ	なし
AWS AppSync[2]	部分的	なし	部分的	はい
Amazon Athena	あり	いいえ	はい	はい
AWS Audit Manager	いいえ	なし	はい	はい
Amazon Aurora	はい	いいえ	はい	いいえ
Amazon Aurora DSQL	いいえ	なし	はい	はい
AWS Auto Scaling	いいえ	なし	はい	はい

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
AWS B2B Data Interchange	はい	いいえ	はい	はい
AWS Backup	はい	いいえ	はい	はい
AWS Batch	はい	いいえ	はい	はい
Amazon Bedrock	いいえ	なし	はい	あり
AWS Billing and Cost Management Data Exports	あり	いいえ	はい	あり
AWS Billing and Cost Management Pricing Calculator	あり	いいえ	はい	はい
AWS Billing Conductor	はい	いいえ	はい	はい
AWS Budgets	はい	いいえ	はい	
Amazon Braket	あり	はい	はい	はい
AWS Certificate Manager	はい	いいえ	はい	いいえ
Amazon Chime SDK	はい	いいえ	はい	
Amazon Comprehend	あり	はい	はい	はい
AWS Clean Rooms	はい	はい	はい	あり
AWS Clean Rooms ML	あり	はい	はい	はい
AWS Cloud9	はい	いいえ	はい	

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
AWS クラウドコントロール API	はい	いいえ	はい	はい
CloudFormation	いいえ	なし	はい	あり
Amazon CloudFront	はい	はい	はい	
AWS CloudHSM	はい	いいえ	はい	はい
AWS CloudTrail	はい	いいえ	はい	あり
Amazon CloudWatch	あり	はい	はい	はい
Amazon CloudWatch Application Insights	いいえ	なし	はい	はい
Amazon CloudWatch Internet Monitor	いいえ	なし	はい	あり
Amazon CloudWatch Logs	あり	はい	はい	あり
Amazon CloudWatch Observability Access Manager	あり	はい	あり	
Amazon CloudWatch Synthetics	あり	いいえ	はい	はい
AWS Cloud Map	はい	はい	はい	あり
AWS クラウド WAN	あり	いいえ	はい	いいえ
AWS CodeArtifact	はい	いいえ	はい	はい

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
Connect Customer Customer Profiles	はい	いいえ	はい	はい
AWS CodeBuild	いいえ	なし	はい	はい
AWS CodeCommit	いいえ	なし	はい	はい
AWS CodeDeploy	いいえ	なし	はい	はい
AWS Compute Optimizer	いいえ	なし	はい	はい
Amazon Comprehend Medical	いいえ	なし	はい	あり
Amazon CodeGuru Profiler	あり	いいえ	はい	はい
Amazon Cognito	あり	いいえ	はい	
AWS Config	いいえ	なし	はい	はい
AWS Control Tower	いいえ	なし	はい	はい
AWS Cost Explorer	はい	いいえ	はい	はい
AWS Cost Optimization Hub	はい	いいえ	はい	はい
AWS Data Exchange	いいえ	なし	はい	はい
Amazon Data Firehose	なし	なし	はい	あり
Amazon Data Lifecycle Manager	あり	いいえ	はい	はい

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
AWS Database Migration Service	はい	いいえ	なし	はい
AWS DataSync	はい	はい	はい	はい
Amazon DataZone	いいえ	なし	はい	はい
AWS Deadline Cloud	はい	いいえ	はい	はい
Amazon Detective	あり	はい	はい	
Direct Connect	はい	はい	いいえ	はい
Directory Service	いいえ	なし	はい	はい
Amazon EBS ダイレクト API	あり	いいえ	はい	あり
Amazon EC2	はい	はい	はい	いいえ
EC2 イメージビルダー	あり	はい	はい	はい
Amazon ECR	あり	いいえ	はい	なし
Amazon ECS	はい	はい	はい	あり
Amazon EFS	はい	はい	はい	あり
Amazon EKS	部分的	部分的	あり	はい
Amazon EMR	いいえ	なし	はい	はい
AWS Elastic Beanstalk	はい	いいえ	はい	はい
AWS Elastic Disaster Recovery	いいえ	なし	はい	あり

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
Elastic Load Balancing	部分的	部分的	はい	いいえ
Amazon ElastiCache	はい	はい	いいえ	はい
AWS Elemental MediaConvert	いいえ	なし	はい	はい
AWS Elemental MediaConnect	はい	はい	はい	部分的
AWS エンドユーザーメッセージング ソーシャル	あり	いいえ	はい	いいえ
AWS Entity Resolution	はい	いいえ	はい	はい
Amazon EventBridge	いいえ	なし	はい	はい
AWS Fargate	はい	いいえ	はい	はい
Amazon FSx	なし	なし	はい	はい
Amazon GameLift Streams	あり	いいえ	はい	はい
AWS Global Accelerator	はい	いいえ	はい	
AWS Glue	はい	いいえ	なし	はい
AWS Glue DataBrew	いいえ	なし	はい	はい
Amazon Managed Grafana [3]	あり	いいえ	はい	あり
AWS Ground Station [4]	あり	いいえ	はい	はい

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
Amazon GuardDuty	いいえ	なし	はい	はい
AWS HealthImaging	いいえ	なし	はい	はい
AWS HealthLake	いいえ	なし	はい	はい
AWS HealthOmics	いいえ	なし	はい	あり
AWS Identity and Access Management (IAM)	はい	はい	はい	いいえ
AWS IAM Access Analyzer	はい	いいえ	はい	はい
AWS IAM アイデンティティセンター	はい	いいえ	はい	
AWS IAM Roles Anywhere	いいえ	なし	はい	あり
Amazon Inspector	あり	はい	はい	はい
Amazon Interactive Video Service (IVS) [5]	はい	いいえ	はい	はい
AWS IoT Core	はい	いいえ	はい	はい
AWS IoT Device Defender	はい	いいえ	はい	いいえ
AWS IoT Device Management	はい	いいえ	はい	いいえ
AWS IoT FleetWise	はい	いいえ	はい	はい

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
AWS IoT Greengrass	はい	いいえ	はい	いいえ
AWS IoT SiteWise	はい	いいえ	はい	はい
AWS IoT TwinMaker	はい	いいえ	はい	はい
AWS IoT Wireless	はい	いいえ	はい	はい
Amazon Kendra	いいえ	なし	はい	いいえ
AWS Key Management Service	はい	部分的	あり	はい
Amazon Keyspaces	はい	はい	はい	はい
Amazon Keyspaces CDC streams	はい	はい	はい	あり
Amazon Kinesis Data Streams	あり	いいえ	はい	はい
AWS Lake Formation	いいえ	いいえ	なし	はい
AWS Lambda	はい	いいえ	はい	はい
AWS Launch Wizard	いいえ	なし	はい	はい
AWS License Manager	いいえ	なし	はい	はい
Amazon Lightsail	はい	はい	はい	はい
Amazon Location Service	いいえ	なし	はい	はい
Amazon MQ	いいえ	なし	はい	はい

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
Amazon MWAA	いいえ	なし	はい	あり
Amazon Macie	あり	いいえ	はい	はい
AWS Mainframe Modernization	はい	いいえ	はい	はい
Amazon Managed Grafana	いいえ	なし	はい	あり
Amazon Managed Service for Prometheus	あり	いいえ	はい	はい
AWS Migration Hub Orchestrator	いいえ	なし	はい	はい
AWS Network Firewall	はい	はい	いいえ	はい
AWS Network Manager	はい	いいえ	はい	いいえ
Amazon OpenSearch Service	はい	いいえ	はい	
AWS Organizations	はい	いいえ	はい	はい
AWS Outposts	いいえ	なし	はい	はい
Amazon Personalize	あり	いいえ	はい	あり
Amazon Pinpoint	あり	いいえ	はい	はい
Amazon Polly	あり	いいえ	はい	はい
AWS の料金表	いいえ	なし	はい	いいえ

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
AWS Private Certificate Authority	はい	いいえ	はい	あり
AWS Private CA Connector for Active Directory	あり	いいえ	はい	あり
AWS Private CA Connector for SCEP	あり	いいえ	はい	はい
AWS PrivateLink	はい	はい	はい	
Amazon Q Business	いいえ	なし	はい	いいえ
Amazon RDS	はい	いいえ	はい	いいえ
Amazon RDS Data API	いいえ	なし	はい	はい
Amazon RDS Performance Insights	いいえ	なし	はい	はい
Amazon Redshift	はい	いいえ	はい	
Amazon Rekognition	いいえ	なし	はい	はい
ごみ箱	あり	いいえ	はい	はい
AWS re:Post Private	はい	いいえ	はい	はい
AWS Resource Access Manager	はい	いいえ	はい	はい
AWS Resource Explorer	はい	いいえ	はい	いいえ
AWS Resource Groups	はい	はい	はい	はい

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
AWS Resource Groups Tagging API	はい	はい	はい	あり
Amazon Route 53	あり	はい	はい	はい
Amazon S3	はい	いいえ	はい	いいえ
Amazon S3 on Outposts	いいえ	なし	はい	いいえ
Amazon SageMaker	いいえ	なし	はい	はい
AWS Secrets Manager	はい	いいえ	はい	はい
AWS Security Hub	いいえ	なし	はい	はい
Amazon Security Lake	はい	いいえ	はい	はい
AWS Security Token Service	はい	いいえ	はい	はい
AWS Service Catalog	いいえ	なし	はい	はい
AWS Shield	はい	はい	いいえ	あり
Amazon Simple Email Service	あり	いいえ	はい	あり
Amazon Simple Notification Service	あり	いいえ	はい	あり
Amazon Simple Queue Service	あり	いいえ	はい	あり
Amazon Simple Workflow Service	あり	いいえ	はい	はい

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
AWS Site-to-Site VPN	はい	いいえ	はい	いいえ
AWS Snow Family	いいえ	なし	はい	
AWS Step Functions	はい	いいえ	はい	はい
AWS Storage Gateway	はい	はい	はい	はい
AWS Systems Manager	いいえ	なし	はい	はい
AWS Systems Manager Incident Manager	いいえ	なし	はい	はい
AWS Systems Manager for SAP	いいえ	なし	はい	はい
Amazon Textract	いいえ	なし	はい	はい
Amazon Timestream	いいえ	なし	はい	あり
Amazon Transcribe	あり	はい	はい	はい
AWS Transfer Family [6]	はい	いいえ	はい	はい
AWS Transit Gateway	はい	いいえ	はい	なし
Amazon Translate	あり	はい	はい	はい
AWS Trusted Advisor	いいえ	なし	はい	はい
AWS User Notifications	いいえ	なし	はい	あり
Amazon Verified Permissions	あり	いいえ	はい	はい
VMware Cloud on AWS	いいえ	なし	はい	あり

サービス名	デュアルスタックサポート	IPv6 のみサポート	パブリックエンドポイントの IPv6 サポート	プライベートエンドポイントの IPv6 サポート[1]
Amazon VPC	はい	はい	はい	いいえ
Amazon VPC Lattice	いいえ	なし	はい	はい
AWS WAF	はい	はい	いいえ	
AWS WAFV2	いいえ	なし	はい	はい
AWS Well-Architected Tool	いいえ	なし	はい	はい
Amazon WorkMail	いいえ	なし	はい	はい
Amazon WorkSpaces	はい	いいえ	はい	はい
AWS X-Ray	はい	いいえ	はい	あり

[1] 空のセルは、サービスが AWS PrivateLink と統合されていないことを示します。

[2] このエントリは、AWS AppSync SDK API を介した AWS AppSync GraphIQL および Event API 設定オペレーションに対する IPv6 サポートを表します。IPv6 は、カスタマー管理の AWS AppSync GraphQL および Event API へのクライアント接続に対してはサポートされていません。

[3] このエントリは、ワークスペースの更新やワークスペースのアクセス許可など、Grafana ワークスペース管理オペレーションに対する IPv6 サポートを表します。ダッシュボードの作成や編集、データソースのクエリなど、一般的な Grafana ワークスペースオペレーションに対する IPv6 サポートはありません。

[4] このエントリは、AWS Ground Station API の呼び出しなど、AWS Ground Station コントロールプレーンオペレーションに対する IPv6 サポートを表します。IPv6 は、AWS Ground Station データプレーンではサポートされていないため、データを配信しているリソース (Amazon EC2 インスタンスなど) に IPv4 経由でアクセス可能であることを確認してください。

[5] このエントリは、IVS エンドポイントの呼び出しなど、コントロールプレーンオペレーションに対する IPv6 サポートを表します。

[6] AWS Transfer Family での IPv6 サポートの詳細については、「IPv6 limitations」を参照してください。

追加の IPv6 サポート

コンピューティング

- Amazon EC2 は、Nitro システムに基づくインスタンスを IPv6 専用サブネットに起動することをサポートしています。
- Amazon EC2 は、インスタンスメタデータサービス (IMDS) と Amazon Time Sync Service に IPv6 エンドポイントを提供します。

ゲーム開発

- Amazon GameLift Streams は、Microsoft Windows Server 2022 Base ランタイムでの IPv6 経由のストリーミングをサポートします。

ネットワークとコンテンツ配信

- Amazon VPC は、IPv6 専用サブネットの作成をサポートします。
- Amazon VPC は、サブネットで DNS64 をサポートし、NAT ゲートウェイで NAT64 をサポートすることで、IPv6 AWS リソースが IPv4 リソースと通信できるようにします。

セキュリティ、アイデンティティ、およびコンプライアンス

- Amazon Detective は、ネットワーク関連の検出結果とエンティティプロファイルで IPv6 アドレスをサポートしています。
- AWS Identity and Access Management (IAM) は、IAM の ID ベースのポリシーにおける IPv6 アドレスをサポートします。
- Amazon Macie は、個人を特定できる情報 (PII) の IPv6 アドレスをサポートします。
- Amazon Security Lake は、ログソースとサブスクライバーのすべてのオペレーションにおいて IPv6 アドレスをサポートします。

マネジメントとガバナンス

- AWS CloudTrail レコードには、送信元 IPv6 情報が含まれます。

- AWS CLI v2 は、IPv6 専用クライアントの IPv6 接続経由のダウンロードをサポートします。

詳細情報

- AWS での IPv6
- デュアルスタックおよび IPv6 専用 Amazon VPC リファレンスアーキテクチャ (PDF)

仮想プライベートクラウドを設定する

Amazon Virtual Private Cloud (VPC) は基本的な構成要素であり、ユーザーはこれを使うことで論理的に分離された仮想ネットワークを AWS クラウド内でプロビジョニングすることができます。独自の VPC を作成することで、IP アドレス範囲、サブネット、ルーティングテーブル、接続オプションを定義する機能を含めネットワーク環境を完全に制御できます。

AWS アカウントには、各 AWS リージョンのデフォルト VPC が含まれています。このデフォルトの VPC には、リソースをすばやく起動するための便利なオプションとなる設定が事前に構成されています。ただし、デフォルト VPC は長期的なネットワークのニーズに適合しない場合もあります。そこで、追加の VPC を作成しておくと有利です。

追加の VPC を作成しておくと、新しい AWS アカウントごとにプロビジョニングされるデフォルト VPC に依存する場合に比べてさまざまな利点があります。セルフマネージド型 VPC を使用すると、多層アプリケーションの実装、オンプレミスリソースへの接続、部門またはビジネスユニットごとのワークロードの分離など、特定の要件に厳密に合うようにネットワークトポロジを設計することができます。

さらに、複数の VPC を作成することで、異なるアプリケーションまたはビジネスユニット間のセキュリティと分離性を強化することができます。各 VPC は個別の仮想ネットワークとして機能するため、セキュリティポリシー、アクセスコントロール、ルーティング設定を各環境に合わせて個別に適用することができます。

デフォルト VPC を使用するか、カスタムの VPC を 1 つ (または複数) 作成するかどうかの判断は、最終的に、具体的なアプリケーション要件、セキュリティニーズ、長期的なスケーラビリティ目標に基づいて下します。VPC のインフラストラクチャは、時間をかけて慎重に設計すれば、堅牢で安全、かつ適応性の高いクラウドネットワーク基盤として役立ちます。

内容

- VPC の基本
- VPC の設定オプション
- デフォルト VPC
- VPC を作成する
- VPC 内のリソースを視覚化する
- CIDR ブロックを追加するまたは VPC から削除する
- Amazon VPC の DHCP オプションセット

- VPC の DNS 属性
- VPC のネットワークアドレスの使用状況
- VPC サブネットを他のアカウントと共有する
- VPC をローカルゾーン、Wavelength Zone、または Outpost に拡張する
- VPC の削除
- Console-to-Code を使用して VPC コンソールアクションから Infrastructure-as-Code を生成する

VPC の基本

VPC は、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。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。詳細については、「サブネット」を参照してください。

内容

- VPC の IP アドレスの範囲
- VPC の図
- VPC リソース

VPC の IP アドレスの範囲

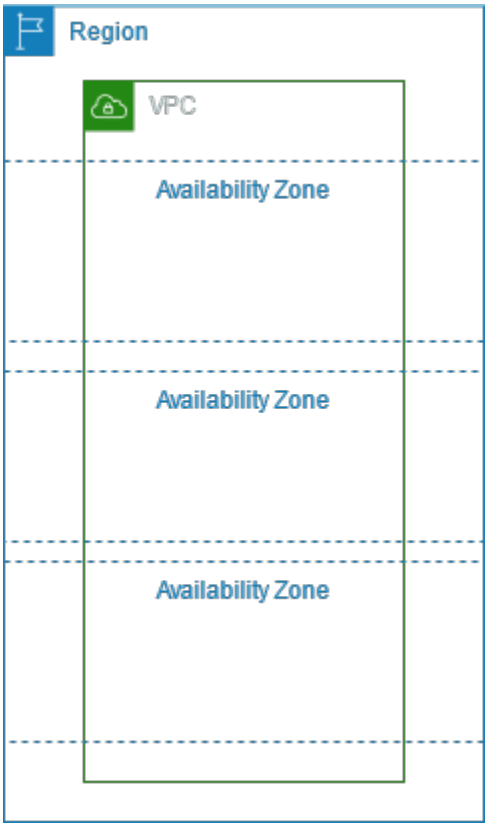
VPC を作成するときは、次のように IP アドレスを指定します。

- IPv4 のみ – VPC には IPv4 CIDR ブロックがありますが、IPv6 CIDR ブロックはありません。
- デュアルスタック – VPC には IPv4 CIDR ブロックと IPv6 CIDR ブロックの両方があります。

詳細については、「VPC とサブネットの IP アドレス指定」を参照してください。

VPC の図

次の図は、追加の VPC リソースのない VPC を示しています。VPC の設定例については、「例」を参照してください。



VPC リソース

各 VPC には次のリソースが自動的に付属します。

- デフォルト DHCP オプションセット
- デフォルトのネットワーク ACL
- デフォルトのセキュリティグループ
- メインルートテーブル

VPC に次のリソースを作成できます。

- ネットワーク ACL
- カスタムルートテーブル
- セキュリティグループ
- インターネットゲートウェイ
- NAT ゲートウェイ

VPC の設定オプション

VPC を作成するときは、次の設定オプションを指定できます。

アベイラビリティーゾーン

AWS リージョンの冗長電源、ネットワーク、および接続を備えた個別のデータセンターです。複数の AZ を使用することで、単一のデータセンターと比較して、可用性、耐障害性、およびスケーラビリティに優れた本番稼働用アプリケーションおよびデータベースを操作することができるようになります。複数の AZ のサブネットで実行されているアプリケーションをパーティショニングすると、停電、落雷、竜巻、および地震などの問題から隔離され、保護されます。

CIDR ブロック

VPC とサブネットの IP アドレス範囲を指定する必要があります。詳細については、「VPC とサブネットの IP アドレス指定」を参照してください。

DNS オプション

サブネットで起動される EC2 インスタンスにパブリック IPv4 DNS ホスト名が必要な場合は、両方の DNS オプションを有効にする必要があります。詳細については、「VPC の DNS 属性」を参照してください。

- [DNS ホスト名を有効化]: VPC 内に起動される EC2 インスタンスが、パブリック IPv4 アドレスに対応するパブリック DNS ホスト名を受信します。
- [DNS 解決を有効化]: プライベート DNS ホスト名の DNS 解決は、Route 53 Resolver と呼ばれる Amazon DNS サーバーによって VPC に提供されます。

インターネットゲートウェイ

VPC をインターネットに接続します。パブリック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のは、インターネット用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がサブネットルートテーブルに含まれているからです。サーバーにインターネットから直接アクセスする必要がない場合は、パブリックサブネットにデプロイしないでください。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。

名前

VPC と他の VPC リソースに指定した名前は、名前タグの作成に使用されます。コンソールで名前タグの自動生成機能を使用する場合、タグ値の形式は *name-resource* です。

NAT ゲートウェイ

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へアウトバウンドトラフィックを送信するのを許可しますが、インターネットがインスタンスに接続するのを禁止します。本番環境では、アクティブな各 AZ に NAT ゲートウェイをデプロイすることをお勧めします。詳細については、「NAT ゲートウェイ」を参照してください。

ルートテーブル

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、ルートと呼ばれる一連のルールが含まれます。詳細については、「ルートテーブル」を参照してください。

サブネット

VPC 内の IP アドレスの範囲。EC2 インスタンスなどの AWS リソースをサブネット内に起動します。各サブネットは 1 つのアベイラビリティーゾーン内に完全に含まれています。2 つ以上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 つのアベイラビリティーゾーンで発生した障害からアプリケーションを保護できます。

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への直接ルートがあります。パブリックサブネット内のリソースは、パブリックインターネットにアクセスできます。プライベートサブネットには、インターネットゲートウェイへの直接ルートがありません。プライベートサブネット内のリソースには、パブリックインターネットへのアクセス用に NAT デバイスなどの別のコンポーネントが必要です。

詳細については、「サブネット」を参照してください。

テナンシー

このオプションは、VPC で起動する EC2 インスタンスを、他の AWS アカウント と共有しているハードウェアで実行するか、または自分専用のハードウェアで実行するかを定義します。VPC のテナンシーで `Default` を選択すると、この VPC で起動された EC2 インスタンスは、インスタンスの起動時に指定されたテナンシーの属性を使用します。詳細については、「Amazon EC2 ユーザーガイド」の「定義済みのパラメータを使用したインスタンスの起動」を参照してください。VPC のテナンシーで `Dedicated` を選択すると、インスタンスは常に、ユーザー専用のハードウェアで実行される、専有インスタンスとして実行されます。AWS Outposts を使用している場合、その Outpost にはプライベート接続が必要となります。つまり、`Default` テナンシーを使用する必要があります。

デフォルト VPC

Amazon VPC の使用開始時には、各 AWS リージョンにデフォルトの VPC があります。デフォルト VPC には、各アベイラビリティーゾーンのパブリックサブネット、インターネットゲートウェイ、および DNS 解決を有効にするための設定があります。そのため、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。

サブネットをデフォルト VPC に追加できます。詳細については、「the section called “サブネットの作成”」を参照してください。

内容

- デフォルト VPC のコンポーネント
- デフォルトサブネット
- デフォルトの VPC とデフォルトのサブネットを使って作業する

デフォルト VPC のコンポーネント

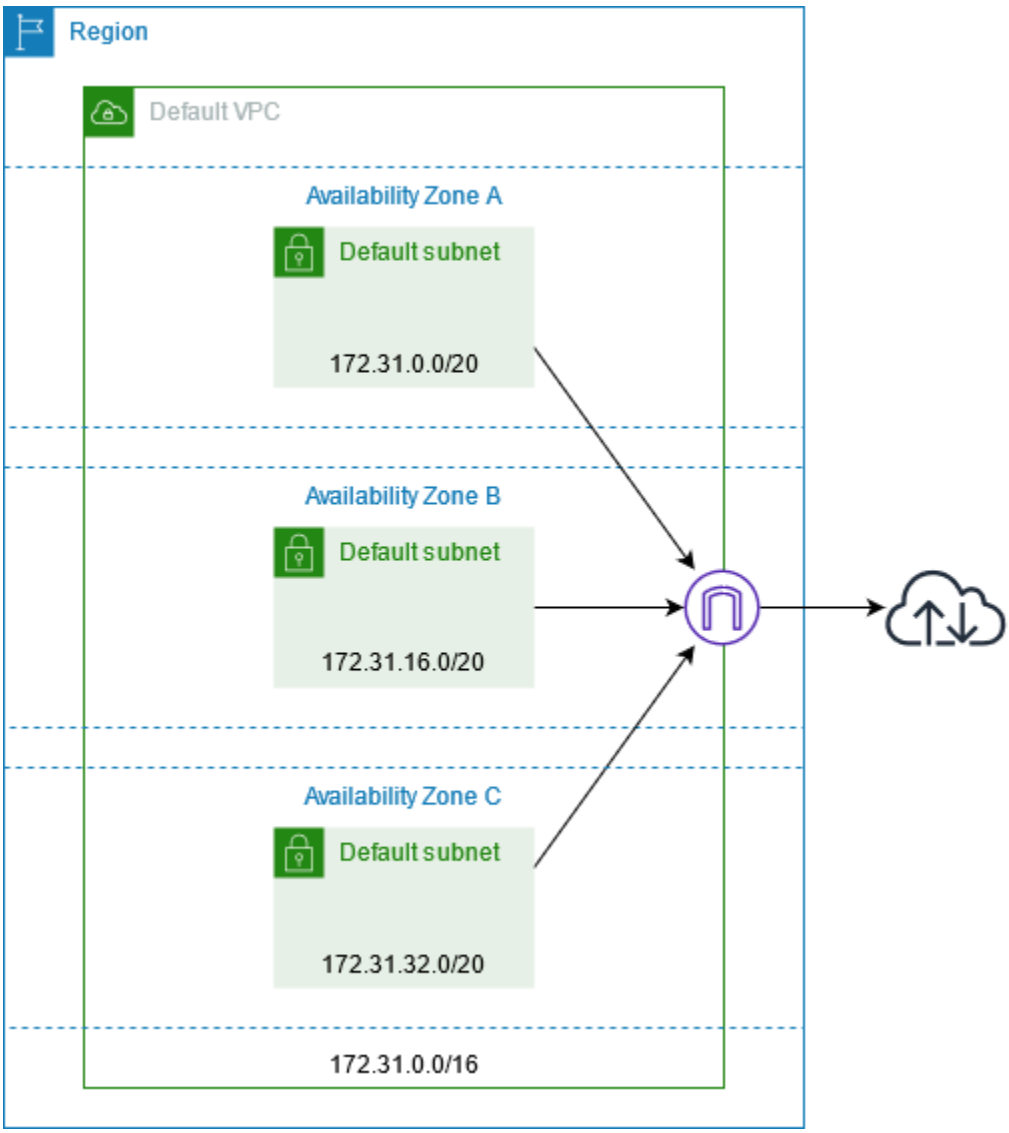
デフォルト VPC を作成するとき、Amazon 側で次の設定を行います。

- 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。インターネットゲートウェイを作成できないようにするには、CreateDefaultVpc と CreateInternetGateway を拒否する必要があります。
> - アカウントのインターネットゲートウェイとの間のすべてのトラフィックをブロックするには、「VPC とサブネットへのパブリックアクセスをブロックする」を参照してください。

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。



次の表は、デフォルト VPC のメインルートテーブルのルートを示しています。

ルーティング先	ターゲット
172.31.0.0/16	ローカル
0.0.0.0/0	*internet_gateway_id*

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。

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、「サブネットの IP アドレス指定属性を変更する」を参照してください

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、「デフォルトのサブネットを作成する」を参照してください。

デフォルトの VPC とデフォルトのサブネットを使って作業する

このセクションでは、デフォルトの VPC とデフォルトのサブネットを使って作業する方法について説明します。

内容

- デフォルト VPC とデフォルトサブネットの表示
- デフォルトの VPC を作成する
- デフォルトのサブネットを作成する
- 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 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。

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。

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。

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。

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

> **Note**
>
> AWS マネジメントコンソール 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。

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、「サブネットからの IPv6 CIDR ブロックを追加または削除する」を参照してください。

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。

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。ただし、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 VPC のサブネットの 1 つを明示的に指定する必要があります。別の VPC がない場合は、サブネットのある

VPC を少なくとも 1 つのアベイラビリティーゾーンで作成する必要があります。詳細については、「VPC を作成する」を参照してください。

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、「デフォルトの VPC を作成する」を参照してください

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」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、「サブネットの IP アドレス指定属性を変更する」を参照してください 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。

VPC を作成する

次のステップを使用して、仮想プライベートクラウド (VPC) を作成します。VPC に AWS リソースを作成する前に、VPC にはサブネット、ルートテーブル、ゲートウェイなどの追加リソースが必要です。

内容

- VPC と他の VPC リソースを作成する
- VPC のみを作成する
- AWS CLI を使用して VPC を作成する

VPC を変更する方法については、「the section called “CIDR ブロックの追加または削除”」を参照してください。

VPC と他の VPC リソースを作成する

次の手順に従って、VPC に加え、サブネット、ルートテーブル、インターネットゲートウェイ、NAT ゲートウェイなど、アプリケーションの実行に必要な追加の VPC リソースを作成します。VPC の設定例については、「例」を参照してください。

コンソールを使用して VPC、サブネット、その他の VPC リソースを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. VPC ダッシュボードで、[Create VPC (VPC を作成する)] を選択します。
3. [Resources to create] (作成するリソース) で、[VPC and more] (VPC など) を選択します。

4. [名前タグの自動生成] を選択したままにすると VPC リソース用の名前タグが作成され、オフにすると VPC リソース用の独自の名前タグが提供されます。
5. [IPv4 CIDR ブロック] に VPC の IPv4 アドレス範囲を入力します。VPC には IPv4 アドレス範囲が必要です。
6. (オプション) IPv6 トラフィックをサポートするには、[IPv6 CIDR ブロック]、[Amazon が提供する IPv6 CIDR ブロック] の順に選択します。
7. [テナンシー] を選択します 。このオプションは、VPC で起動する EC2 インスタンスを、他の AWS アカウント と共有しているハードウェアで実行するか、または自分専用のハードウェアで実行するかを定義します。VPC のテナンシーとして `Default` を選択すると、この VPC で起動した EC2 インスタンスは、インスタンスの起動時に指定したテナンシー属性を使用します。詳細については、「Amazon EC2 ユーザーガイド」の「定義済みのパラメータを使用したインスタンスの起動」を参照してください。VPC のテナンシーで `Dedicated` を選択すると、インスタンスは常に、ユーザー専用のハードウェアで実行される、専有インスタンスとして実行されます。AWS Outposts を使用している場合、その Outpost にはプライベート接続が必要となります。つまり、`Default` テナンシーを使用する必要があります。
8. [アベイラビリティーゾーン (AZ) の数] では、本番環境のため、サブネットを 2 つ以上のアベイラビリティーゾーンでプロビジョニングしておくことが推奨されます。サブネットの AZ を選択するには、[AZ のカスタマイズ] を展開します。それ以外の場合は、AWS で自動的に選択します。
9. サブネットを設定するには、[パブリックサブネットの数] と [プライベートサブネットの数] の値を選択します。サブネットの IP アドレス範囲を選択するには、[サブネット CIDR ブロックをカスタマイズ] を展開します。それ以外の場合は、AWS で自動的に選択します。
10. (オプション) プライベートサブネットのリソースが IPv4 経由でパブリックインターネットにアクセスする必要がある場合、[NAT ゲートウェイ] で、NAT ゲートウェイを作成する AZ の数を選択します。本番環境では、パブリックインターネットへのアクセスを必要とするリソースがある各 AZ に NAT ゲートウェイをデプロイすることをお勧めします。NAT ゲートウェイにはコストが発生することに注意してください。詳細については、「NAT ゲートウェイの料金」を参照してください。
11. (オプション) プライベートサブネット内のリソースが IPv6 経由でパブリックインターネットにアクセスする必要がある場合、[Egress Only インターネットゲートウェイ] で、[はい] をクリックします。
12. (オプション) VPC から Amazon S3 に直接アクセスする必要がある場合は、[VPC エンドポイント]、[S3 ゲートウェイ] の順に選択します。これにより、Amazon S3 用のゲートウェイ VPC エンドポイントが作成されます。詳細については、「AWS PrivateLink ガイド」の「Gateway endpoints」を参照してください。

13. (オプション) [DNS オプション] では、ドメイン名解決の両方のオプションがデフォルトで有効になっています。デフォルトの設定がニーズに合わない場合は、これらのオプションを無効にできます。
14. (オプション) VPC にタグを追加するには、[追加のタグ] を展開して、[新しいタグを追加] を選択し、タグキーとタグ値を入力します。
15. [プレビュー] ペインでは、設定した VPC リソース間の関係を視覚化できます。実線はリソース間の関係を表します。点線は、NAT ゲートウェイ、インターネットゲートウェイ、およびゲートウェイエンドポイントへのネットワークトラフィックを表します。VPC の作成後、[リソースマップ] タブを使用することで、VPC 内のリソースをこの形式でいつでも視覚化できます。詳細については、「VPC 内のリソースを視覚化する」を参照してください。
16. VPC の設定が終了したら、[VPC の作成] を選択します。

VPC のみを作成する

以下の手順で、Amazon VPC コンソールを使用して、追加の VPC リソースのない VPC を作成します。

コンソールを使用して追加の VPC リソースのない VPC を作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. VPC ダッシュボードで、[Create VPC (VPC を作成する)] を選択します。
3. [作成するリソース]で、[VPC のみ] を選択します。
4. (オプション) [名前タグ] に、使用する VPC 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。
5. [IPv4 CIDR block] で、次のいずれかを実行します。
   - [IPv4 CIDR 手動入力] を選択し、VPC の IPv4 アドレス範囲を入力します。
   - [IPAM が割り当てられた IPv4 CIDR ブロック] を選択し、Amazon VPC IP Address Manager (IPAM) の IPv4 アドレスプールとネットマスクを選択します。CIDR ブロックのサイズは、IPAM プールの割り当てルールによって制限されます。IPAM は、AWS ワークロードの IP アドレスを簡単に計画、追跡、およびモニタリングできる VPC 機能です。詳細については、「Amazon VPC IPAM ユーザーガイド」を参照してください。

     IPAM を使用して IP アドレスを管理している場合は、このオプションを選択することをお勧めします。このオプションを選択しないと、VPC に指定した CIDR ブロックが IPAM CIDR 割り当てと重複する可能性があります。

6. (オプション) デュアルスタック VPC を作成するには、VPC の IPv6 アドレス範囲を指定します。[IPv6 CIDR ブロック] で、次のいずれかを実行します。

   - Amazon VPC IP アドレスマネージャーを使用していて、IPAM プールから IPv6 CIDR をプロビジョニングする場合は、[IPAM 割り当て済み IPv6 CIDR ブロック] を選択します。IPAM 割り当ての IPv6 CIDR ブロックを使用して VPC に IPv6 CIDR をプロビジョニングすると、VPC 作成用に連続した IPv6 CIDR の利点を得られます。連続して割り当てられた CIDR とは、続いて割り当てられた CIDR を意味します。これにより、セキュリティルールとネットワークルールを簡素化できます。IPv6 CIDR は、アクセスコントロールリスト、ルートテーブル、セキュリティグループ、ファイアウォールなどのネットワークおよびセキュリティ構造全体で 1 つのエントリに集約できます。

     CIDR ブロックで VPC に IP アドレス範囲をプロビジョニングするには、2 つのオプションがあります。

     - ネットマスク長: CIDR のネットマスク長を選択するには、このオプションを選択します。次のいずれかを行います :

       - IPAM プールにデフォルトのネットマスク長が選択されている場合は、[デフォルトの IPAM ネットマスク長] を選択して、IPAM 管理者が IPAM プールに設定したデフォルトのネットマスク長を使用できます。オプションのデフォルトネットマスク長割り当てルールの詳細については、「Amazon VPC IPAM ユーザーガイド」の「リージョンの IPv6 プールの作成」を参照してください。
       - IPAM プールにデフォルトのネットマスク長が選択されていない場合は、IPAM プール CIDR のネットマスク長よりも具体的なネットマスク長を選択します。例えば、IPAM プールの CIDR が /50 の場合、VPC のネットマスク長は /52 から /60 の間で選択できます。ネットマスク長さは /44 から /60 の間で、/4 刻みです。

     - CIDR の選択: IPv6 アドレスを手動で入力するには、このオプションを選択します。選択できるネットマスクの長さは IPAM プール CIDR のネットマスク長より具体的である必要があります。例えば、IPAM プールの CIDR が /50 の場合、VPC のネットマスク長は /52 から /60 の間で選択できます。IPv6 のネットマスク長は /44 から /60 の間で、/4 ずつ増えます。

   - [Amazon が提供する IPv6 CIDR ブロック] 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] (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。Amazon では IPv6 CIDR ブロックサイズが /56 に固定されています。

   - [自身で所有する IPv6 CIDR] を選択して、AWS に導入した IPv6 CIDR をプロビジョニングします。AWS に独自の IP アドレスを導入する場合の詳細は、「Amazon EC2 ユーザーガイド」の「Amazon EC2 で自分の IP アドレスを使用する (BYOIP)」を参照してくださ

い。[CIDR ブロック] の次のオプションを使用して、VPC の IP アドレス範囲をプロビジョニングできます。

- 指定なし: /56 のネットマスク長を使用する場合は、このオプションを選択します。
- CIDR の選択: このオプションを選択すると、IPv6 アドレスを手動で入力し、BYOIP CIDR のサイズよりも具体的なネットマスク長を選択できます。例えば、BYOIP プール CIDR が /50 の場合、VPC のネットマスク長は /52 から /60 の間で選択できます。IPv6 のネットマスク長は /44 から /60 の間で、/4 刻みです。

7. (オプション) [テナンシー] を選択します 。このオプションは、VPC で起動する EC2 インスタンスを、他の AWS アカウント と共有しているハードウェアで実行するか、または自分専用のハードウェアで実行するかを定義します。VPC のテナンシーで `Default` を選択すると、この VPC で起動された EC2 インスタンスは、インスタンスの起動時に指定されたテナンシーの属性を使用します。詳細については、「Amazon EC2 ユーザーガイド」の「定義済みのパラメータを使用したインスタンスの起動」を参照してください。VPC のテナンシーで `Dedicated` を選択すると、インスタンスは常に、ユーザー専用のハードウェアで実行される、専有インスタンスとして実行されます。AWS Outposts を使用している場合、その Outpost にはプライベート接続が必要となります。つまり、`Default` テナンシーを使用する必要があります。
8. (オプション) VPC にタグを追加するには、[新しいタグを追加] を選択し、タグキーとタグ値を入力します。
9. [Create VPC (VPC の作成)] を選択します。
10. VPC の作成後、サブネットを追加できます。詳細については、「サブネットの作成」を参照してください。

AWS CLI を使用して VPC を作成する

以下の手順には、VPC に加え、アプリケーションの実行に必要な追加の VPC リソースを作成する AWS CLI コマンド例が含まれています。この手順のすべてのコマンドを実行すると、VPC、パブリックサブネット、プライベートサブネット、各サブネットのルートテーブル、インターネットゲートウェイ、エグレス専用インターネットゲートウェイ、パブリック NAT ゲートウェイが作成されます。これらのリソースのすべてを必要としない場合は、必要なサンプルコマンドのみを使用できます。

前提条件

開始する前に、AWS CLI をインストールして設定します。AWS CLI を設定するときには、AWS 認証情報の入力を求められます。この手順の例では、デフォルトのリージョンも設定済みであることを前提としています。設定していない場合は、`--region` オプションを各コマンドに追加します。詳

細については、「AWS CLI のインストールまたは更新」および「AWS CLI の設定」を参照してください。

タグ付け

タグは、create-tags コマンドを使用してリソースを作成した後に、リソースに追加できます。または、以下のようにリソースの作成コマンドに --tag-specification オプションを追加することもできます。

```
--tag-specifications ResourceType=vpc,Tags=[{Key=Name,Value=my-project}]
```

AWS CLI を使用して VPC と VPC リソースを作成するには

1. 以下の create-vpc コマンドを使用して、指定された IPv4 CIDR ブロックを持つ VPC を作成します。

   ```
   aws ec2 create-vpc --cidr-block 10.0.0.0/24 --query Vpc.VpcId --output text
   ```

   または、デュアルスタック VPC を作成するには、次の例に示すように、Amazon が提供する IPv6 CIDR ブロックを追加する --amazon-provided-ipv6-cidr-block オプションを追加します。

   ```
   aws ec2 create-vpc --cidr-block 10.0.0.0/24 --amazon-provided-ipv6-cidr-block --
   query Vpc.VpcId --output text
   ```

   これらのコマンドは、新しい VPC の ID を返します。以下に例を示します。

   ```
   vpc-1a2b3c4d5e6f1a2b3
   ```

2. [デュアルスタック VPC] 以下の describe-vpcs コマンドを使用して、VPC に関連付けられている IPv6 CIDR ブロックを取得します。

   ```
   aws ec2 describe-vpcs --vpc-id vpc-1a2b3c4d5e6f1a2b3 --query
    Vpcs[].Ipv6CidrBlockAssociationSet[].Ipv6CidrBlock --output text
   ```

   以下は出力の例です。

   ```
   2600:1f13:cfe:3600::/56
   ```

3. ユースケースに応じて、1 つ以上のサブネットを作成します。本番環境では、少なくとも 2 つのアベイラビリティーゾーンでリソースを起動することをお勧めします。以下のいずれかのコマンドを使用して各サブネットを作成します。

   - IPv4 専用サブネット — 特定の IPv4 CIDR ブロックを持つサブネットを作成するには、次の create-subnet コマンドを使用します。

     ```
     aws ec2 create-subnet --vpc-id vpc-1a2b3c4d5e6f1a2b3 --cidr-block 10.0.1.0/20
      --availability-zone us-east-2a --query Subnet.SubnetId --output text
     ```

   - デュアルスタックサブネット — デュアルスタック VPC を作成した場合、次のコマンドに示すように、--ipv6-cidr-block オプションを使用してデュアルスタックサブネットを作成できます。

     ```
     aws ec2 create-subnet --vpc-id vpc-1a2b3c4d5e6f1a2b3 --cidr-block 10.0.1.0/20
      --ipv6-cidr-block 2600:1f13:cfe:3600::/64 --availability-zone us-east-2a --
     query Subnet.SubnetId --output text
     ```

   - IPv6 専用サブネット — デュアルスタック VPC を作成した場合、次のコマンドに示すように、--ipv6-nativeオプションを使用して IPv6 専用サブネットを作成できます。

     ```
     aws ec2 create-subnet --vpc-id vpc-1a2b3c4d5e6f1a2b3 --ipv6-native --ipv6-
     cidr-block 2600:1f13:cfe:3600::/64 --availability-zone us-east-2a --query
      Subnet.SubnetId --output text
     ```

   これらのコマンドは、新しいサブネットの ID を返します。以下に例を示します。

   ```
   subnet-1a2b3c4d5e6f1a2b3
   ```

4. ウェブサーバーまたは NAT ゲートウェイにパブリックサブネットが必要な場合は、次の操作を行います。

   a. 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。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。

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

   b. 以下の 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。前の手順で返されたインターネットゲートウェイ ID を使用します。

```
aws ec2 attach-internet-gateway --vpc-id vpc-1a2b3c4d5e6f1a2b3 --internet-
gateway-id igw-id
```

c. 以下の create-route-table コマンドを使用して、パブリックサブネットのカスタムルートテーブルを作成します。このコマンドは、新しいルートテーブルの ID を返します。

```
aws ec2 create-route-table --vpc-id vpc-1a2b3c4d5e6f1a2b3 --query
 RouteTable.RouteTableId --output text
```

d. 以下の create-route コマンドを使用して、すべての IPv4 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ルートテーブルに作成します。パブリックサブネット用のルートテーブル ID を使用します。

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

e. 以下の associate-route-table コマンドを使用して、ルートテーブルをパブリックサブネットに関連付けます。パブリックサブネット用のルートテーブル ID、パブリックサブネットの ID を使用します。

```
aws ec2 associate-route-table --route-table-id rtb-id-public --subnet-
id subnet-id-public-subnet
```

5. [IPv6] Egress-Only インターネットゲートウェイを追加すると、プライベートサブネットのインスタンスは 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はインスタンスにアクセスすることはできません。

a. 以下の create-egress-only-internet-gateway コマンドを使用して、エグレス専用インターネットゲートウェイを作成します。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。

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-1a2b3c4d5e6f1a2b3 --
query EgressOnlyInternetGateway.EgressOnlyInternetGatewayId --output text
```

b. 以下の create-route-table コマンドを使用して、プライベートサブネットのカスタムルートテーブルを作成します。このコマンドは、新しいルートテーブルの ID を返します。

```
aws ec2 create-route-table --vpc-id vpc-1a2b3c4d5e6f1a2b3 --query
 RouteTable.RouteTableId --output text
```

c. 以下の create-route コマンドを使用して、すべての IPv6 トラフィックをエグレス専用インターネットゲートウェイに送信するルートをプライベートサブネット用のルートテーブルに作成します。前の手順で返されたルートテーブル ID を使用します。

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

d. 以下の associate-route-table コマンドを使用して、ルートテーブルをプライベートサブネットに関連付けます。

```
aws ec2 associate-route-table --route-table-id rtb-id-private --subnet-
id subnet-id-private-subnet
```

6. プライベートサブネット内のリソースに NAT ゲートウェイが必要な場合は、以下を実行してください。

a. 以下の allocate-address コマンドを使用して、NAT ゲートウェイ用の Elastic IP アドレスを作成します。

```
aws ec2 allocate-address --domain vpc --query AllocationId --output text
```

b. 以下の create-nat-gateway コマンドを使用して、パブリックサブネットに NAT ゲートウェイを作成します。前の手順で返された割り当て ID を使用します。

```
aws ec2 create-nat-gateway --subnet-id subnet-id-public-subnet --allocation-
id eipalloc-id
```

c. (オプション) ステップ 5 でプライベートサブネット用のルートテーブルを既に作成している場合は、このステップをスキップしてください。それ以外の場合は、次の create-route-table コマンドを使用して、プライベートサブネット用のルートテーブルを作成します。このコマンドは、新しいルートテーブルの ID を返します。

```
aws ec2 create-route-table --vpc-id vpc-1a2b3c4d5e6f1a2b3 --query
 RouteTable.RouteTableId --output text
```

d. 以下の create-route コマンドを使用して、すべての IPv4 トラフィックを NAT ゲートウェイに送信するルートをプライベートサブネット用のルートテーブルに作成します。このステップまたはステップ 5 のいずれかで作成したプライベートサブネット用のルートテーブル ID を使用します。

```
aws ec2 create-route --route-table-id rtb-id-private --destination-cidr-
block 0.0.0.0/0 --gateway-id nat-id
```

e. (オプション) ステップ 5 ですでにルートテーブルをプライベートサブネットに関連付けている場合は、このステップをスキップしてください。それ以外の場合は、次の associate-route-table コマンドを使用して、ルートテーブルをプライベートサブネットに関連付けます。このステップまたはステップ 5 のいずれかで作成したプライベートサブネット用のルートテーブル ID を使用します。

```
aws ec2 associate-route-table --route-table-id rtb-id-private --subnet-
id subnet-id-private-subnet
```

VPC 内のリソースを視覚化する

このセクションでは、[リソースマップ] タブを使用して VPC 内のリソースの視覚的表示を閲覧する方法について説明します。リソースマップには以下のリソースが表示されます。

- VPC
- サブネット
  - アベイラビリティーゾーンは文字で表されます。
  - パブリックサブネットは緑色です。
  - プライベートサブネットは青です。

- ルートテーブル
- 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
- NAT ゲートウェイ
- ゲートウェイエンドポイント (Amazon S3 および Amazon DynamoDB)

リソースマップは、VPC 内のリソース間の関係と、サブネットから NAT ゲートウェイ、インターネットゲートウェイ、およびゲートウェイエンドポイントにトラフィックが流れる方法を表示します。

リソースマップを使用することで、VPC のアーキテクチャを理解するとともに、VPC に何個のサブネットが含まれるか、どのサブネットがどのルートテーブルに関連付けられているか、どのルートテーブルに NAT ゲートウェイ、インターネットゲートウェイ、およびゲートウェイエンドポイントへのルートがあるかを確認できます。

リソースマップは、NAT ゲートウェイから切断されたプライベートサブネットや、インターネットゲートウェイへの直接的なルートを持つプライベートサブネットなど、望ましくない、または不適切な設定を見つけるために使用することもできます。リソースマップ内でルートテーブルなどのリソースを選択し、これらのリソースの設定を編集することも可能です。

VPC 内のリソースを視覚化する

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] を選択します。
3. VPC を選択します。
4. [リソースマップ] タブを選択すると、視覚化されたリソースが表示されます。
5. [詳細を表示] を選択すると、デフォルトで表示されるリソース ID とゾーンに加えて詳細が表示されます。

   - VPC: VPC に割り当てられた IPv4 と IPv6 CIDR の範囲。
   - サブネット: 各サブネットに割り当てられた IPv4 と IPv6 CIDR の範囲。
   - [ルートテーブル]: サブネットの関連付けと、ルートテーブル内のルート数
   - [ネットワーク接続]: 各タイプの接続に関連する詳細:
     - VPC にパブリックサブネットがある場合は、インターネットゲートウェイを使用するトラフィックのルート数と送信元および送信先サブネットを含むインターネットゲートウェイリソースがあります。
     - Egress-only インターネットゲートウェイがある場合は、Egress-only インターネットゲートウェイを使用するトラフィックのルート数と送信元および送信先サブネットを含む Egress-only インターネットゲートウェイリソースがあります。
     - NAT ゲートウェイがある場合は、その NAT ゲートウェイのネットワークインターフェイス数と Elastic IP アドレスを含む NAT ゲートウェイリソースがあります。
     - ゲートウェイエンドポイントがある場合は、エンドポイントを使用して接続できる AWS サービス (Amazon S3 または Amazon DynamoDB) の名前のゲートウェイエンドポイントリソースがあります。
6. リソースにカーソルを合わせると、リソース間の関係を確認できます。実線はリソース間の関係を表します。点線は、ネットワーク接続へのネットワークトラフィックを表します。

CIDR ブロックを追加するまたは VPC から削除する

このセクションでは、IPv4 および IPv6 CIDR ブロックを追加する、または VPC から削除する方法について説明します。

> **Important**
>
> - デフォルトでは、VPC に IPv4 と IPv6 CIDR ブロックを最大 5 つまで追加できますが、この上限は調整可能です。詳細については、「Amazon VPC クォータ」を参照してください。VPC に追加できる CIDR ブロックの上限に関する詳細は、「VPC CIDR ブロック」を参照してください。
> - VPC に複数の IPv4 CIDR ブロックが関連付けられている場合は、VPC から IPv4 CIDR ブロックを削除できます。プライマリ IPv4 CIDR ブロックを削除することはできません。CIDR ブロック全体を削除する必要があります。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を削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。
> - VPC で IPv6 が不要になっても、IPv4 リソース作成して通信するために VPC を引き続き使用する場合は、IPv6 CIDR ブロックを削除できます。
> - IPv6 CIDR ブロックを削除するには、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。
> - IPv6 CIDR ブロックを削除しても、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。

コンソールを使用して CIDR ブロックを追加するまたは VPC 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、[Your VPCs (お使いの VPC)] 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. CIDR を削除するには、CIDR の横にある [削除] を選択します。
5. CIDR を追加するには、[新しい IPv4 CIDR の追加] または [新しい IPv6 CIDR の追加] を選択します。
6. IPv4 CIDR ブロックに CIDR を追加するには、次のいずれかを実行します。
   - [IPv4 CIDR 手動入力] を選択し、IPv4 CIDR ブロックを入力します。

- [IPAM が割り当てられた IPv4 CIDR] を選択し、IPv4 IPAM プールから CIDR を選択します。
- [保存] を選択します。

7. IPv6 CIDR ブロックに CIDR を追加するには、次のいずれかを実行します。

- Amazon VPC IP アドレスマネージャーを使用していて、IPAM プールから IPv6 CIDR をプロビジョニングする場合は、[IPAM 割り当て済み IPv6 CIDR ブロック] を選択します。CIDR ブロックで VPC に IP アドレス範囲をプロビジョニングするには、2 つのオプションがあります。
  - ネットマスク長: CIDR のネットマスク長を選択するには、このオプションを選択します。次のいずれかを行います：
    - IPAM プールにデフォルトのネットマスク長が選択されている場合は、[デフォルトの IPAM ネットマスク長] を選択して、IPAM 管理者が IPAM プールに設定したデフォルトのネットマスク長を使用できます。オプションのデフォルトネットマスク長割り当てルールの詳細については、「Amazon VPC IPAM ユーザーガイド」の「リージョンの IPv6 プールの作成」を参照してください。
    - IPAM プールにデフォルトのネットマスク長が選択されていない場合は、IPAM プール CIDR のネットマスク長よりも具体的なネットマスク長を選択します。例えば、IPAM プールの CIDR が /50 の場合、VPC のネットマスク長は /52 から /60 の間で選択できます。ネットマスク長さは /44 から /60 の間で、/4 刻みです。
  - CIDR の選択: IPv6 アドレスを手動で入力するには、このオプションを選択します。選択できるネットマスクの長さは IPAM プール CIDR のネットマスク長より具体的である必要があります。例えば、IPAM プールの CIDR が /50 の場合、VPC のネットマスク長は /52 から /60 の間で選択できます。IPv6 のネットマスク長は /44 から /60 の間で、/4 ずつ増えます。
- [Amazon が提供する IPv6 CIDR ブロック] 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] (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。Amazon では IPv6 CIDR ブロックサイズが /56 に固定されています。
- [自身で所有する IPv6 CIDR] を選択して、AWS に導入した IPv6 CIDR をプロビジョニングします。詳細については、「Amazon EC2 ユーザーガイド」の「Amazon EC2 に自分の IP アドレスを持ち込む (BYOIP)」を参照してください。CIDR ブロックで VPC に IP アドレス範囲をプロビジョニングするには、2 つのオプションがあります。
  - 指定なし: /56 のネットマスク長を使用する場合は、このオプションを選択します。
  - CIDR の選択: このオプションを選択すると、IPv6 アドレスを手動で入力し、BYOIP CIDR のサイズよりも具体的なネットマスク長を選択できます。例えば、BYOIP プール CIDR

が /50 の場合、VPC のネットマスク長は /52 から /60 の間で選択できます。IPv6 のネットマスク長は /44 から /60 の間で、/4 刻みです。

- 完了したら [CIDR を選択] を選択します。

8. [閉じる] を選択してください。
9. VPC に CIDR ブロックを追加すると、その新しい CIDR ブロックを使用するサブネットを作成できます。詳細については、「サブネットの作成」を参照してください。

AWS CLI を使用して CIDR ブロックを関連付ける、または VPC から関連付けを解除するには associate-vpc-cidr-block と disassociate-vpc-cidr-block の各コマンドを使用します。

Amazon VPC の DHCP オプションセット

VPC 内のネットワークデバイスは Dynamic Host Configuration Protocol (DHCP) を使用します。DHCP オプションセットを使用すると、仮想ネットワーク内のネットワーク構成の次の側面を制御できます。

- VPC 内のデバイスで使用される DNS サーバー、ドメイン名、または Network Time Protocol (NTP) サーバー。
- DNS 解決が VPC で有効かどうか。

内容

- DHCP とは
- DHCP オプションセットの概念
- DHCP オプションセットの使用

DHCP とは

TCP/IP ネットワーク上のすべてのデバイスには、ネットワークを介して通信するための IP アドレスが必要です。以前は、ネットワーク内の各デバイスに IP アドレスを手動で割り当てる必要がありました。現在、IP アドレスは、Dynamic Host Configuration Protocol (DHCP) を使用して DHCP サーバーによって動的に割り当てられます。

EC2 インスタンスで実行されているアプリケーションは、必要に応じて Amazon DHCP サーバーと通信して、IP アドレスリースまたは他のネットワーク構成情報 (Amazon DNS サーバーの IP アドレスや VPC 内のルーターの IP アドレスなど) を取得できます。

DHCP オプションセットを使用して、Amazon DHCP サーバーによって提供されるネットワーク構成を指定できます。

アプリケーションが Amazon IPv6 DHCP サーバーに直接リクエストすることを要求する VPC 設定がある場合は、次の点に注意してください。

- デュアルスタックサブネット内の EC2 インスタンスは、IPv6 DHCP サーバーからのみその IPv6 アドレスを取得できます。DNS サーバー名やドメイン名など、IPv6 DHCP サーバーから追加のネットワーク構成を取得することはできません。
- IPv6 のみのサブネット内の EC2 インスタンスは、IPv6 DHCP サーバーから IPv6 アドレスを取得することができるとともに、DNS サーバー名やドメイン名などの追加のネットワーク構成情報を取得できます。
- IPv6 のみのサブネットにある EC2 インスタンスで、DHCP オプションセットに "AmazonProvidedDNS" が明示的に記載されている場合、IPv4 DHCP サーバーはネームサーバーとして 169.254.169.253 を返します。オプションセットに "AmazonProvidedDNS" がない場合、IPv4 DHCP サーバーは、オプションセットに他の IPv4 ネームサーバーが記述されているかどうかにかかわらず、アドレスを返しません。

Amazon DHCP サーバーは、プレフィックス委任を使用して、IPv4 または IPv6 プレフィックス全体を VPC のネットワークインターフェイスに提供することもできます (「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)。IPv4 プレフィクス委任は DHCP レスポンスでは提供されません。インターフェイスに割り当てられた IPv4 プレフィックスは、IMDS を使用して取得できます (「Amazon EC2 ユーザーガイド」の「インスタンスメタデータカテゴリ」を参照してください)。

DHCP オプションセットの概念

DHCP オプションセットは、EC2 インスタンスなど、VPC 内のリソースが仮想ネットワーク経由で通信するために使用するネットワーク設定のグループです。

各リージョンにデフォルトの DHCP オプションセットがあります。カスタム DHCP オプションセットを作成して VPC に関連付けるか、DHCP オプションセットなしで VPC を設定しない限り、各 VPC はそのリージョンのデフォルトの DHCP オプションセットを使用します。

VPC に DHCP オプションセットが設定されていない場合:

- Nitro System 上に構築された EC2 インスタンスの場合、AWS は 169.254.169.253 をデフォルトのドメインネームサーバーとして設定します。

- Xen 上に構築された EC2 インスタンスの場合、ドメインネームサーバーは設定されません。また、VPC 内のインスタンスは DNS サーバーにアクセスできないため、インターネットにアクセスできません。

DHCP オプションセットを複数の VPC に関連付けることはできますが、各 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。

内容

- デフォルト DHCP オプションセット
- カスタム DHCP オプションセット

デフォルト DHCP オプションセット

デフォルトの DHCP オプションセットには次の設定が含まれます。

- [ドメインネームサーバー]: ネットワークインターフェイスがドメイン名の解決に使用する DNS サーバー。デフォルトの DHCP オプションセットの場合、これは常に `AmazonProvidedDNS` です。詳細については、「Amazon DNS サーバー」を参照してください。
- [ドメイン名]: ドメインネームシステム (DNS) を使用してホスト名を解決する際にクライアントが使用するドメイン名。EC2 インスタンスに使用されるドメイン名に関する詳細については、「Amazon EC2 インスタンスのホスト名のタイプ」を参照してください。
- IPv6 優先リースタイム: IPv6 が割り当てられた実行中のインスタンスが DHCPv6 リースを更新する頻度。デフォルトのリースタイムは 140 秒です。リースの更新は通常、リースタイムの半分が経過した時点で行われます。

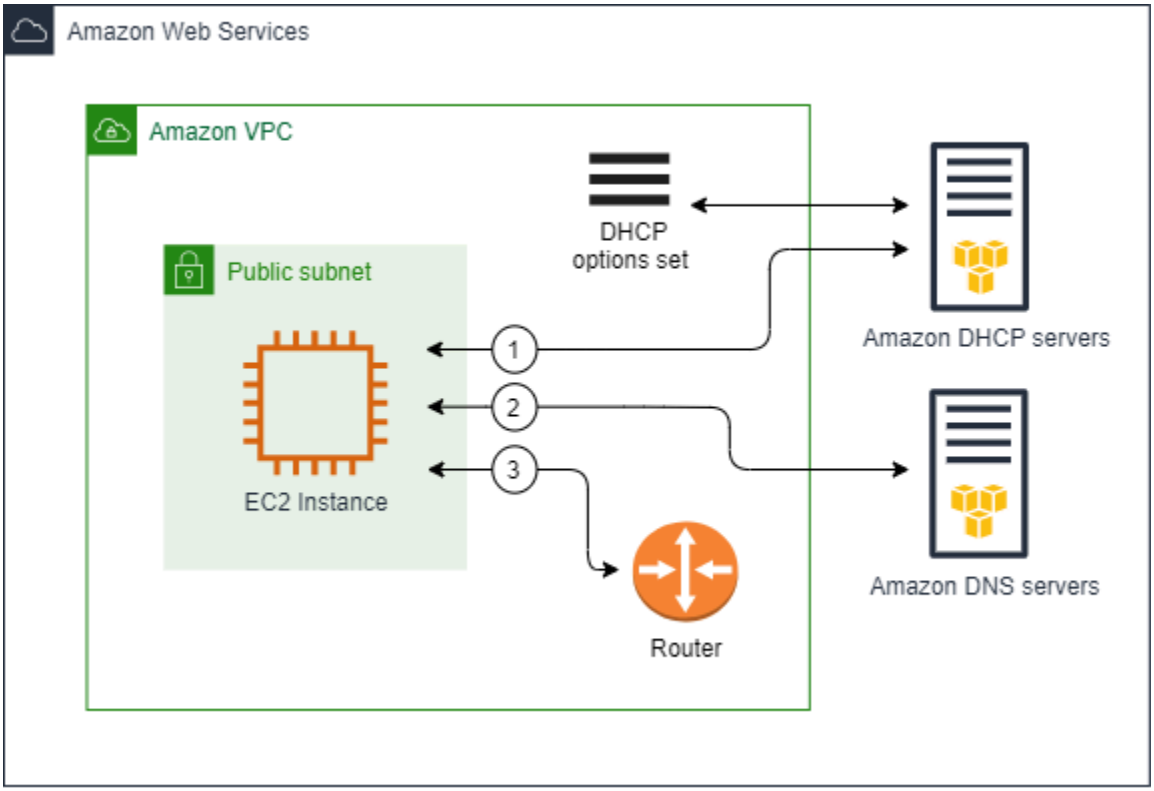
デフォルトの DHCP オプションセットを使用する場合、以下の設定は使用されません。ただし、EC2 インスタンスにはデフォルトがあります。

- [NTP サーバー]: EC2 インスタンスはデフォルトで Amazon Time Sync Service を使用して時刻を取得します。
- [NetBIOS ネームサーバー]: Windows を実行する EC2 インスタンスでは、NetBIOS コンピュータ名は、ネットワーク上でインスタンスを識別するためにそのインスタンスに割り当てられるフレンドリ名です。NetBIOS ネームサーバーは、NetBIOS をネーミングサービスとして使用するネッ

トワークの NetBIOS コンピュータ名とネットワークアドレス間のマッピングのリストを保持します。

- [NetBIOS ノードタイプ]: Windows を実行する EC2 インスタンスでは、これは、インスタンスが NetBIOS 名を IP アドレスに解決するために使用するメソッドです。

デフォルトのオプションセットを使用する場合、Amazon DHCP サーバーはデフォルトのオプションセットのネットワーク設定を使用します。VPC 内でインスタンスを起動すると、インスタンスは、次の図に示すように (1) DHCP サーバーとインタラクションし、(2) Amazon DNS サーバーとインタラクションし、(3) VPC のルーターを介してネットワーク内の他のデバイスに接続します。インスタンスはいつでも Amazon DHCP サーバーとインタラクションして、IP アドレスリースと追加のネットワーク設定を取得できます。



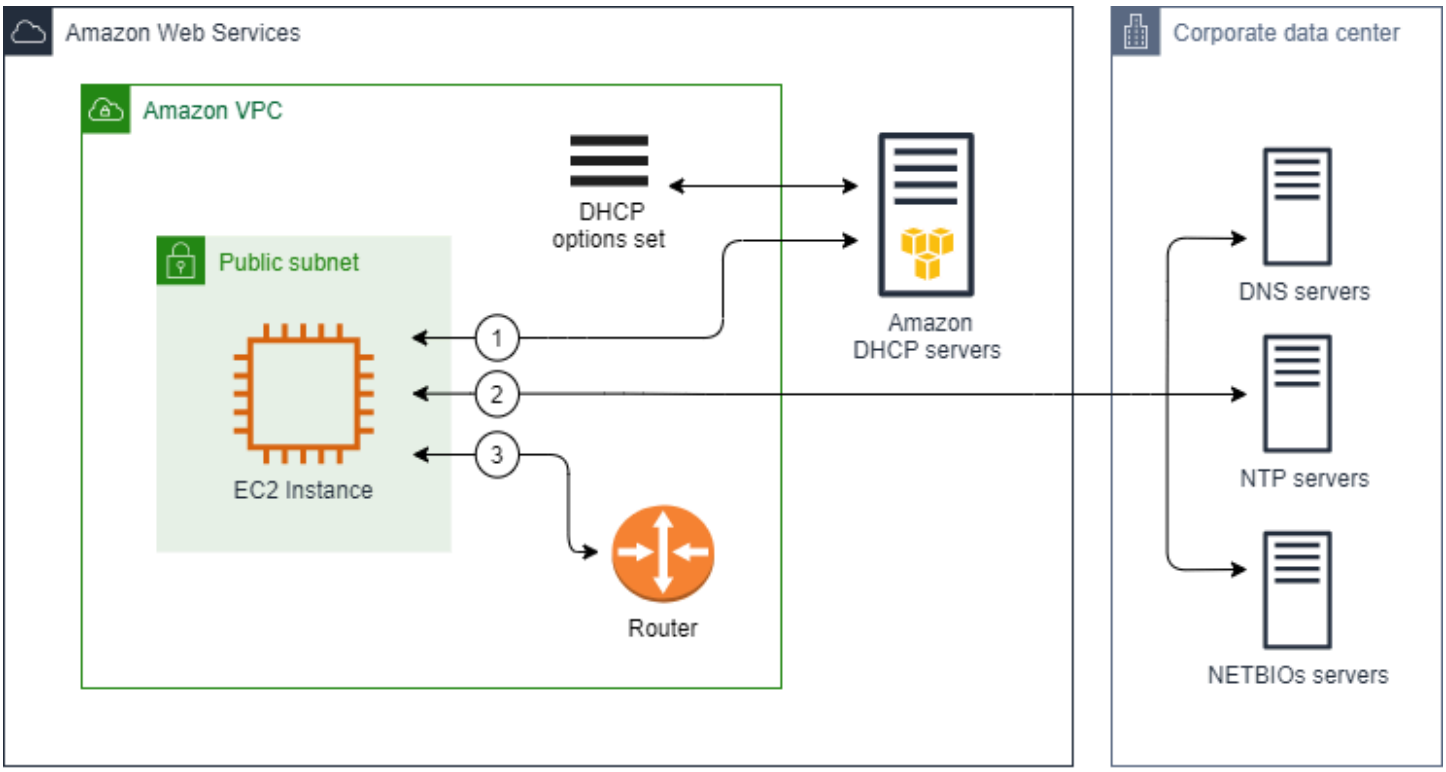
カスタム DHCP オプションセット

次の設定でカスタム DHCP オプションセットを作成し、それを VPC に関連付けることができます。

- [ドメインネームサーバー]: ネットワークインターフェイスがドメイン名の解決に使用する DNS サーバー。
- [ドメイン名]: ドメインネームシステム (DNS) を使用してホスト名を解決する際にクライアントが使用するドメイン名。

- [NTP サーバー]: インスタンスに時間を提供する NTP サーバー。
- [NetBIOS ネームサーバー]: Windows を実行する EC2 インスタンスでは、NetBIOS コンピュータ名は、ネットワーク上でインスタンスを識別するためにそのインスタンスに割り当てられるフレンドリ名です。NetBIOS ネームサーバーは、NetBIOS をネーミングサービスとして使用するネットワークの NetBIOS コンピュータ名とネットワークアドレス間のマッピングのリストを保持します。
- [NetBIOS ノードタイプ]: Windows を実行する EC2 インスタンスでは、インスタンスが NetBIOS 名を IP アドレスに解決するために使用するメソッドです。
- IPv6 優先リースタイム (オプション): IPv6 が割り当てられた実行中のインスタンスが DHCPv6 リースを更新する頻度の値 (秒、分、時、または年単位)。許容値は 140 秒から 4294967295 秒 (約 138 年) です。値を入力しない場合、デフォルトのリースタイムは 140 秒です。EC2 インスタンスに長期アドレス指定を使用すると、リースタイムを長くすることができ、頻繁にリースの更新をリクエストする必要がなくなります。リースの更新は通常、リースタイムの半分が経過した時点で行われます。

カスタムオプションセットを使用する場合、VPC で起動されたインスタンスは、次の図に示すように (1) カスタム DHCP オプションセットのネットワーク設定を使用し、(2) カスタム DHCP オプションセットで指定された DNS、NTP、および NetBIOS サーバーとインタラクションし、(3) VPC のルーターを介してネットワーク内の他のデバイスに接続します。



関連タスク

- DHCP オプションセットを作成する
- VPC に関連付けられたオプションセットを変更する

DHCP オプションセットの使用

DHCP オプションセットを表示する、または使用するには、次の手順を使用してください。DHCP オプションセットの仕組みの詳細については、「the section called “DHCP オプションセットの概念”」を参照してください。

タスク

- DHCP オプションセットを作成する
- VPC に関連付けられたオプションセットを変更する
- DHCP オプションセットを削除する

DHCP オプションセットを作成する

カスタム DHCP オプションセットを使用すると、独自の DNS サーバー、ドメイン名などを使用して VPC をカスタマイズできるようになります。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。

> **Note**
>
> 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 の DHCP オプションを更新するには、新しい DHCP オプションセットを作成して VPC に関連付ける必要があります。

コンソールを使用して DHCP オプションセットを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [DHCP option sets] (DHCP オプションセット) 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。値を入力すると、DHCP オプションセットの [Name] (名前) タグが自動的に作成されます。
5. [DHCP オプション] で、必要な構成設定を入力します。

   - [ドメイン名] (オプション): DNS を使用してホスト名を解決する際にクライアントが使用するドメイン名を入力します。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、「VPC の DNS 属性」を参照してください。

     > **Note**
     >
     > 全面的に管理できるドメイン名のみを使用してください。

     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、Windows や他の Linux オペレーティングシステムでは、この値は単一のドメインとして処理されるため、予期せぬ動作の原因となります。DHCP オプションセットが、単一のドメインとして値を処理するオペレーティングシステムを実行するインスタンスを持つ 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。

   - [Domain name servers] (ドメインネームサーバー (オプション)): ホストの名前からホストの IP アドレスを解決するために使用される DNS サーバーを入力します。

     AmazonProvidedDNS またはカスタムドメインネームサーバーのいずれかを入力できます。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。最大 4 つの IPv4 ドメインネームサーバー (または最大 3 つの IPv4 ドメインネームサーバーと **AmazonProvidedDNS**) と 4 つの IPv6 ドメインネームサーバーの IP アドレスをコンマで区切って入力できます。最大 8 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、制限がより低く設定されている場合があります。AmazonProvidedDNS および Amazon DNS サーバーの詳細については、「Amazon DNS サーバー」を参照してください。

     > **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ー] 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、VPC 内のインスタンスが DNS にアクセスできません。これにより、インターネットアクセスが無効になります。

- [NTP servers] (NTP サーバー) (オプション): 最大 8 つの Network Time Protocol (NTP) サーバー (4 つの IPv4 アドレス、4 つの IPv6 アドレス) の IP アドレスを入力します。

  NTP サーバーは、ネットワークに時間を提供します。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。インスタンスはデフォルトで Amazon Time Sync Service と通信します。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あることに留意してください。

  NTP サーバーオプションの詳細については、「RFC 2132」を参照してください。Amazon Time Sync Service の詳細については、「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。

- [NetBIOS name servers] (NetBIOS ネームサーバー) (オプション): 最大 4 つの NetBIOS ネームサーバーの IP アドレスを入力します。

  Windows OS を実行する EC2 インスタンスでは、NetBIOS コンピュータ名は、ネットワーク上でインスタンスを識別するためにそのインスタンスに割り当てられるフレンドリ名です。NetBIOS ネームサーバーは、NetBIOS をネーミングサービスとして使用するネットワークの NetBIOS コンピュータ名とネットワークアドレス間のマッピングのリストを保持します。

- [NetBIOS node type] (NetBIOS ノードタイプ) (オプション): **1**、**2**、**4**、または **8** を入力します。**2** (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。

  Windows OS を実行する EC2 インスタンスでは、これは、インスタンスが NetBIOS 名を IP アドレスに解決するために使用するメソッドです。デフォルトオプションセットには、NetBIOS ノードタイプの値はありません。

- IPv6 優先リースタイム (オプション): IPv6 が割り当てられた実行中のインスタンスが DHCPv6 リースを更新する頻度の値 (秒、分、時、または年単位)。許容値は 140 秒から 2147483647 秒 (約 68 年) です。値を入力しない場合、デフォルトのリースタイムは 140 秒です。EC2 インスタンスに長期アドレス指定を使用すると、リースタイムを長くすることがで

き、頻繁にリースの更新をリクエストする必要がなくなります。リースの更新は通常、リースタイムの半分が経過した時点で行われます。

6. [Tags] (タグ) を追加します。
7. [Create DHCP options set] を選択します。新しい DHCP オプションセットの名前または ID を書き留めておきます。
8. 新しいオプションセットを使用するように VPC を設定するには、「VPC に関連付けられたオプションセットを変更する」を参照してください。

コマンドラインを使用して VPC の DHCP オプションセットを作成するには

- create-dhcp-options (AWS CLI)
- New-EC2DhcpOption (AWS Tools for Windows PowerShell)

VPC に関連付けられたオプションセットを変更する

DHCP オプションセットを作成したら、それを 1 つ以上の VPC に関連付けることができます。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 VPC に関連付けないと、VPC のドメイン名解決が無効になります。

新しい DHCP オプションセットを VPC に関連付けると、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。

コンソールを使用して、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. ナビゲーションペインで、[Your VPCs (お使いの VPC)] 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 VPC settings] (VPC 設定の編集) の順に選択します。
4. [DHCP options set] (DHCP オプションセット) で、新しい DHCP オプションセットを選択します。または、[DHCP オプションセットがありません] を選択して、VPC のドメイン名解決を無効にします。
5. [保存] を選択します。

コマンドラインを使用して、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

- associate-dhcp-options (AWS CLI)
- Register-EC2DhcpOption (AWS Tools for Windows PowerShell)

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。使用中の DHCP オプションセットは削除できません。削除する DHCP オプションセットに関連付けられている VPC ごとに、異なる DHCP オプションセットを VPC に関連付けるか、DHCP オプションセットを使用しないように VPC を設定する必要があります。詳細については、「the section called “VPC に関連付けられたオプションセットを変更する”」を参照してください。

コンソールを使用して DHCP オプションセットを削除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [DHCP option sets] (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、[アクション]、[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** と入力し、[DHCP オプションセットの削除] を選択します。

コマンドラインを使用して DHCP オプションセットを削除するには

- delete-dhcp-options (AWS CLI)
- Remove-EC2DhcpOption (AWS Tools for Windows PowerShell)

VPC の DNS 属性

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 とサブネットの IP アドレス指定」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) 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、「Amazon VPC の DHCP オプションセット」を参照してください。

内容

- Amazon DNS について理解する
- EC2 インスタンスの DNS ホスト名を表示する
- VPC の DNS 属性の表示と更新

Amazon DNS について理解する

ユーザーが AWS アーキテクトまたは管理者として対処することになる基本のネットワーク要素の 1 つが、Amazon DNS サーバーです。Route 53 Resolver ともいいます。この DNS リゾルバーサービスは、AWS リージョン内の各アベイラビリティーゾーンにネイティブに統合されており、仮想プライベートクラウド (VPC) 内でのドメイン名解決のために信頼性の高いスケーラブルなソリューションを提供しています。このセクションでは、Amazon DNS サーバーの IP アドレス、解決可能なプライベート DNS ホスト名、その使用を管理するルール、について説明します。

内容

- Amazon DNS サーバー
- ルールと考慮事項
- EC2 インスタンスの DNS ホスト名
- VPC の DNS 属性
- DNS クォータ
- プライベートホストゾーン

Amazon DNS サーバー

Route 53 Resolver (「Amazon DNS サーバー」または「AmazonProvidedDNS」とも呼ばれます) は、AWS リージョン内の各アベイラビリティーゾーンに組み込まれている DNS リゾルバーサービスです。Route 53 Resolver は `169.254.169.253` (IPv4)、`fd00:ec2::253` (IPv6)、および VPC +2 にプロビジョニングされたプライマリプライベート IPV4 CIDR 範囲に配置されています。例えば、IPv4 CIDR が `10.0.0.0/16` で、IPv6 CIDR が `2001:db8::/32` の VPC がある場合、Route 53 Resolver には `169.254.169.253` (IPv4)、`fd00:ec2::253` (IPv6)、または `10.0.0.2` (IPv4) でアクセスできます。VPC 内のリソースは DNS クエリにリンクローカルアドレスを使用します。これ

らのクエリは Route 53 Resolver にプライベート転送されるため、ネットワーク上では表示されません。IPv6 専用サブネットでは、"AmazonProvidedDNS" が DHCP オプションセット内のネームサーバーである限り、IPv4 リンクローカルアドレス (169.254.169.253) に引き続きアクセスできます。

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。

プライベート DNS ホスト名の形式は、EC2 インスタンスを起動したときの設定方法によって異なります。プライベート DNS ホスト名のタイプの詳細については、Amazon EC2 ユーザーガイドの「Amazon EC2 インスタンスのホスト名タイプ」を参照してください。

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。
- 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。

EC2 インスタンスの 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 の DNS 属性」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解決されます。

プライベート IPv4 DNS 名

インスタンスのプライベートIPv4 DNSホスト名は、そのプライベートIPv4アドレスに解決されます。プライベート IIPv4 DNS ホスト名は、同じ VPC 内または接続された VPC 内のインスタンス間の通信に使用できます。詳細については、「Amazon EC2 ユーザーガイド」の 「プライベートIPv4アドレス」を参照してください。

パブリック IPv4 DNS 名

インスタンスのパブリック IPv4 DNS ホスト名は、パブリック IPv4 アドレス (インスタンスのネットワーク外部) またはプライベート IPv4 アドレス (インスタンスのネットワーク内部) に解決されます。詳細については、「Amazon EC2 ユーザーガイド」の「パブリック IPv4 アドレス」を参照してください。

VPC ピアリング接続経由でパブリック IPv4 DNS 名をプライベート IPv4 アドレスに解決するには、ピアリング接続の DNS 解決を有効にする必要があります。詳細については、「VPC ピアリング接続の DNS 解決を有効にする」を参照してください。

プライベートリソース DNS 名

このインスタンスで選択された A および AAAA DNS レコードに解決できる RBN ベースの DNS 名。この DNS ホスト名は、デュアルスタックおよび IPv6 専用サブネットのインスタンスの詳細で表示されます。RBN の詳細については、[Amazon EC2 ユーザーガイド]の [EC2 インスタンスホスト名のタイプ]を参照してください。

VPC 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」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。 VPC がデフォルト VPC でない限り、この属性のデフォルトは `false` です。この属性における以下のルールと考慮事項に注意してください。
`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」を参照してください。 この属性のデフォルトは `true` です。この属性における以下のルールと考慮事項に注意してください。

ルールと考慮事項

- 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。
  - 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。
  - 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。Route 53 Resolver サーバーを使用しない場合、カスタムドメインネームサーバーは必要に応じて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。

- Route 53 Resolver は、プライベート DNS ホスト名を、「RFC 1918」に指定されているプライベート IPv4 アドレス範囲外の VPC の IPv4 アドレス範囲を含む、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。ただし、2016 年 10 月より前に作成した VPC の場合、VPC の IPv4 アドレス範囲がこれらの範囲外であると、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、サポート までお問い合わせください。

DNS クォータ

リンクローカルアドレスを使用するサービスには 1,024 パケット/秒 (PPS) の制限があります。この制限には、Route 53 Resolver DNS クエリ、インスタンスメタデータサービス (IMDS) リクエスト、Amazon Time Service Network Time Protocol (NTP) リクエスト、および Windows Licensing Service (Microsoft Windows ベースのインスタンス向け) リクエストの総計が含まれます。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 ユーザーガイド」の「インスタンスメタデータの取得」を参照してください。

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)。

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。

> **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` 属性の両方を `true` に設定する必要があります。

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。リソースに接続する際はこれらのホスト名を理解しておくことが重要です。

[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、「the section called “VPC の DNS 属性”」を参照してください。

インスタンス

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

- 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

- describe-network-interfaces (AWS CLI)
- Get-EC2NetworkInterface (AWS Tools for Windows PowerShell)

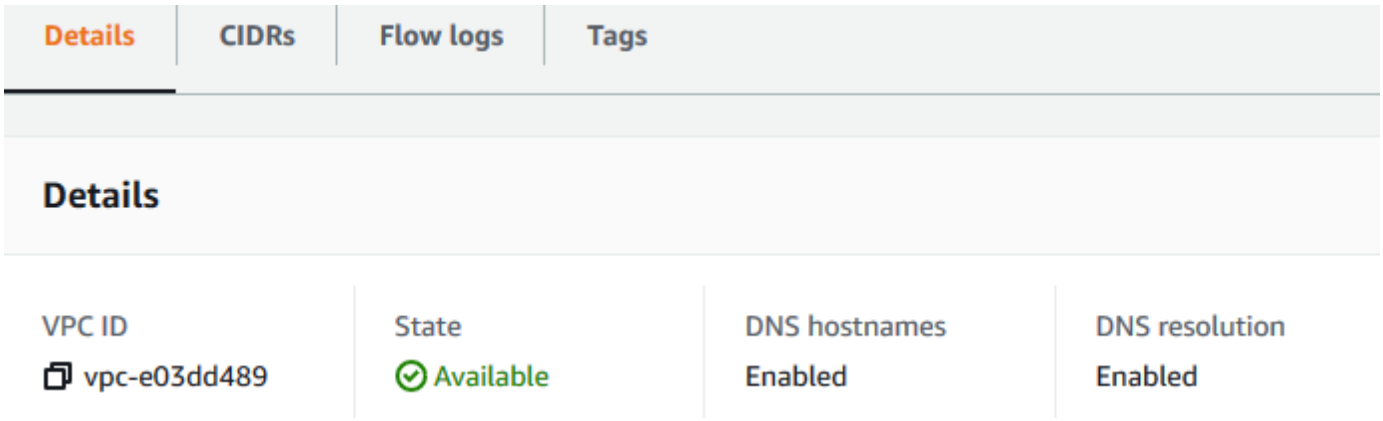
VPC の DNS 属性の表示と更新

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。これらの設定により、インスタンスでパブリック DNS ホスト名を取得するかどうか、また、Amazon DNS サーバーがプライベート DNS 名を解決できるかどうかを制御します。VPC 内でのシームレスな通信を確保するために、これらの属性を正しく設定しておくことが不可欠です。

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. ナビゲーションペインで、[Your VPCs (お使いの VPC)] 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) を選択してから [Edit VPC settings] (VPC 設定の編集) を選択します。該当する DNS 属性の [Enable] (有効化) のチェックをオンまたはオフにして、[Save changes] (変更を保存する) 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

- 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

- modify-vpc-attribute (AWS CLI)
- Edit-EC2VpcAttribute (AWS Tools for Windows PowerShell)

VPC のネットワークアドレスの使用状況

ネットワークアドレス使用状況 (NAU) は、仮想ネットワーク内のリソースに適用されるメトリクスで、VPC のサイズを計画および監視するのに役立ちます。各 NAU ユニットは、VPC のサイズを表す合計に寄与します。

次の VPC クォータによって VPC のサイズが制限されるため、VPC の NAU を構成するユニットの総数を把握することが重要です。

- ネットワークアドレスの使用状況 — 1 つの VPC に搭載できる NAU ユニットの最大数。各 VPC は、デフォルトで最大 64,000 NAU ユニットを持つことができます。最大 256,000 までクォータ引き上げをリクエストできます。
- ピアリングされたネットワークアドレスの使用状況 — VPC とそれがピアリングされているすべての VPC の NAU ユニットの最大数。VPC が同じリージョン内の他の VPC とピアリングされている場合、VPC を合わせると、デフォルトで最大 128,000 NAU ユニットを使用できます。最大 512,000 までクォータ引き上げをリクエストできます。このクォータは、異なる AWS アカウントの VPC 間のピアリングなど、同じリージョン内の VPC 間のピアリングに適用されます。異なるリージョンにまたがってピアリングされた VPC は、このクォータには影響しません。

以下の方法で NAU を使用できます。

- 仮想ネットワークを作成する前に NAU ユニットを計算しておくと、複数の VPC にワークロードを分散すべきかどうかを判断しやすくなります。
- VPC を作成したら、NAU のクォータを超えないように、Amazon CloudWatch を使用して VPC の NAU 使用状況を監視します。NAU クォータをモニタリングするには、Amazon CloudWatch アラームを作成することをお勧めします。詳細については、「the section called “CloudWatch のメトリクス”」を参照してください。

NAU クォータの超過

VPC の NAU クォータを超えると、次の API コールはクライアント側の例外で失敗します。

- `RunInstances`
- `AttachNetworkInterface`
- `AssignPrivateIpAddresses`
- `AssignIpv6Addresses`
- `AcceptVpcPeeringConnection`

特定の例外は、どのクォータを超えるかによって異なります。

- `NetworkAddressUsageLimitExceeded` – VPC が NAU クォータを超えている。
- `NetworkAddressUsagePeeredLimitExceeded` – VPC がピア接続された VPC NAU クォータを超えている。

これらの障害は、インスタンスの起動、ネットワークインターフェイスのアタッチ、新しいアドレスの割り当て、VPC ピアリング接続の受け入れ、影響を受ける VPC のワークロードのスケールまたは管理に影響を与えます。中断を回避するには、Amazon CloudWatch で NAU の使用状況を監視し、クォータに達する前に引き上げをリクエストしてください。

NAU の計算方法

NAU の計算方法を理解していれば、VPC のスケーリングを計画するのに役立ちます。

次の表は、VPC 内の NAU 数に含まれるリソースと、各リソースが使用する NAU ユニットの数をまとめたものです。一部の AWS リソースは単一の NAU ユニットとして表されていますが、複数の NAU ユニットとして表されているリソースもあります。この表を使用して、NAU の計算方法を確認できます。

リソース	NAU ユニット
VPC 内にある EC2 インスタンスのネットワークインターフェイスに割り当てられた IPv4 および IPv6 の各プライベートアドレスまたはパブリックアドレス	1
EC2 インスタンスにアタッチされた追加のネットワークインターフェイス	1
ネットワークインターフェイスに割り当てられたプレフィックス	1
AZ (アベイラビリティーゾーン) あたりの Network Load Balancer	6
AZ (アベイラビリティーゾーン) あたりのゲートウェイロードバランサー	6
AZ (アベイラビリティーゾーン) あたりの VPC エンドポイント	6
Transit Gateway アタッチメント	6
Lambda 関数	6
NAT ゲートウェイ	6
EFS マウントターゲット	6

リソース	NAU ユニット
EFA インターフェイス (ENA デバイスを使用した EFA) または EFA 専用インターフェイス	1
Amazon EKS ポッド	1

NAU の例

以下の例は、NAU の計算方法を示します。

例 1 - VPC ピアリングを使用して 2 つの VPC が接続されている

同じリージョン内でピアリングされた VPC は、合計 NAU のクォータに寄与します。

- VPC 1
  - 個別のアベイラビリティーゾーン内の 2 つのサブネットにある 50 個の Network Load Balancer - 600 NAU ユニット
  - 1 つのサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) と、別のサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) - 20,000 ユニット
  - 100 個の Lambda 関数 - 600 NAU ユニット
- VPC 2
  - 個別のアベイラビリティーゾーン内の 2 つのサブネットにある 50 個の Network Load Balancer - 600 NAU ユニット
  - 1 つのサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) と、別のサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) - 20,000 ユニット
  - 100 個の Lambda 関数 - 600 NAU ユニット
- ピアリング NAU 総数: 42,400 ユニット
- デフォルトのピアリング NAU クォータ: 128,000 ユニット

例 2 - Transit Gateway を使用して接続された 2 つの VPC

Transit Gateway を使用して接続されている VPC は、ピアリングされた VPC のように組み合わされた NAU クォータには寄与しません。

- VPC 1
  - 個別のアベイラビリティーゾーン内の 2 つのサブネットにある 50 個の Network Load Balancer - 600 NAU ユニット
  - 1 つのサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) と、別のサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) - 20,000 ユニット
  - 100 個の Lambda 関数 - 600 NAU ユニット
- VPC 2
  - 個別のアベイラビリティーゾーン内の 2 つのサブネットにある 50 個の Network Load Balancer - 600 NAU ユニット
  - 1 つのサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) と、別のサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) - 20,000 ユニット
  - 100 個の Lambda 関数 - 600 NAU ユニット
- VPC あたりの NAU 総数: 21,200 ユニット
- VPC あたりのデフォルト NAU クォータ: 64,000 ユニット

例 3 - クロスリージョン VPC ピアリングを使用して 2 つの VPC が接続されている

異なるリージョン間でピアリングされている VPC は、結合された NAU クォータには影響しません。AWS は、VPC ごとの独自の NAU クォータに対して各 VPC を評価し、クロスリージョンピアのリソースは VPC のピアリングされた NAU のいずれにもカウントされません。この例では、各 VPC の唯一のピアリングが表示されているクロスリージョンピアリングであることを前提としています (追加のリージョン内ピアなし)。

- VPC 1 (リージョン A)
  - 個別のアベイラビリティーゾーン内の 2 つのサブネットにある 50 個の Network Load Balancer - 600 NAU ユニット
  - 1 つのサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) と、別のサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) - 20,000 ユニット
  - 100 個の Lambda 関数 - 600 NAU ユニット
- VPC 2 (リージョン B)

  - 個別のアベイラビリティーゾーン内の 2 つのサブネットにある 50 個の Network Load Balancer - 600 NAU ユニット
  - 1 つのサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) と、別のサブネットにインスタンス 5,000 個 (それぞれ IPv4 アドレスと IPv6 アドレスを持つ) - 20,000 ユニット
  - 100 個の Lambda 関数 - 600 NAU ユニット
- VPC あたりの NAU 総数: 21,200 ユニット
- VPC あたりのデフォルト NAU クォータ: 64,000 ユニット
- VPC あたりのピアリング NAU 数: 21,200 ユニット (VPC 独自のリソース数のみ。クロスリージョンピアは 0)
- VPC あたりのデフォルトのピアリング NAU クォータ: 128,000 ユニット (クロスリージョンピアはこのクォータにはカウントされません)

VPC サブネットを他のアカウントと共有する

VPC サブネット共有を使用すると、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仮想プライベートクラウド (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 サブネットを共有して、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使用できます。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 サブネット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 サブネット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。

VPC サブネット共有に関連するクォータがあります。詳細については、「VPC サブネット共有」を参照してください。

内容

- 共有サブネットの前提条件
- 共有サブネットの操作
- 所有者と参加者の請求と計測
- 所有者および参加者の責任と権限
- AWS リソースと共有 VPC サブネット

共有サブネットの前提条件

このセクションでは、共有サブネットを使用するための前提条件について説明します。

- VPC の所有者および参加者のアカウントは、AWS Organizations によって管理される必要があります。
- 組織の管理アカウントで、AWS RAM コンソールでのリソース共有を有効にしておく必要があります。詳細については、「AWS RAM ユーザーガイド」の「AWS Organizations 内でリソース共有を有効にする」を参照してください。
- リソース共有を作成する必要があります。リソース共有の作成時に、共有するサブネットを指定できます。または次のセクションの手順を使用して、後でリソース共有にサブネットを追加することもできます。詳細については、「AWS RAM ユーザーガイド」の「Create a resource share」を参照してください。

共有サブネットの操作

このセクションでは、AWS コンソールおよび AWS CLI で共有サブネットを使用する方法について説明します。

内容

- サブネットを共有する
- 共有サブネットの共有を解除する
- 共有サブネットの所有者の識別

サブネットを共有する

以下のように、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。さらに、AWS Organizations 間でセキュリティグループを共有できます。詳細については、「AWS Organizations とセキュリティグループを共有する」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

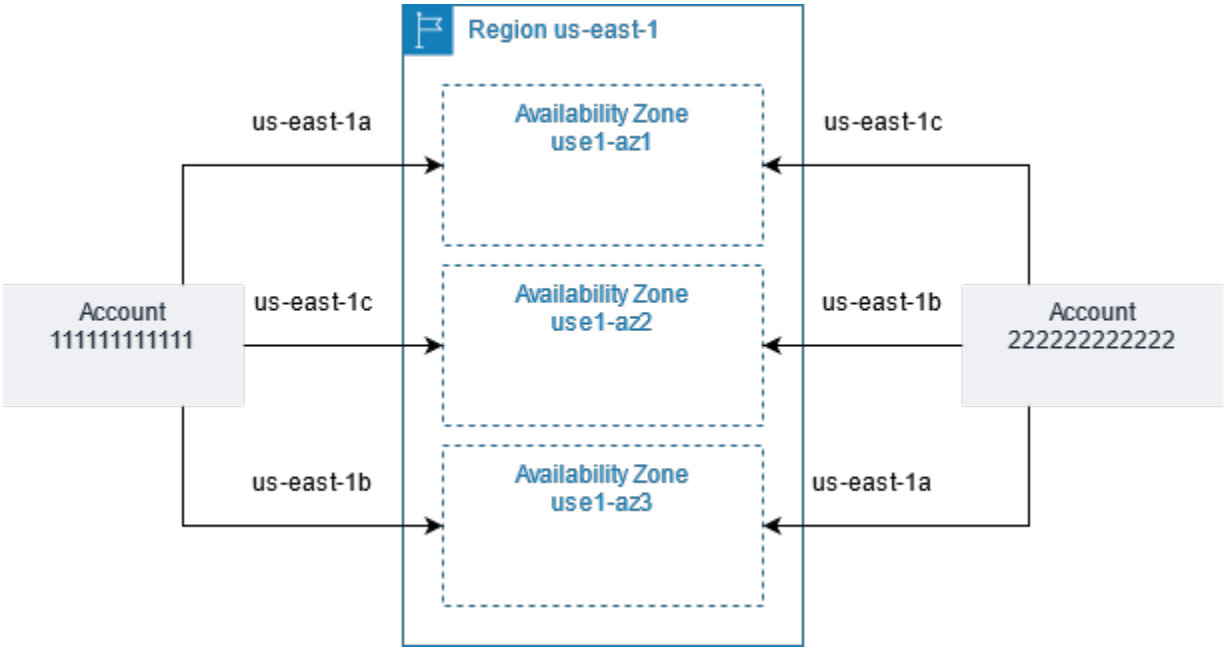
create-resource-share および associate-resource-share コマンドを使用します。

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。例えば、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うちの 1 つの AZ ID です。AZ ID を使用して、アカウント間でリソースの場所を区別できます。Amazon VPC コンソールで、各サブネットの AZ ID を確認できます。

次の図表は、アベイラビリティーゾーンのコードの AZ ID に対するマッピングが異なる 2 つのアカウントを示しています。



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。自動ワークフローやマネージドワークフロー (Auto Scaling やノード交換など) を備えた AWS マネージドサービス (Elastic Load Balancing など) では、一部のリソースで共有サブネットへの継続的なアクセスが必要になる場合があります。
- 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。

所有者と参加者の請求と計測

このセクションでは、共有サブネットを所有するユーザーと共有サブネットを使用するユーザーの、請求と計測の詳細について説明します。

- 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間のデータ転送、VPC ピアリング接続を介するデータ転送、インターネットゲートウェイおよび 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支払います。
- 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) 。さらに、共有 VPC で使用されるパブリック IPv4 アドレスについては VPC 所有者に請求されます。詳細は「Amazon VPC の料金」ページの「パブリック IPv4 アドレス」タブを参照してください。
- 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。

所有者および参加者の責任と権限

このセクションでは、共有サブネットを所有するユーザー (所有者) と共有サブネットを使用するユーザー (参加者) の責任とアクセス許可について詳しく説明します。

所有者のリソース

所有者は、自分が所有する VPC リソースに対して責任を負います。VPC 所有者は、共有 VPC に関連付けられたリソースの作成、管理、削除に責任を負います。これらには、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、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 アタッチメントが含まれます。

参加者のリソース

参加者は、自分が所有する VPC リソースに対して責任を負います。参加者は、共有 VPC 内に VPC リソースの限定されたセットを作成できます。例えば、参加者はネットワークインターフェイスおよびセキュリティグループを作成し、自分が所有するネットワークインターフェイスの VPC フローログを有効にできます。参加者が作成した VPC リソースは、所有者アカウントではなく参加者アカウントの VPC クォータにカウントされます。詳細については、「VPC サブネット共有」を参照してください。

VPC リソース

共有 VPC サブネットを使用する場合、VPC リソースには次の責任と権限が適用されます。

フローログ

- 参加者は、共有 VPC サブネット内で所有するネットワークインターフェイスのフローログを作成、削除、および記述できます。
- 参加者は、共有 VPC サブネット内で所有していないネットワークインターフェイスのフローログを作成、削除、または記述することはできません。
- 参加者は、共有 VPC サブネットのフローログを作成、削除、または記述することはできません。
- VPC 所有者は、共有 VPC サブネット内で所有していないネットワークインターフェイスのフローログを作成、削除、および記述できます。
- VPC 所有者は、共有 VPC サブネットのフローログを作成、削除、および記述できます。
- VPC 所有者は、参加者が作成したフローログを記述したり削除したりすることはできません。

インターネットゲートウェイと Egress-Only インターネットゲートウェイ

- 参加者は、共有 VPC サブネットでインターネットゲートウェイと Egress-Only インターネットゲートウェイを作成、接続、削除することはできません。参加者は、共有 VPC サブネット内のインターネットゲートウェイを記述することができます。参加者は、共有 VPC サブネット内の Egress-Only インターネットゲートウェイを記述することができます。

NAT ゲートウェイ

- 参加者は、共有 VPC サブネット内の NAT ゲートウェイを作成、削除、または記述することはできません。

ネットワークアクセスコントロールリスト (NACL)

- 参加者は、共有 VPC サブネット内の NACL を作成、削除、または置き換えることはできません。参加者は、VPC 所有者が共有 VPC サブネットで作成した NACL について記述できます。

ネットワークインターフェイス

- 参加者は共有 VPC サブネットでネットワークインターフェイスを作成できます。参加者は、共有 VPC サブネット内の VPC 所有者が作成したネットワークインターフェイスを、ネットワークインターフェイスの接続、切断、変更など、他の方法で操作することはできません。参加者は、自分が作成した共有 VPC のネットワークインスタンスを変更または削除できます。例えば、参加者は、作成したネットワークインターフェイスの IP アドレスへの関連付けや関連付け解除を実行できます。
- VPC 所有者は、共有 VPC サブネット内の参加者が所有するネットワークインターフェイスを記述することができます。VPC 所有者は、共有 VPC サブネット内の参加者が所有するネットワークインターフェイスのアタッチ、デタッチ、変更など、参加者が所有するネットワークインターフェイスを他の方法で操作することはできません。

ルートテーブル

- 参加者は、共有 VPC サブネット内でルートテーブルを操作 (ルートテーブルの作成、削除、関連付けなど) することはできません。参加者は共有 VPC サブネットでルートテーブルを記述できます。

セキュリティグループ

- 参加者は、共有 VPC サブネット内に所有するセキュリティグループの受信ルールと送信ルールを作成、削除、記述、変更、作成できます。VPC 所有者がセキュリティグループを参加者と共有している場合、参加者は VPC 所有者によって作成されたセキュリティグループを操作できます。
- 参加者は、自分が所有するセキュリティグループ内に、他の参加者または VPC 所有者に属するセキュリティグループを参照するルールを作成できます: account-number/security-group-id
- 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インスタンスを起動することはできません。
- 参加者は、セキュリティグループが共有されていない限り、VPC 所有者または他の参加者が所有するデフォルト以外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することはできません。
- VPC 所有者は、共有 VPC サブネットの参加者が作成したセキュリティグループについて記述できます。VPC 所有者は、参加者が作成したセキュリティグループを他の方法で操作することはできません。たとえば、VPC 所有者は、参加者が作成したセキュリティグループを使用してインスタンスを起動することはできません。

サブネット

- 参加者は共有サブネットの属性、またはそれに関連する属性を変更することはできません。VPC 所有者のみができます。参加者は共有 VPC サブネット内のサブネットを記述できます。
- VPC 所有者は、AWS Organizations 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。VPC 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。

Transit Gateway

- VPC 所有者のみが、共有 VPC サブネットにトランジットゲートウェイを接続できます。参加者はできません。

VPC

- 参加者は VPC の属性、またはそれに関連する属性を変更することはできません。VPC 所有者のみができます。参加者は VPC、その属性、および DHCP オプションセットについて記述できます。
- 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。

- 参加者は、自分のセキュリティグループを共有 VPC に関連付けることができます。これにより、参加者は、共有 VPC 内で所有している Elastic Network Interface でセキュリティグループを使用できます。

AWS リソースと共有 VPC サブネット

次の AWS のサービス は共有 VPC サブネットのリソースをサポートしています。詳細については、対応するサービスドキュメントへのリンクを参照してください。

- Amazon Aurora
- AWS Database Migration Service
- Amazon EC2
- Amazon ECS
- Amazon ElastiCache (Redis OSS)
- Amazon EFS
- Amazon Elastic Kubernetes Service
- Elastic Load Balancing
  - アプリケーション ロード バランサー
  - Gateway Load Balancers
  - Network Load Balancers
- Amazon EMR
- AWS Glue
- AWS Lambda
- Apache MQ (Rabbit MQ ではない) を実行する Amazon MQ
- Amazon MSK
- AWS Network Manager
  - AWS Cloud WAN
  - Network Access Analyzer
  - Reachability Analyzer
- Amazon OpenSearch Service
- AWS PrivateLink†
- Amazon Relational Database Service (RDS)

- Amazon Redshift *
- Amazon Route 53:
- Amazon SageMaker Unified Studio
- AWS Transit Gateway
- AWS Verified Access
- Amazon VPC
  - ピア接続
  - トラフィックのミラーリング。
- Amazon VPC Lattice

† 共有 VPC 内の VPC エンドポイントを使用して PrivateLink をサポートするすべての AWS サービスに接続できます。PrivateLink をサポートするサービスのリストについては、「AWS PrivateLink ガイド」の「AWS PrivateLink と統合する AWS のサービス」を参照してください。

このリストは、共有 VPC サブネットでのリソースの起動をサポートするすべてのサービスを含めることを目的としています。最大限の努力にもかかわらず、共有 VPC サブネットでのリソースの起動をサポートしていても、ここに記載されていないサービスが存在する可能性があります。ご質問がある場合は、ドキュメントのフィードバックを送信することをお勧めします。

VPC をローカルゾーン、Wavelength Zone、または Outpost に拡張する

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 Zone 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。

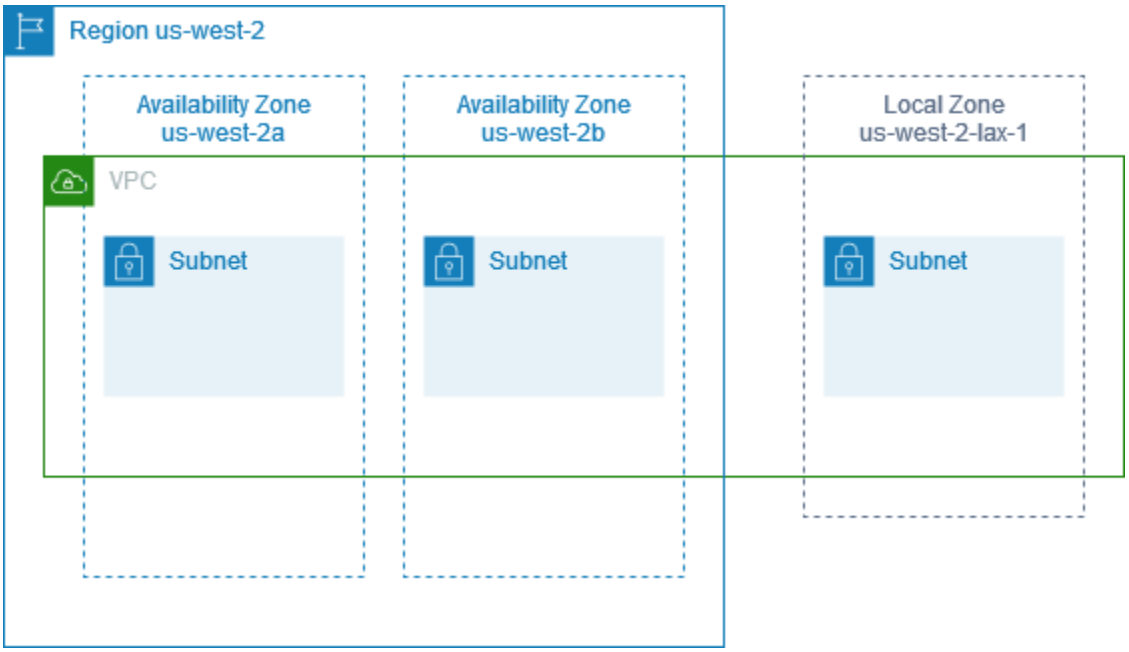
AWS Local Zones 内のサブネット

AWS Local Zones では、リソースを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、次のプロセスを使用します。

- ローカルゾーンにオプトインします。
- ローカルゾーン内にサブネットを作成します。
- ローカルゾーンサブネットでリソースを起動し、アプリケーションとユーザーを近づけます。

次の図は、アベイラビリティーゾーンとローカルゾーンにまたがる米国西部 (オレゴン) (us-west-2) リージョンの VPC を示しています。



VPC を作成する際、Amazon が提供するパブリック IP アドレスのセットを選択して VPC に割り当てることができます。また、アドレスにネットワーク境界グループを設定して、アドレスをそのグループに制限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場合、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。ローカルゾーンのネットワークトラフィックは、ローカルゾーンの親リージョンを経由せずにインターネットまたはポイントオブプレゼンス (POP) に直接送信されます。これにより、低レイテンシーのコンピューティングへのアクセスが可能です。ローカルゾーンとそれに対応する親リージョンの完全なリストについては、「AWS Local Zones ユーザーガイド」の「Available Local Zones」を参照してください。

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。
- 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。
- 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境界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`)。
- ローカルゾーンが IPv6 をサポートしている場合、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。IPv6 をサポートするローカルゾーンのリストについては、「AWS Loal Zones ユーザーガイドの「Considerations」を参照してください。
- ローカルゾーンのサブネット内に VPC エンドポイントを作成することはできません。

Local Zones の操作に関する詳細については、「AWS Local Zones ユーザーガイド」を参照してください。

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、「」を参照してくださいthe section called “Elastic IP アドレス”

  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、「」を参照してくださいthe section called “Elastic IP アドレス”
- 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、「」を参照してくださいthe section called “インターネットゲートウェイへのルーティング”

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 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、Direct Connect の場所への Direct Connect 接続があります。

> **Note**
>
> Direct Connect を使用してローカルゾーンのサブネットを送信先のトラフィックは、ローカルゾーンの親リージョンを経由しません。代わりに、トラフィックはローカルゾーンへの最短経路をたどります。これにより、レイテンシーが減少し、アプリケーションの応答性が向上します。

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 VPC コンソールのサブネットの詳細ページ、または describe-subnets コマンド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、レイテンシーパフォーマンスを最適化するために、サブネットを拡張するローカルゾーンに最も近い Direct Connect ロケーションを使用することをお勧めします。

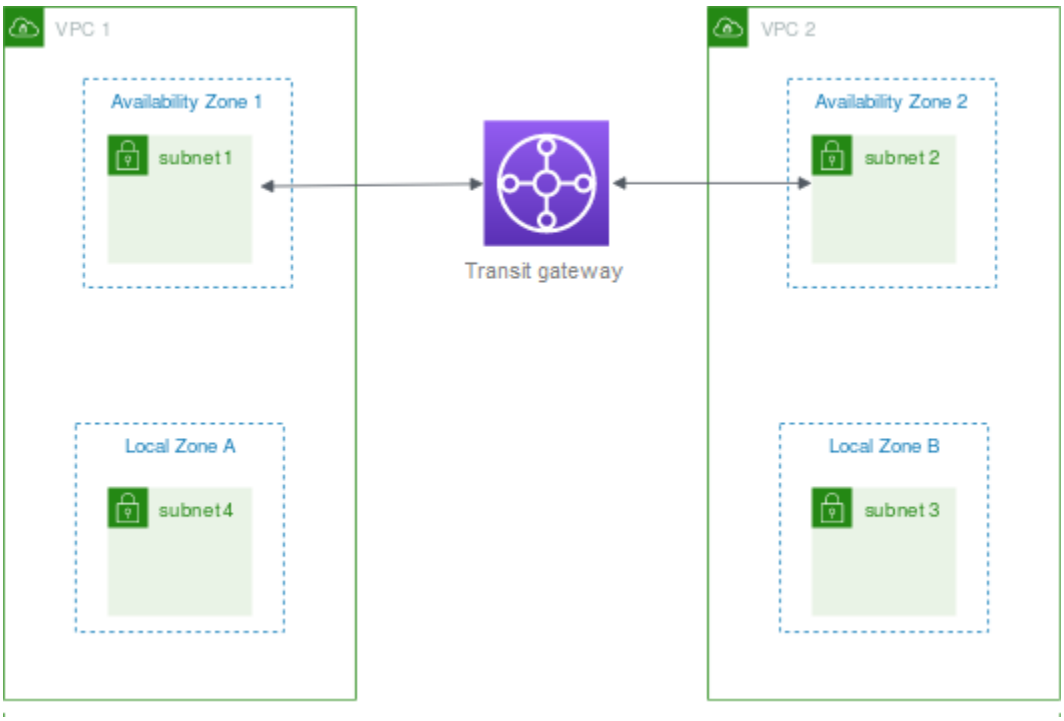
  接続の注文方法については、Direct Connect ユーザーガイドの「クロスコネクト」を参照してください。

- 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、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、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。
- 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、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 Transit Gateway に接続する

ローカルゾーンでサブネットに Transit Gateway 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 Transit Gateway 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 Transit Gateway 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 Transit Gateway 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。

> **Note**
>
> Transit Gateway から発信されるローカルゾーン内のサブネット宛てのトラフィックは、まず親リージョンを通過します。



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 “サブネットの作成”」を参照してください。
- Transit Gateway。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 Transit Gateway アタッチメント。詳細については、「Amazon VPC Transit Gateway」の「VPC への Transit Gateway アタッチメントを作成する」を参照してください。
- Transit Gateway アタッチメントに関連付けられた Transit Gateway 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、他の VPC の CIDR を送信先とし、Transit Gateway アタッチメントに対するネットワークインターフェイスの ID を対象としたローカルゾーンサブネットの、サブネットルートテーブル内のエントリ。Transit Gateway 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、 Transit Gateway 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”」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。

ルーティング先	Target
VPC 1 CIDR	*local*
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。

ルーティング先	Target
VPC 2 CIDR	*local*
VPC 1 CIDR	*vpc2-attachment-network-interface-id*

Transit Gateway 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 Transit Gateway ルートテーブルに伝播されます。

CIDR	添付ファイル	ルートタイプ
VPC 1 CIDR	*VPC 1 ########*	伝播済み
VPC 2 CIDR	*VPC 2 ########*	伝播済み

AWS Wavelength のサブネット

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。デベロッパーは、仮想プライベートクラウド (VPC) を 1 つ以上の Wavelength Zone に拡張し、Amazon EC2 イン

スタンスなどの AWS リソースを使用して、超低レイテンシーを必要としたりリージョンの AWS のサービスに接続したりするアプリケーションを実行できます。

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 Zone 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、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 Zone 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 Zone 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 Zone 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 Zone 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 Zone 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 Zone でサブネットへの Transit Gateway 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、Transit Gateway 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 Zone に関する考慮事項

同じ VPC 内の異なる Wavelength Zone 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 Zone 間の通信が必要な場合、AWS では Wavelength Zone 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 Zone のインスタンス間で通信が可能になります。

Wavelength Zone 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 Zone 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 Zone (Wavelength Zone A と Wavelength Zone 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。

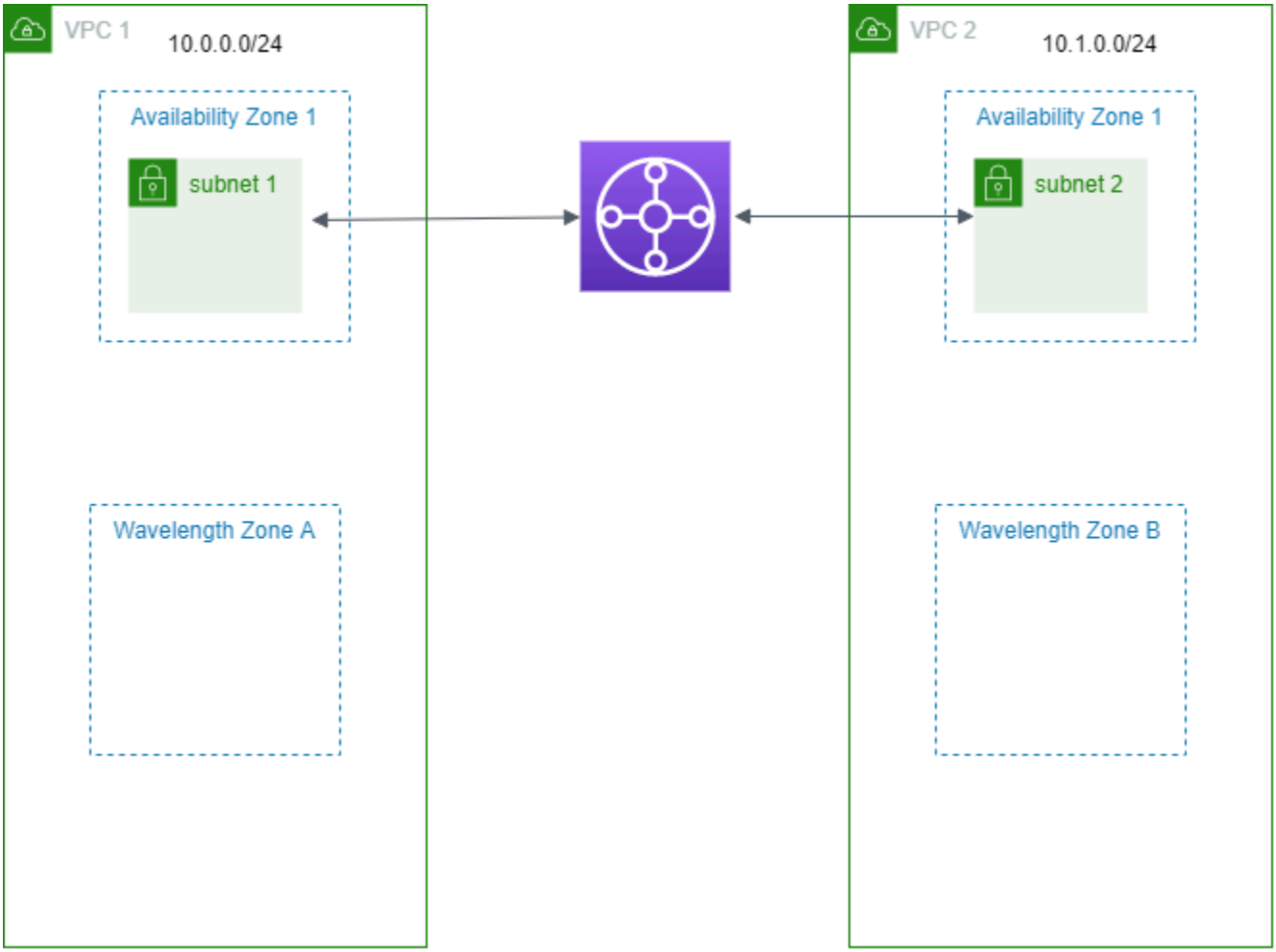
- 各 Wavelength Zone について、その Wavelength Zone の親アベイラビリティーゾーンであるアベイラビリティーゾーン内のサブネット。この例では、サブネット 1 とサブネット 2 を作成します。サブネットの作成の詳細については、「the section called “サブネットの作成”」を参照してください。親ゾーンを確認するには、describe-availability-zones コマンドを使用します。
- Transit Gateway。VPC に接続する Transit Gateway。Transit Gateway の作成方法の詳細については、Amazon VPC Transit Gatewaysの「Transit Gateway の作成」を参照してください。
- Wavelength Zone の親アベイラビリティーゾーン内の Transit Gateway 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 Transit Gateway アタッチメントを作成する」を参照してください。
- Transit Gateway ルートテーブル内の各 VPC のエントリ。Transit Gateway ルートの作成方法の詳細については、Amazon VPC Transit Gateways ガイドの「Transit Gateway 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、Transit Gateway 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、「」を参照してくださいthe section called “トランジットゲートウェイのルーティング”

  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

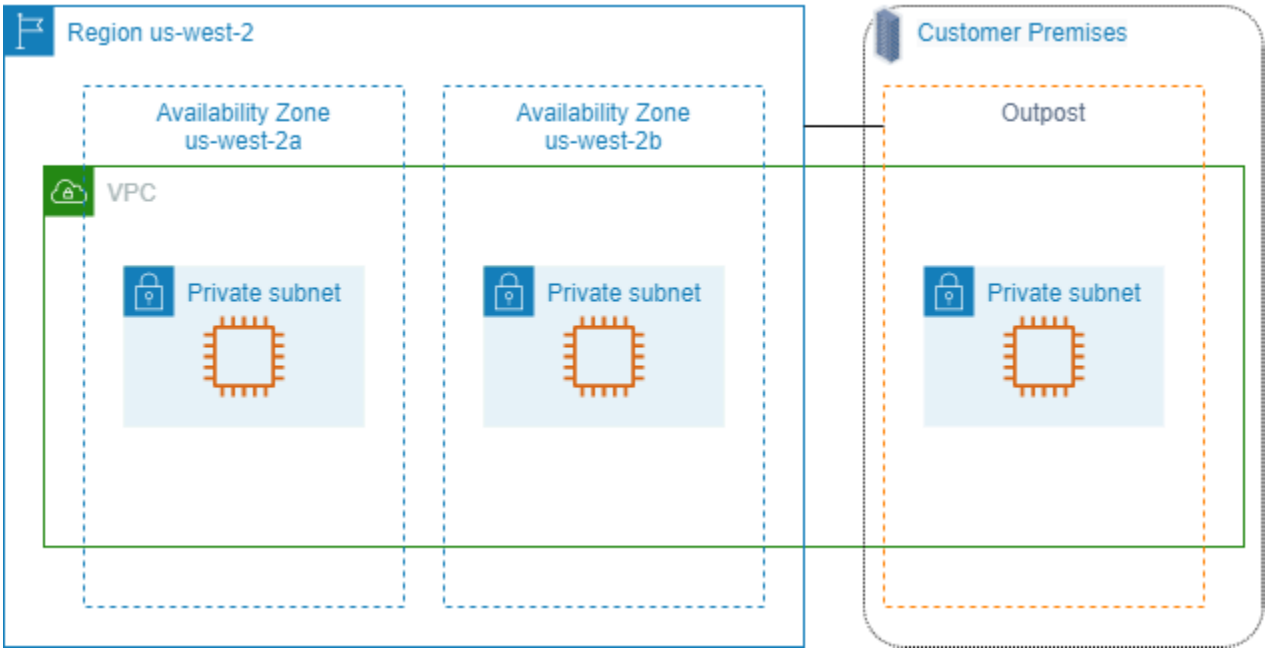
AWS Outposts では、同じ AWS ハードウェアインフラストラクチャ、サービス、API、ツールを提供、オンプレミスやクラウドでアプリケーションを構築して実行することができます。AWS Outposts はオンプレミスのアプリケーションやシステムに対し低レイテンシーのアクセスを必要と

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。の詳細については、AWS Outposts を参照してください。。AWS Outposts

VPC は、AWS 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。Outpost を親リージョンに接続したら、その VPC に Outpost 用のサブネットを作成することで、リージョン内の任意の VPC を Outpost に拡張できます。

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。
- Outpost のサブネットを作成するには、サブネットを作成するときに Outpost の Amazon リソースネーム (ARN) を指定します。
- Outposts ラック - 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。詳細については、「AWS OutpostsOutposts ラックのユーザーガイド」の「Local gateways」を参照してください。
- Outposts サーバー - ローカルネットワークインターフェイスは、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。詳細については、「AWS OutpostsOutposts サーバー用ユーザーガイド」の「Local network interfaces」を参照してください。
- デフォルトでは、Outpost のサブネットを含む VPC で作成するすべてのサブネットは、VPC のメインルートテーブルに暗黙的に関連付けられます。または、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を送信先とす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。



VPC の削除

不要になった VPC は、削除することができます。

要件

VPC を削除する前に、まず VPC でリクエスタマネージドネットワークインターフェイスを作成したリソースを終了または削除する必要があります。例えば、EC2 インスタンスを終了し、ロードバランサー、NAT ゲートウェイ、Transit Gateway の VPC アタッチメント、およびインターフェイス VPC エンドポイントを削除する必要があります。

> **Note**
>
> 削除する VPC のフローログを作成した場合、削除された VPC のフローログは最終的に自動削除されることに留意してください。

内容

- コンソールを使用して VPC を削除する
- コマンドラインを使用して VPC を削除する

コンソールを使用して VPC を削除する

Amazon 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。

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

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、「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. VPC を削除する前に削除または終了する必要があるリソースがある場合は、それらが表示されます。これらのリソースを削除または終了後、もう一度お試しください。そのようなリソースがない場合は、VPC に加えて削除するリソースが表示されます。リストを確認して、次のステップに進みます。
7. (オプション) 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。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、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。
8. 確認を求められたら、**delete**と入力し、[削除] を選択します。

コマンドラインを使用して VPC を削除する

コマンドラインを使って VPC を削除する前に、VPC でリクエスタマネージドネットワークインターフェイスを作成したリソースを、終了または削除します。また、サブネット、セキュリティグループ、ネットワーク ACL、ルートテーブル、インターネットゲートウェイ、エグレス専用インターネットゲートウェイなど、自分で作成したすべての VPC リソースも、削除またはデタッチします。デフォルトのセキュリティグループ、デフォルトのルートテーブル、デフォルトのネットワーク ACL は、削除する必要はありません。

以下の手順は、一般的な VPC リソースを削除した後に VPC を削除する場合のコマンドを示したものです。これらのコマンドは、次の順序で使用する必要があります。追加の VPC リソースを作成した場合は、その VPC を削除する前に、対応する delete コマンドも使用する必要があります。

AWS CLI を使用して VPC を削除するには

1. 以下の delete-security-group コマンドを使用して、セキュリティグループを削除します。

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

2. 以下の delete-network-acl コマンドを使用して、各ネットワーク ACL を削除します。

```
aws ec2 delete-network-acl --network-acl-id acl-id
```

3. 以下の delete-subnet コマンドを使用して、各サブネットを削除します。

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

4. 以下の delete-route-table コマンドを使用して、各カスタムルートテーブルを削除します。

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

5. 以下の detach-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 からデタッチします。

```
aws ec2 detach-internet-gateway --internet-gateway-id igw-id --vpc-id vpc-id
```

6. 以下の delete-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削除します。

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

7. [デュアルスタック VPC] 以下の delete-egress-only-internet-gateway コマンドを使用して、エグレス専用インターネットゲートウェイを削除します。

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

8. 以下の delete-vpc コマンドを使用して、VPC を削除します。

```
aws ec2 delete-vpc --vpc-id vpc-id
```

Console-to-Code を使用して VPC コンソールアクションから Infrastructure-as-Code を生成する

コンソールには、リソースの作成とプロトタイプのテストを行うためのガイド付きのパスが用意されています。同じリソースを大規模に作成したい場合は、自動化コードが必要です。Console-to-Code は Amazon Q Developer の機能であり、オートメーションコードの使用を開始するために役立ちま

す。Console-to-Code は、デフォルト値や互換性のあるパラメータを含むコンソールのアクションを記録します。次に、生成 AI を使用して、必要なアクションに応じて、推奨される Infrastructure as Code (IaC) 形式のコードを提案します。コンソールワークフローは指定されたパラメータ値の併用が有効であることを確実にするため、Console-to-Code を使用して生成されたコードには、互換性のあるパラメータ値があります。このコードを出発点として使用し、特定のユースケースの本番環境に対応するようにカスタマイズできます。

例えば、Console-to-Code では、ユーザーが VPC コンソールを使用してサブネット、セキュリティグループ、NACL、カスタムルーティングテーブル、およびインターネットゲートウェイを作成し、CloudFormation JSON 形式でコードを生成する方法を記録できます。その後、そのコードをコピーして、CloudFormation テンプレートでの使用向けにカスタマイズできます。

現在、Console-to-Code は次の言語と形式で Infrastructure as Code (IaC) を生成できます。

- CDK Java
- CDK Python
- CDK TypeScript
- CloudFormation JSON
- CloudFormation YAML

Console-to-Code の使用方法に関する詳細と手順については、「Amazon Q Developer User Guide」の「Automating AWS services with Amazon Q Developer Console-to-Code」を参照してください。

VPC のサブネット

サブネットは、VPC の IP アドレスの範囲です。特定のサブネットには、EC2 インスタンスなどの AWS リソースを作成できます。

内容

- サブネットの基本
- サブネットのセキュリティ
- サブネットの作成
- サブネットからの IPv6 CIDR ブロックを追加または削除する
- サブネットの IP アドレス指定属性を変更する
- サブネット CIDR 予約
- ルートテーブルを設定する
- ミドルボックスルーティングウィザード
- サブネットを削除する

サブネットの基本

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、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 AWS リソースを起動することにより、1 つのアベイラビリティーゾーンで発生した障害からアプリケーションを保護できます。

内容

- サブネット IP アドレス範囲
- サブネットタイプ
- サブネットの図表
- サブネットのルーティング
- サブネットの設定

サブネット IP アドレス範囲

サブネットを作成するときは、VPC の設定に応じて次のように IP アドレスを指定します。

- IPv4 のみ – サブネットには IPv4 CIDR ブロックがありますが、IPv6 CIDR ブロックはありません。IPv4 のみのサブネット内のリソースは IPv4 経由で通信する必要があります。
- デュアルスタック – サブネットには IPv4 CIDR ブロックと IPv6 CIDR ブロックの両方があります。VPC には、IPv4 CIDR ブロックと IPv6 CIDR ブロックの両方があります。デュアルスタックのサブネット内のリソースは、IPv4 および IPv6 経由で通信できます。
- IPv6 のみ – サブネットには IPv6 CIDR ブロックがありますが、IPv4 CIDR ブロックはありません。VPC には IPv6 CIDR ブロックが必要です。IPv6 のみのサブネット内のリソースは IPv6 経由で通信する必要があります。

> **Note**
>
> IPv6 専用サブネット内のリソースには、CIDR ブロック 169.254.0.0/16 から IPv4 リンクローカルアドレスが割り当てられます。これらのアドレスは、VPC でのみ利用可能なサービスと通信するために使用されます。例については、Amazon EC2 ユーザーガイドの「リンクローカルアドレス」を参照してください。

詳細については、「VPC とサブネットの IP アドレス指定」を参照してください。

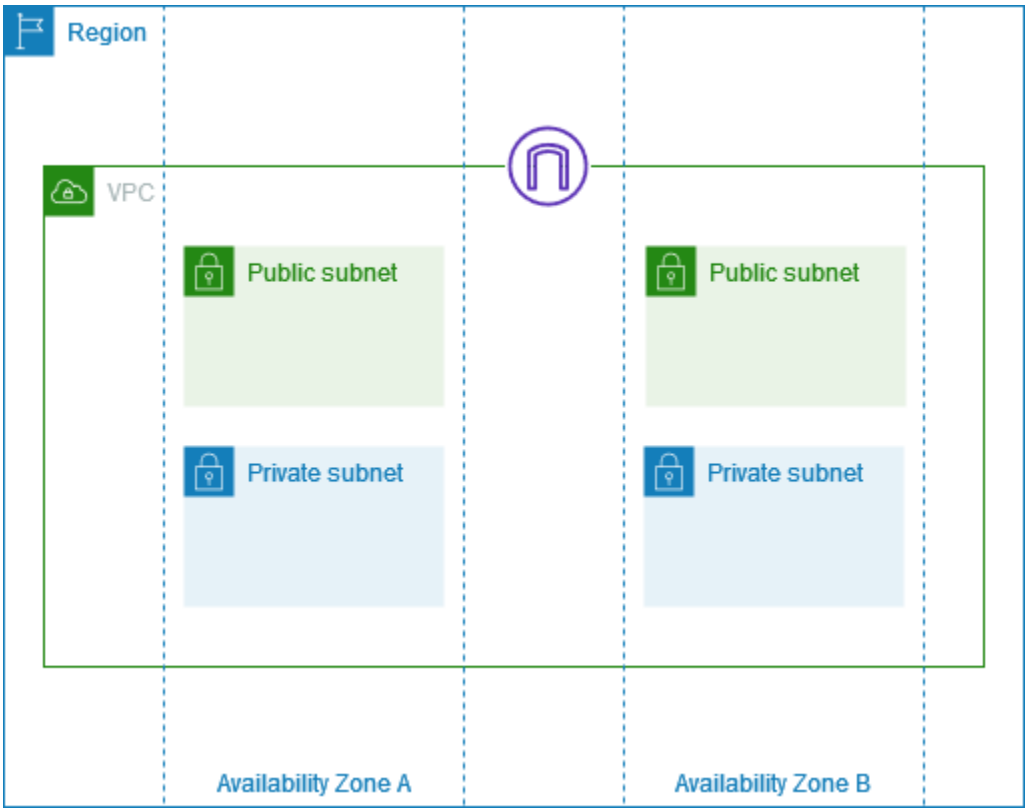
サブネットタイプ

サブネットのタイプは、サブネットのルーティングの設定方法により決まります。例えば、次のようになります。

- パブリックサブネット – サブネットには、インターネットゲートウェイへの直接ルートがあります。パブリックサブネット内のリソースは、パブリックインターネットにアクセスできます。
- プライベートサブネット – サブネットには、インターネットゲートウェイへの直接ルートがありません。プライベートサブネット内のリソースには、パブリックインターネットへのアクセス用に NAT デバイスが必要です。
- VPN のみのサブネット – サブネットには、仮想プライベートゲートウェイを介した Site-to-Site VPN 接続へのルートがあります。サブネットには、インターネットゲートウェイへのルートがありません。
- 隔離されたサブネット — このサブネットには、その VPC の外にある宛先へのルートがありません。隔離されたサブネット内のリソースは、同じ VPC 内にある他のリソースとの間でのみアクセスし、アクセスされることが可能です。
- EVS サブネット – このタイプのサブネットは Amazon EVS を使用して作成されます。詳細については、「Amazon EVS User Guide」の「VLAN subnet」を参照してください。

サブネットの図表

次の図は、2 つのアベイラビリティーゾーン、1 つのインターネットゲートウェイのサブネットを使用する VPC を示しています。各アベイラビリティーゾーンには、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。



ローカルゾーンと Wavelength Zone のサブネットを示す図については、「AWS ローカルゾーンの仕組み」と「AWS Wavelength の仕組み」を参照してください。

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、「ルートテーブルを設定する」を参照してください。

サブネットの設定

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 (例えば、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。

サブネットを作成したら、サブネットの次の設定を変更できます。

- IP 設定の自動割り当て: このサブネットの新しいネットワークインターフェイスのパブリック IPv4 または IPv6 アドレスを自動的にリクエストするように、IP 設定の自動割り当てを設定できます。
- リソースベース名 (RBN) の設定: このサブネット内の EC2 インスタンスのホスト名タイプを指定し、DNS A および AAAA レコードクエリの処理方法を設定できます。詳細については、「Amazon EC2 ユーザーガイド」の「Amazon EC2 インスタンスホスト名のタイプ」を参照してください。

サブネットのセキュリティ

AWS リソースを保護するために、プライベートサブネットを使用することをお勧めします。プライベートサブネット内にあるリソース (EC2 インスタンスなど) へのインターネットアクセスを許可するには、踏み台ホストまたは NAT デバイスを使用します。

AWS では、機能を使用して、VPC リソース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、EC2 インスタンスなど、関連付けられたリソー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制御します。ネットワーク ACL を使用して、サブネットレベルでインバウンドトラフィックとアウトバウンドトラフィックを許可または拒否します。ほとんどの場合、セキュリティグループはニーズを満たすことができます。ただし、セキュリティの追加のレイヤーが必要な場合は、ネットワーク ACL を使用できます。詳細については、「the section called “セキュリティグループとネットワーク ACL を比較する”」を参照してください。

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。デフォルトのネットワーク ACL では、すべてのインバウンドトラフィックとアウトバウンドトラフィックを許可します。デフォルトのネットワーク ACL を更新したり、カスタムネットワーク ACL を作成して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、「ネットワークアクセスコントロールリストを使用して、サブネットのトラフィックを制御する」を参照してください。

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。(詳しくは、「VPC フローログを使用した IP トラフィックのログ記録」を参照してください。)

サブネットの作成

次のステップを使用して、仮想プライベートクラウド (VPC) のサブネットを作成します。必要な接続によっては、ゲートウェイとルートテーブルの追加も必要になる場合があります。

考慮事項

- VPC の範囲のサブネットに IPv4 CIDR ブロックを指定する必要があります。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。詳細については、「VPC とサブネットの IP アドレス指定」を参照してください。
- IPv6 のみのサブネットを作成する場合は、次の点に注意してください。IPv6 のみのサブネットで起動される EC2 インスタンスは、IPv6 アドレスを受信しますが、IPv4 アドレスは受信しません。IPv6 のみのサブネットで起動するインスタンスは、Nitro システム上に構築されたインスタンスである必要があります。
- ローカルゾーンまたは Wavelength Zone 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。詳細については、「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。

サブネットを VPC に追加するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[Subnets (サブネット)] を選択します。
3. [サブネットの作成] を選択します。
4. [VPC ID] で サブネットの VPC を選択します。
5. (オプション) [Subnet name] (サブネット名) に、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。
6. [Availability Zone] (アベイラビリティーゾーン) で、サブネットのゾーンを選択するか、デフォルトの [No Preference] (設定なし) のままにして AWS が選択できるようにします。
7. IPv4 CIDR ブロックの場合は、[手動入力] を選択してサブネットの IPv4 CIDR ブロック (例: `10.0.1.0/24`) を入力するか、[IPv4 CIDR なし] を選択します。Amazon VPC IP Address Manager (IPAM) を使用して AWS ワークロードの IP アドレスを計画、追跡、モニタリングしている場合、サブネットを作成すると、IPAM (IPAM Address Manager) から CIDR ブロックを割

り当てることができます (IPAM Address Manager)。サブネット IP 割り当て用の VPC IP アドレス空間の計画の詳細については、「Amazon VPC IPAM ユーザーガイド」の「チュートリアル: サブネット IP 割り当て用の VPC IP アドレス空間の計画」を参照してください。

8. IPv6 CIDR ブロックの場合は、[手動入力] を選択して、サブネットを作成する VPC の IPv6 CIDR を選択します。このオプションは、VPC に IPv6 CIDR ブロックが関連付けられている場合にのみ使用できます。Amazon VPC IP Address Manager (IPAM) を使用して AWS ワークロードの IP アドレスを計画、追跡、モニタリングしている場合、サブネットを作成すると、IPAM から CIDR ブロックを割り当てる (IPAM-allocated) オプションを利用できます。サブネット IP 割り当て用の VPC IP アドレス空間の計画の詳細については、「Amazon VPC IPAM ユーザーガイド」の「チュートリアル: サブネット IP 割り当て用の VPC IP アドレス空間の計画」を参照してください。
9. IPv6 VPC CIDR ブロックを選択します。
10. IPv6 サブネット CIDR ブロックでは、VPC CIDR と同じかそれよりも具体的なサブネットの CIDR を選択します。例えば、VPC プール CIDR が /50 の場合、サブネットのネットマスク長は /50 から /64 の間で選択できます。IPv6 のネットマスク長は /44 から /64 の間で、/4 刻みです。
11. [サブネットの作成] を選択します。

AWS CLI を使用してサブネットを VPC に追加するには

create-subnet コマンドを使用します。

次のステップ

サブネットを作成したら、次のように設定できます。

- ルーティングを設定します。その後、インターネットゲートウェイなど、VPC に関連付けられているゲートウェイにトラフィックを送信するカスタムルートテーブルおよびルートを作成できます。詳細については、「ルートテーブルを設定する」を参照してください。
- IP アドレス設定動作を変更します。サブネットで起動されたインスタンスがパブリック IPv4 アドレス、IPv6 アドレス、またはその両方を受け取るかどうかを指定できます。詳細については、「サブネットの IP アドレス指定属性を変更する」を参照してください。
- リソースベース名 (RBN) の設定を変更します。詳細については、Amazon EC2 インスタンスホスト名タイプを参照してください。
- 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、「ネットワークアクセスコントロールリストを使用して、サブネットのトラフィックを制御する」を参照してください。
- 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。

サブネットからの IPv6 CIDR ブロックを追加または削除する

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。

サブネットで IPv6 が不要になっても、IPv4 リソースを作成して通信するためにサブネットを引き続き使用する場合は、IPv6 CIDR ブロックを削除できます。

IPv6 CIDR ブロックを削除する前に、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。

サブネットに IPv6 CIDR ブロックを追加するまたはこれを削除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[サブネット] を選択してください。
3. サブネットを選択し、[Actions] (アクション)、[Edit IPv6 CIDRs] (IPv6 CIDR の編集) の順に選択します。
4. CIDR を追加するには、[IPv6 CIDR を追加] を選択し、[VPC CIDR ブロック] を選択し、[サブネットの CIDR ブロック] に入力して、VPC CIDR のネットマスク長と同等かそれよりも詳細なネットマスク長を選択します。例えば、VPC プール CIDR が /50 の場合、サブネットのネットマスク長は /50 から /64 の間で選択できます。IPv6 のネットマスク長は /44 から /64 の間で、/4 刻みです。
5. CIDR を削除するには、IPv6 CIDR ブロックを見つけて [削除] を選択します。
6. [保存] を選択します。

AWS CLI 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 コマンドを使用します。

AWS CLI 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 コマンドを使用します。

サブネットの IP アドレス指定属性を変更する

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。

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 IP アドレス指定の動作を変更するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[サブネット] を選択してください。
3. サブネットを選択してから、[Actions] (アクション)、[Edit subnet settings] (サブネット設定の編集) の順に選択します。
4. [パブリック IPv4 アドレスの自動割り当てを有効化] 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、[保存] を選択します。
5. [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、[保存] を選択します。

AWS CLI を使用してサブネット属性を変更するには

modify-subnet-attribute コマンドを使用します。

サブネット CIDR 予約

サブネット CIDR 予約は、AWS がネットワークインターフェイスに割り当てないように設定された IPv4 アドレスまたは IPv6 アドレスの範囲で行われます。これは、ネットワークインターフェイスで使用する IPv4 または IPv6 CIDR ブロック (「プレフィックス」とも呼ばれます) を予約することを可能にします。

サブネット CIDR 予約を作成する際、予約された IP アドレス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 — 単一のネットワークインターフェイスにプレフィックスを割り当てることができます。詳細は、「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クスの割り当て」を参照してください。
- 明示的 — 個々の IP アドレスを単一のネットワークインターフェイスに手動で割り当てることができます。

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 CIDR 予約を作成する際、既に使用されているアドレスを IP アドレスの範囲に含めることができます。サブネット予約を作成しても、既に使用されている IP アドレスの割り当てが解除されることはありません。
- サブネットごとに複数の CIDR 範囲を予約できます。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。
- サブネット予約を削除すると、AWS による未使用の IP アドレスのネットワークインターフェイスへの割り当てが可能になります。サブネット予約を削除しても、使用中の IP アドレスの割り当てが解除されることはありません。
- 予約タイプは、サブネットで利用可能な IP アドレスの数に影響します。プレフィックス予約を作成すると、数はすぐに減少します。明示的なプレフィックス予約を作成すると、数は IP アドレスが割り当てられたときに減少します。

Classless Inter-Domain Routing (CIDR) 表記の詳細については、「IP アドレス指定」を参照してください。

内容

- コンソールを使用してサブネット CIDR 予約を操作する
- AWS CLI を使用してサブネット CIDR 予約を操作する

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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[サブネット] を選択してください。
3. サブネットを選択します。
4. [CIDR の予約] タブを選択し、既存のサブネット CIDR の予約に関する情報を取得します。
5. サブネット CIDR の予約を追加または削除するには、[アクション]、[CIDR 予約を編集] の順に選択してから、次の操作を行います。
   - 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 の予約を削除するには、そのサブネット CIDR 予約の [削除] を選択します。

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
- サブネット CIDR 予約の表示
- サブネット CIDR 予約の削除

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
```

以下は出力の例です。

```
{
    "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
```

ルートテーブルを設定する

ルートテーブルサーバーは、仮想プライベートクラウド (VPC) のトラフィックコントローラーとして機能します。各ルートテーブルには、ルートと呼ばれる一連のルールが含まれています。ルートは、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決定します。VPC を作成するときは、VPC のメインルートテーブルも作成されます。VPC のネットワークパスをより細かく制御できるように、VPC 用の追加のルートテーブルを作成できます。

ルートテーブルを使用して、VPC が通信できるネットワーク (他の VPC やオンプレミスネットワークなど) を指定できます。各ルートは、送信先 (CIDR ブロックまたはプレフィックスリスト) とターゲット (インターネットゲートウェイ、NAT ゲートウェイ、VPC ピアリング接続、VPN 接続など)

を指定します。トラフィックは、送信先 IP アドレスに基づいてターゲットにルーティングされます。ルートテーブルを使用すると、パブリックサブネット、プライベートサブネット、VPN 専用サブネット、隔離されたサブネットを含めた複雑なネットワークアーキテクチャを作成できます。

内容

- ルートテーブルの概念
- サブネットのルートテーブル
- ゲートウェイルートテーブル
- ルーティングの優先度の仕組み
- ルーティングオプションの例
- VPC のカスタムルートテーブルを作成する
- サブネットルートテーブルを管理する
- メインルートテーブルの置換
- ゲートウェイルートテーブルを使用して VPC に進入するトラフィックを制御する
- ローカルルートのターゲットを置換または復元する
- VPC 内での高度なルーティング
- VPC 内の到達可能性問題をトラブルシューティングする

ルートテーブルの概念

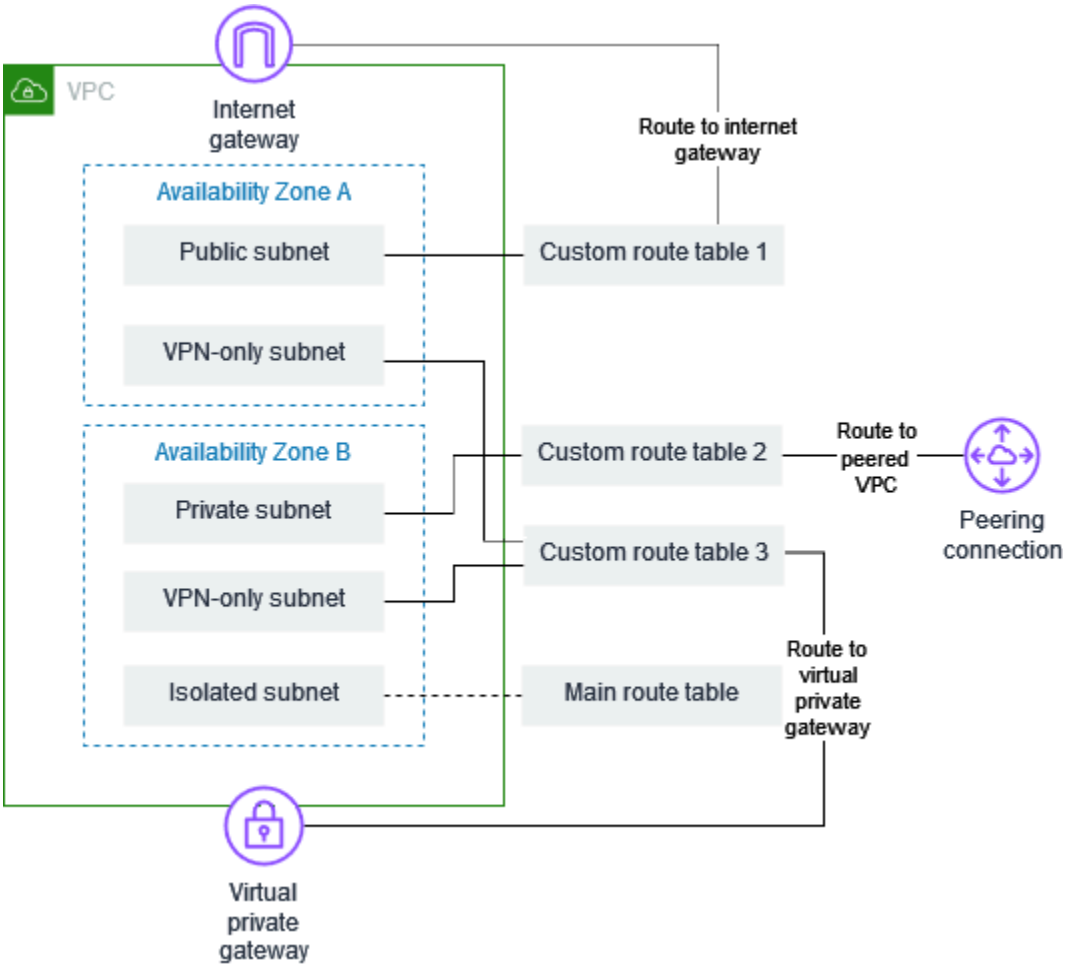
ルートテーブルの主な概念は次のとおりです。

- 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。
- [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。
- [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。VPC に IPv4 アドレスと IPv6 アドレスの両方がある場合は、IPv4 用のローカルルートと IPv6 用のローカルルートがあります。
- 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。
- 伝達 – VPC に仮想プライベートゲートウェイをアタッチし、ルート伝達を有効にすると、サブネットルートテーブルへの VPN 接続のルートが自動的に追加されます。これは、VPN ルートを手動で追加または削除する必要がないことを意味します。詳細については、「Site-to-Site VPN ユーザーガイド」の「Site-to-Site VPN のルーティングオプション」を参照してください。
- 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。
- 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。
- Transit Gateway ルートテーブル — Transit Gateway に関連付けられているルートテーブル。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。
- 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。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。

ルートテーブルを使用する VPC の例

以下の図は、5 つのサブネット、メインルートテーブル、3 つのカスタムルートテーブルが設定された VPC を示しています。4 つのルートテーブルすべてにローカルルートがあります。カスタムルートテーブル 1 にはインターネットゲートウェイへのルートがあり、アベイラビリティーゾーン A 内のパブリックサブネットに関連付けられています。カスタムルートテーブル 2 にはピアリングされた VPC へのルートがあり、アベイラビリティーゾーン B 内のプライベートサブネットに関連付けられています。カスタムルートテーブル 3 には仮想プライベートゲートウェイへのルートがあり、両方のアベイラビリティーゾーン内の VPN 専用サブネットに関連付けられています。



サブネットのルートテーブル

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。

内容

- ルート
- メインルートテーブル
- カスタムルートテーブル
- サブネットとルートテーブルの関連付け

ルート

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。

ルーティング先	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。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。

ルールと考慮事項

- 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。ターゲットは、NAT ゲートウェイ、ネットワークインターフェイス、Gateway Load Balancer エンドポイントである必要があります。
- ルートテーブルに複数のルートがある場合、トラフィックと一致する (最長プレフィックス一致) 最も明確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。
- 完全一致、または次の範囲のサブセットである IPv4 アドレスにルートを追加することはできません: 169.254.168.0/22。この範囲はリンクローカルアドレススペース内にあり、AWS のサービスで使用するために予約されています。例えば、Amazon EC2 は、インスタンスメタデータサービス (IMDS) や Amazon DNS サーバーなどの EC2 インスタンスからのみアクセスできるサービスに、この範囲のアドレスを使用します。より大きいが 169.254.168.0/22 と重複する CIDR ブロックを使用できますが、169.254.168.0/22 のアドレスを送信先とするパケットは転送されません。

- 完全一致、または次の範囲のサブセットである IPv6 アドレスにルートを追加することはできません: fd00:ec2::/32。この範囲は一意のローカルアドレス (ULA) スペース内にあり、AWS のサービスで使用するために予約されています。例えば、Amazon EC2 は、インスタンスメタデータサービス (IMDS) や Amazon DNS サーバーなどの EC2 インスタンスからのみアクセスできるサービスに、この範囲のアドレスを使用します。より大きいが fd00:ec2::/32 と重複する CIDR ブロックを使用できますが、fd00:ec2::/32 のアドレスを宛先とするパケットは転送されません。
- 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。詳細については、「the section called “ミドルボックスアプライアンスのルーティング”」を参照してください。

例

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。IPv4 トラフィックと IPv6 トラフィックは、次のルートテーブルに示すように別々に扱われます。

ルーティング先	ターゲット
10.0.0.0/16	Local
2001:db8:1234:1a00::/56	Local
172.31.0.0/16	pcx-11223344556677889
0.0.0.0/0	igw-12345678901234567
::/0	eigw-aabbccddee1122334

- VPC (10.0.0.0/16) 内でルーティングされる IPv4 トラフィックは Local ルートの対象となります。
- VPC 内でルーティングされる IPv6 トラフィック (2001:db8:1234:1a00::/56) は Local ルートの対象となります。
- 172.31.0.0/16 のルートは、トラフィックをピアリング接続に送信します。
- すべての IPv4 トラフィック (0.0.0.0/0) のルートは、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。そのため、VPC 内のトラフィックとピアリング接続へのトラフィックを除くすべての IPv4 トラフィックは、インターネットゲートウェイにルーティングされます。
- すべての IPv6 トラフィックのルート (::/0) は、トラフィックを Egress-Only インターネットゲートウェイに送信します。そのため、VPC 内のトラフィックを除く IPv6 トラフィックはすべて、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。

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。VPC を作成するNAT ゲートウェイを選択すると、Amazon VPC はゲートウェイのメインルートテーブルにルートを自動的に追加します。

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で、ルートを追加、削除、変更することができます。
- メインルートテーブルを削除することはできません。
- 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を置き換えるには、カスタムルートテーブルをサブネットに関連付けます。
- 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、「メインルートテーブルの置換」を参照してください

カスタムルートテーブル

デフォルトでは、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。パブリックサブネットを VPC を作成する および選択すると、Amazon VPC 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を指す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。

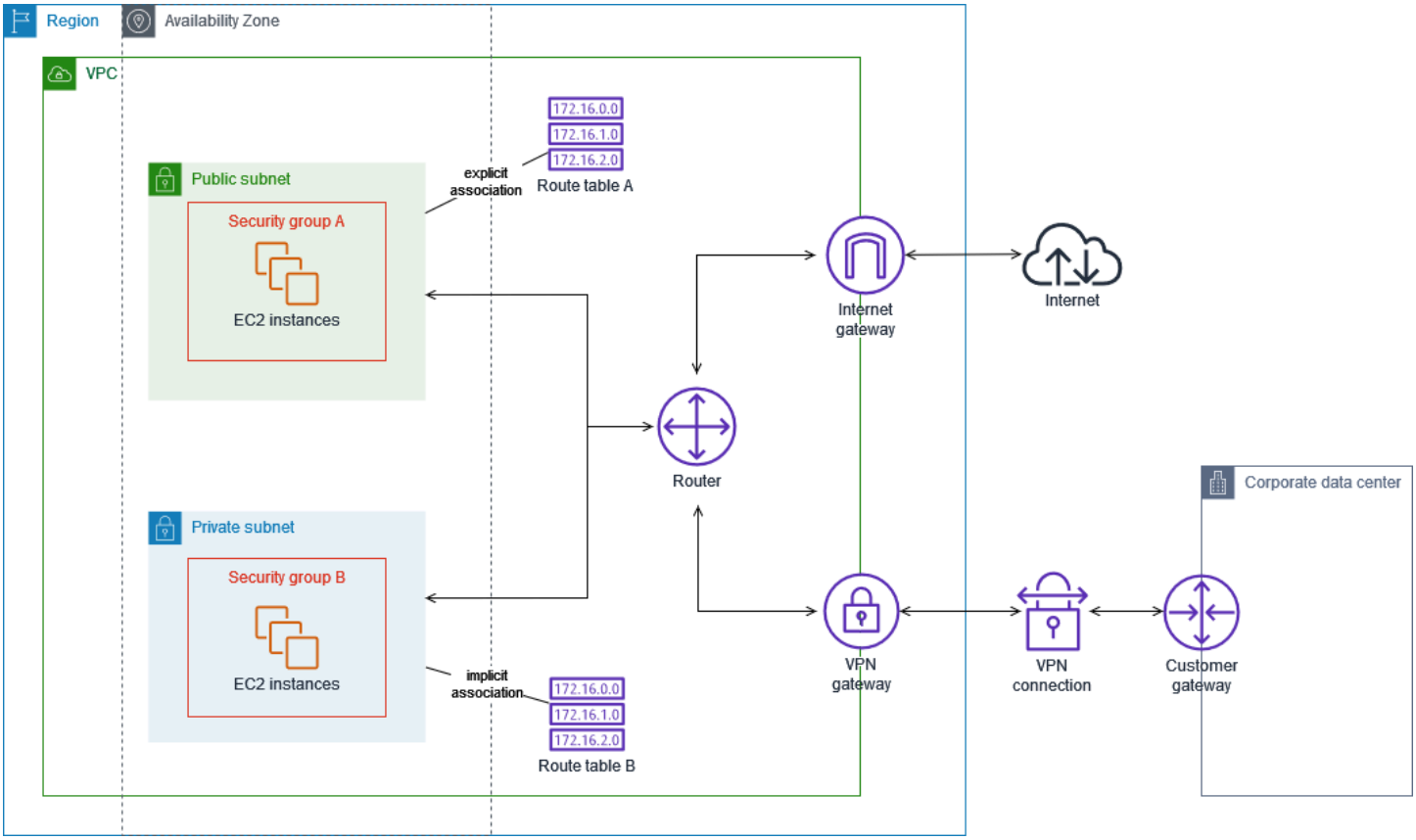
サブネットとルートテーブルの関連付け

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、「明示的な関連付けを判断する」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。



ルートテーブル A はカスタムルートテーブルで、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。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送信するルートがあり、それによりサブネットはパブリックサブネットになります。

ルーティング先	Target
VPC CIDR	ローカル
0.0.0.0/0	*igw-id*

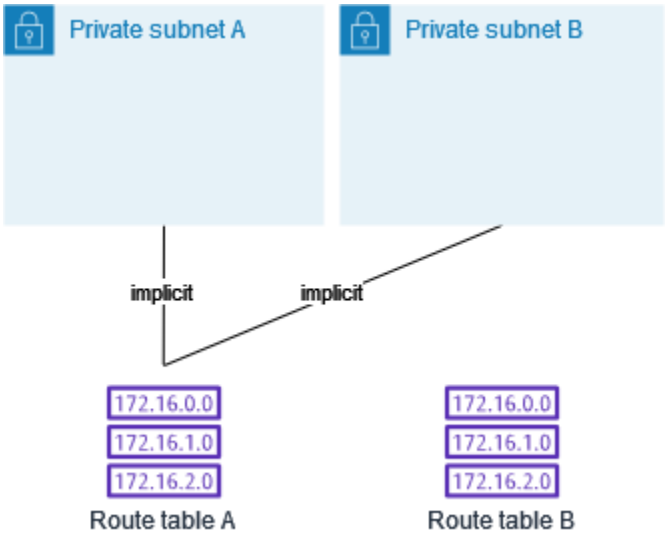
ルートテーブル B は、メインルートテーブルです。プライベートサブネットに暗黙的に関連付けられています。すべてのトラフィックを仮想プライベートゲートウェイに送信するルートがありますが、インターネットゲートウェイには送信しないため、サブネットは VPN のみのサブネットになります。この VPC に別のサブネットを作成し、カスタムルートテーブルを関連付けない場合、そのサブネットはメインルートテーブルであるため、このルートテーブルにも暗黙的に関連付けられます。

ルーティング先	Target
VPC CIDR	ローカル
0.0.0.0/0	*vgw-id*

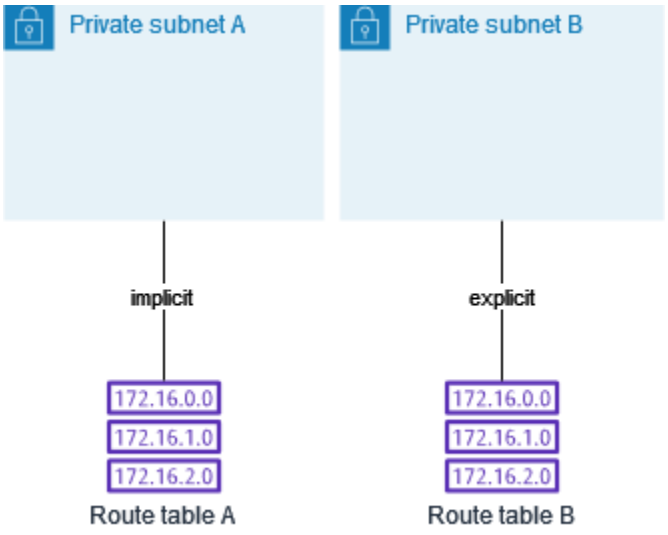
例 2: メインルートテーブルを置き換える

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。

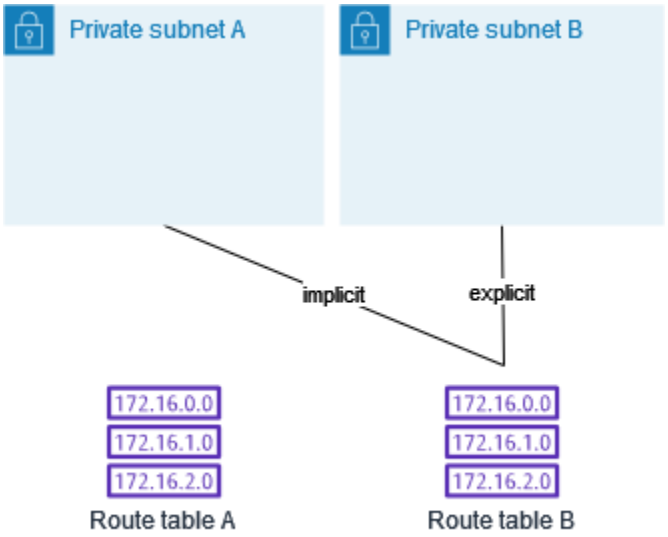
次の図は、2 つのサブネットと 2 つのルートテーブルを示しています。サブネット A は、メインルートテーブルであるルートテーブル A に暗黙的に関連付けられています。サブネット B はルートテーブル A に暗黙的に関連付けられています。カスタムルートテーブルであるルートテーブル B は、どちらのサブネットにも関連付けられていません。



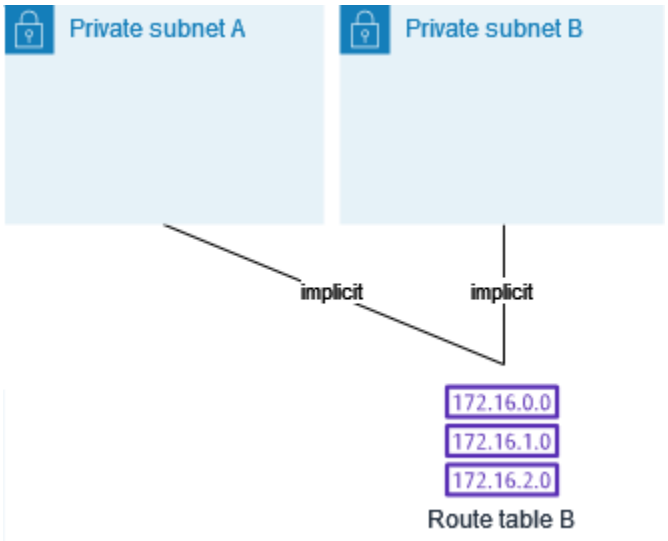
メインルートテーブルを置き換えるには、まずサブネット B とルートテーブル B の間に明示的な関連付けを作成します。ルートテーブル B をテストします。



ルートテーブル B をテスト後、そのテーブルをメインルートテーブルにします。サブネット B とルートテーブル B との間には、まだ明示的な関連付けがあります。ただし、ルートテーブル B が新しいメインルートテーブルであるため、サブネット A とルートテーブル B の間には暗黙的な関連付けができます。ルートテーブル A はいずれのサブネットにも関連付けられていない状態となりました。



(オプション) サブネット B 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 B とルートテーブル B との間の暗黙的な関連付けは残ります。ルートテーブル A が必要なくなった場合は削除できます。



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。

内容

- ゲートウェイルートテーブルルート
- ルールと考慮事項

ゲートウェイルートテーブルルート

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
- 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

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、「ローカルルートのターゲットを置換または復元する」を参照してください。

例

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	ターゲット
172.31.0.0/16	ローカル
172.31.0.0/20	*eni-id*

例

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	ターゲット
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”」を参照してください。

- 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。
- すべてのLocal Zones が仮想プライベートゲートウェイとのエッジ関連付けをサポートしているわけではありません。使用可能なゾーンの詳細については、「AWSLocal Zones ユーザーガイド」の「考慮事項」を参照してください。

ルーティングの優先度の仕組み

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。

次のリストは、以下のセクションへのリンクを含むルート優先度の概要と、より詳細な情報および例を示しています。

1. 最長プレフィックス (例: 10.10.2.15/32 は 10.10.2.0/24 よりも優先されます)

2. 静的ルート (VPC ピアリングやインターネットゲートウェイ接続など)
3. プレフィックスリストルート
4. 伝播されたルート
   a. Direct Connect BGP ルート (動的ルート)
   b. VPN 静的ルート
   c. VPN BGP ルート (動的ルート) (仮想プライベートゲートウェイなど)

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

静的および動的に伝播されたルートのルート優先度

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静的ルートの送信先と同じ場合、静的ルートが優先されます。次のリソースは静的ルートを使用します。

- 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 ピアリング接続
- 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。

ルーティング先	Target	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

プレフィックスリストのルーティング優先度

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。

- 伝播された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参照するルートと一致する場合は、プレフィックスリストを参照するルートが優先されます。重複するルートについては、伝播されたルート、静的ルート、またはプレフィクスリストを参照するルートであるかどうかにかかわらず、より具体的なルートが常に優先されることに注意してください。

- 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。

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。

内容

- インターネットゲートウェイへのルーティング
- NAT デバイスへのルーティング
- 仮想プライベートゲートウェイへのルーティング
- AWS Outposts ローカルゲートウェイへのルーティング
- VPC ピアリング接続へのルーティング
- ゲートウェイ VPC エンドポイントへのルーティング
- Egress-Only インターネットゲートウェイへのルーティング
- トランジットゲートウェイのルーティング
- ミドルボックスアプライアンスのルーティング
- プレフィックスリストを使用したルーティング
- Gateway Load Balancer エンドポイントにルーティングする

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、「インターネットゲートウェイを使用して VPC のインターネットアクセスを有効にする」を参照してください

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、特定のリージョンにおける Amazon S3 の IP アドレス範囲を含むプレフィックスリスト) 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 ピアリング接続にルーティングされます。これらの 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 ピアリング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 デバイス」を参照してください。

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 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、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

このセクションでは、AWS Outposts ローカルゲートウェイにルーティングするためのルーティングテーブル設定について説明します。

内容

- Outpost サブネットとオンプレミスネットワーク間のトラフィックを有効にする
- Outposts 全体で同じ VPC 内のサブネット間のトラフィックを有効にする

Outpost サブネットとオンプレミスネットワーク間のトラフィックを有効にする

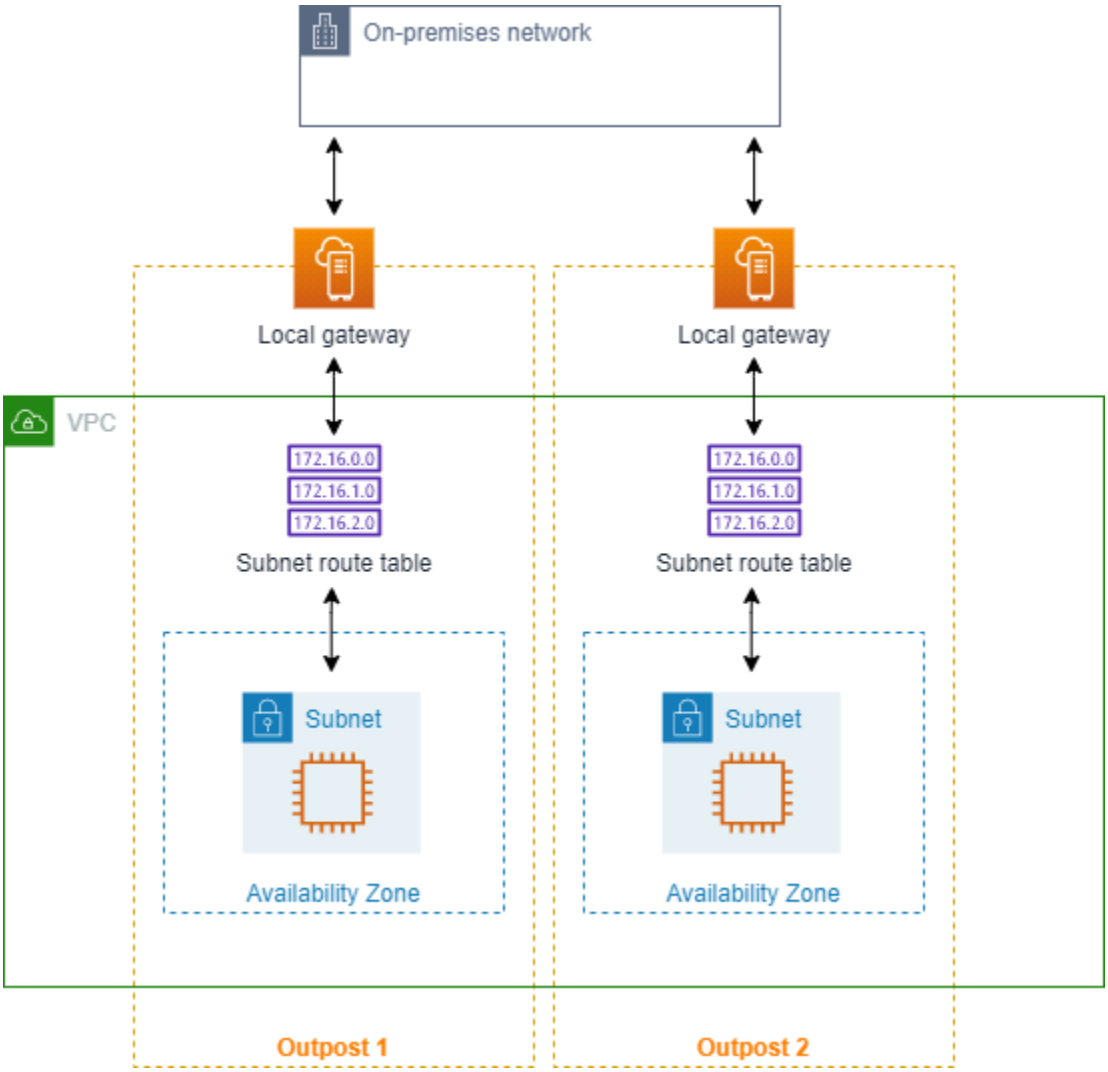
AWS Outposts に関連付けられた VPC 内のサブネットには、ローカルゲートウェイの追加ターゲットタイプを設定できます。送信先アドレス 192.168.10.0/24 のトラフィックをローカルゲートウェイでカスタマーネットワークにルーティングする場合を考えます。これを行うには、送信先ネットワークとローカルゲートウェイ (lgw-xxxx) のターゲットで次のルートを追加します。

送信先	ターゲット
192.168.10.0/24	*lgw-id*

Outposts 全体で同じ VPC 内のサブネット間のトラフィックを有効にする

Outpost のローカルゲートウェイとオンプレミスネットワークを使用して、異なる Outposts 全体で同じ VPC 内のサブネット間の通信を確立できます。

この機能を使用して、異なる AZ に固定されている Outposts ラック間の接続を確立することにより、Outposts ラック上で実行されるオンプレミスアプリケーションのマルチアベイラビリティーゾーン (AZ) アーキテクチャと同様のアーキテクチャを構築できます。



この機能を有効にするには、Outpost ラックのサブネットルートテーブルに、そのルートテーブル内のローカルルートよりも具体的で、ターゲットタイプがローカルゲートウェイであるルートを追加します。ルートの送信先は、別の Outpost にある VPC 内のサブネットの IPv4 ブロック全体と一致させる必要があります。通信が必要なすべての Outpost サブネットに対して、この設定を繰り返します。

Important

- この機能を使用するには、ダイレクト VPC ルーティングを使用する必要があります。自分のカスタマー所有 IP アドレスは使用できません。
- Outposts ローカルゲートウェイが接続されているオンプレミスネットワークには、サブネットが相互にアクセスするためのルーティングが必要です。
- サブネット内のリソースにセキュリティグループを使用する場合は、Outpost サブネットの送信元または送信先として、IP アドレス範囲を含むルールを使用する必要があります。セキュリティグループ ID は使用できません。
- 場合によっては、既存の Outposts ラックを更新して、複数の Outposts 間の VPC 内通信をサポートできるようにする必要があります。この機能を使用できない場合は、AWS サポートに連絡してください。

Example例

CIDR が 10.0.0.0/16 の VPC、CIDR が 10.0.1.0/24 の Outpost 1 サブネット、CIDR が 10.0.2.0/24 の Outpost 2 サブネットの場合、Outpost 1 サブネットのルートテーブルのエントリは次のようになります。

ルーティング先	ターゲット
10.0.0.0/16	ローカル
10.0.2.0/24	*lgw-1-id*

Outpost 2 サブネットのルートテーブルのエントリは次のようになります。

ルーティング先	ターゲット
10.0.0.0/16	ローカル
10.0.1.0/24	*lgw-2-id*

VPC ピアリング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 ピアリング接続にある 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 ピアリング接続を指す 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	ターゲット
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。

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 ブロックのすべてまたは一部にアクセスできるようにします。

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	ターゲット
172.31.0.0/16	ローカル
10.0.0.0/16	pcx-11223344556677889
2001:db8:1234:1a00::/56	pcx-11223344556677889

VPC ピアリング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。

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、「Egress-Only インターネットゲートウェイを使用してアウトバウンド IPv6 トラフィックを有効にする」を参照してください

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”」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 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、「ゲートウェイルートテーブル」を参照してください。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、「ミドルボックスシナリオ」を参照してください。

内容

- アプライアンスに関する考慮事項
- 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
- 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

アプライアンスに関する考慮事項

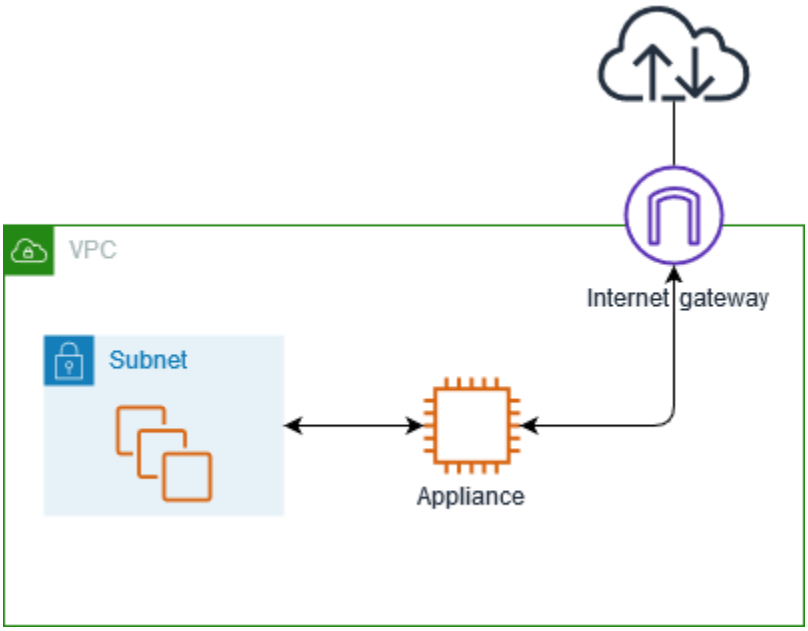
AWS Marketplace からサードパーティー製アプライアンスを選択することも、独自のアプライアンスを設定することもできます。アプライアンスを作成または設定するときは、次の点に注意してください。

- 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、「Amazon EC2 ユーザーガイド」の「Changing the Source or Destination Checking」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場合、VPC、サブネット、アプライアンスが IPv6 をサポートしていることを確認します。
-

> **Important**
>
> トランジットゲートウェイ、NAT ゲートウェイ、Network Load Balancer などの AWS マネージドサービスからのトラフィックをミドルボックスアプライアンス経由で、そのサービスがアタッチされているサブネットにルーティングしないでください。この設定は、予測不可能なルーティング動作を引き起こし、サポートされません。

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。

ルーティング先	Target
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。

ルーティング先	Target
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。

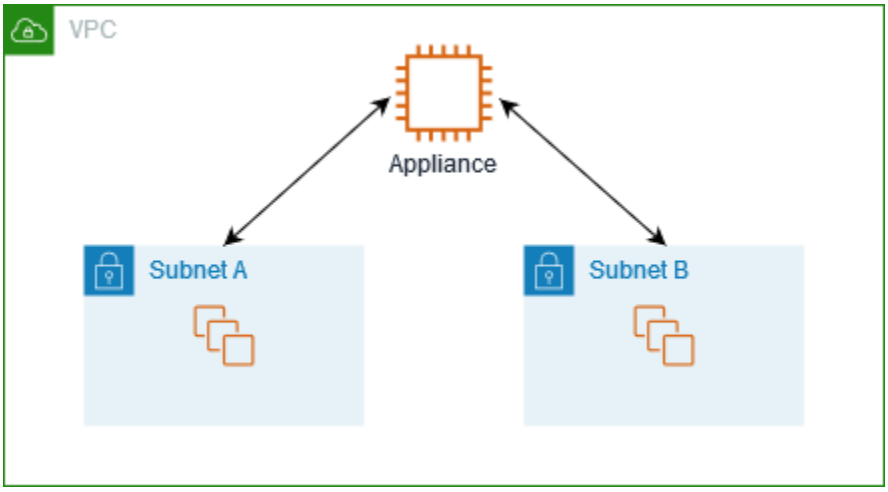
送信先	Target
VPC CIDR	ローカル
0.0.0.0/0	*###################### ID*

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。

ルーティング先	Target
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。

ルーティング

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。

ルーティング先	Target
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。

ルーティング先	Target
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。

ルーティング先	Target
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	ターゲット
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。Gateway Load Balancer を作成し、Gateway Load Balancer エンドポイントサービスを設定し、VPC に Gateway Load Balancer エンドポイントを作成してサービスに接続できます。

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 “セキュリティアプライアンスを使用してトラフィックを検査する”」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。

ルーティング先	Target
10.0.1.0/24	*vpc-endpoint-id*

Gateway Load Balancer エンドポイントをターゲットとして使用する場合、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。VPC エンドポイントをターゲットとするプレフィックスリストルートを作成または置換しようとすると、「Cannot create or replace a prefix list route targeting a VPC Endpoint」というエラーが表示されます。

詳細については、「Gateway Load Balancer の開始方法」をご参照ください。

VPC のカスタムルートテーブルを作成する

VPC のカスタムルートテーブルを作成して設定するには、次のタスクを実行します。デフォルトで、新しいルートテーブルには VPC 内での通信を可能にするローカルルートが含まれます。送信先 IP アドレスの範囲に基づいてネットワークトラフィックを特定のターゲットにルーティングするルートを追加できます。

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。

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。

> **Note**
>
> 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、「Amazon VPC クォータ」を参照してください。

タスク

- ルートテーブルを作成する
- ルートテーブルにルートを追加する
- サブネットをルートテーブルに関連付ける

ルートテーブルを作成する

コンソールを使用して ルートテーブルを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[Route tables] (ルートテーブル) 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) [Name] (名前) 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) タグを追加するには、[Add new tag] (新しいタグを追加) を選択し、タグキーとタグ値を入力します。
7. [ルートテーブルの作成] を選択します。

AWS CLI を使用してルートテーブルを作成するには

create-route-table コマンドを使用します。

ルートテーブルにルートを追加する

コンソールを使用してルートテーブルにルートを追加する

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [ルートテーブル] (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. [Add Rule (ルートの追加)] を選択します。
5. [送信先] に、次のいずれかを入力します。
   - IP アドレス範囲 – 192.168.0.0/16 など
   - シングル IP アドレス – 192.168.10.1/32 など

- プレフィックスリストの ID – pl-0abcdef1234567890 など

6. [ターゲット] にリソースタイプ (ネットワークインターフェイスなど) を選択してから、リソースの ID (eni-11223344556677889 など) を入力します。
7. [Save changes] (変更の保存) をクリックします。

AWS CLI 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 コマンドを使用します。

サブネットをルートテーブルに関連付ける

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 [ルートテーブル] (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ます。
5. [Save associations] (関連付けを保存する) を選択します。

AWS CLI を使用してサブネットをルートテーブルに関連付ける、または関連付けを解除するには

- associate-route-table
- disassociate-route-table

サブネットルートテーブルを管理する

ルートテーブルを使用して VPC ルーティングを管理するには、次の手順に従います。

タスク

- サブネット用のルートテーブルの決定
- 明示的な関連付けを判断する
- ルートを追加、変更、削除する

- ルート伝達は有効または無効にできます。
- サブネット用のルートテーブルの編集

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。

コンソールを使用してサブネットのルートテーブルを判断する

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[サブネット] を選択してください。
3. サブネットを選択します。
4. [ルートテーブル] タブを選択すると、ルートテーブルとそのルートに関する情報が表示されます。関連付けがメインルートテーブルとのものか、また、そ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、「明示的な関連付けを判断する」を参照してください。

明示的な関連付けを判断する

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」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] (ルートテーブル) を選択します。
3. [明示的なサブネットの関連付け] (Explicit subnet association) 列を確認して、明示的に関連付けられたサブネットを特定し、[メイン] (Main) 列を確認して、これがメインルートテーブルかどうかを判断します。
4. ルートテーブルを選択し、[サブネットの関連付け] (Subnet associations) タブを選択します。

5. [明示的なサブネットの関連付け] (Explicit subnet associations) の下のサブネットが、ルートテーブルに明示的に関連付けられています。[明示的な関連付けのないサブネット] (Subnets without explicit associations) の下のサブネットは、ルートテーブルと同じ VPC に属していますが、どのルートテーブルにも関連付けられていません。そのため、VPC のメインルートテーブルと暗黙的に関連付けられています。

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] (ルートテーブル) を選択します。
3. ルートテーブルを選択し、[Edge associations] (エッジの関連付け) タブを選択します。

AWS CLI 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 コマンドを使用します。

ルートを追加、変更、削除する

ルートテーブルのルートを追加、変更、削除できます。

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。

考慮事項

- 変更できるのは、追加したルートのみです。
- ルートを変更または削除すると、これらのルートを使用する既存の接続に影響します。他のルートを使用する接続は影響を受けません。
- ルートテーブルごとに追加できるルートの数にはクォータが設定されています。詳細については、「Amazon VPC クォータ」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のルートを更新するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [ルートテーブル] (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. 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] で、IP アドレス範囲、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。[ターゲット] で、リソースタイプを選択し、リソースの ID を入力します。
5. ルートを変更するには、新しい送信先 CIDR ブロックまたはプレフィックスリスト ID を入力し、ターゲットを選択します。
6. ルートを削除するには、[Remove] (削除) を選択します。
7. [Save changes] (変更の保存) をクリックします。

AWS CLI を使用してルートテーブルのルートを更新するには

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。

- create-route
- replace-route
- delete-route

ルート伝達は有効または無効にできます。

ルート伝達は、仮想プライベートゲートウェイがルートテーブルにルートを自動的に伝達できるようにします。これは、VPN ルートを手動で追加または削除する必要がないことを意味します。

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。

詳細については、「Site-to-Site VPN ユーザーガイド」の「Site-to-Site VPN routing options」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または無効にするには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [ルートテーブル] (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 [有効化] チェックボックスをオンまたはオフにします。

5. [保存] を選択します。

AWS CLI を使用してルート伝達を有効または無効にするには

- enable-vgw-route-propagation
- disable-vgw-route-propagation

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. [Route table] (ルートテーブル) タブから、[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. [Route table ID] (ルートテーブル ID) で、新しいルートテーブルを選択します。
5. [保存] を選択します。

AWS CLI 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 コマンドを使用します。

メインルートテーブルの置換

このセクションでは、VPC のメインルートテーブルであるルートテーブルを変更する方法について説明します。

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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [Route tables] (ルートテーブル) を選択してから、新しいメインルートテーブルを選択します。

3. [Actions] (アクション)、[Set main route table] (メインルートテーブルの設定) を順に選択します。
4. 確認を求められたら、「**set**」と入力してから、[OK] を選択します。

AWS CLI 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 コマンドを使用します。

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 [ルートテーブル] (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) タブから、[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ます。
5. [Save associations] (関連付けを保存する) を選択します。

ゲートウェイルートテーブルを使用して VPC に進入するトラフィックを制御する

VPC に進入するトラフィックをゲートウェイルートテーブルを使って制御するには、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をルートテーブルに関連付けるか、関連付けを解除します。詳細については、「ゲートウェイルートテーブル」を参照してください。

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または関連付けを解除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [ルートテーブル] (Route tables) を選択して、ルートテーブルを選択します。
3. [Edge associations] (エッジの関連付け) タブから、[Edit edge associations] (Edge の関連付けを編集) を選択します。

4. ゲートウェイのチェックボックスをオンまたはオフにします。
5. [Save changes] (変更の保存) をクリックします。

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。次の例は、指定されたルートテーブルを指定されたインターネットゲートウェイに関連付けます。

```
aws ec2 associate-route-table
    --route-table-id rtb-01234567890123456 \
    --gateway-id igw-11aa22bb33cc44dd1
```

AWS CLI を使用して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

disassociate-route-table コマンドを使用します。ルートテーブルとゲートウェイ間の関連付けの ID を指定します。

```
aws ec2 disassociate-route-table \
    --association-id rtbassoc-0abcdef1234567890
```

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を置き換える

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [ルートテーブル] (Route tables) を選択して、ルートテーブルを選択します。
3. [Routes] (ルート) タブから、[Edit routes] (ルートの編集) を選択します。
4. ローカルルートの場合は、[Target] (ターゲット) をオフにしてから、新しいターゲットを選択します。
5. [Save changes] (変更の保存) をクリックします。

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 [ルートテーブル] (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. ルートの場合は、[Target] (ターゲット) をオフにしてから、[Local] (ローカル) を選択します。
5. [Save changes] (変更の保存) をクリックします。

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。次の例は、ローカルルートのターゲットを指定されたネットワークインターフェースに置き換えます。

```
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

次の例は、指定されたルートテーブル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
```

VPC 内での高度なルーティング

VPC に高度なルーティングシナリオを設定します。このセクションでは、トラフィックフローを管理するための静的ルーティングアプローチと動的ルーティングアプローチの両方を説明します。

- 静的イングレスルーティング: BYOIP (Bring Your Own IP) アドレスプール宛てのインバウンドインターネットトラフィックが VPC 内の特定のネットワークインターフェイスに転送されるように静的ルートを設定します。

- VPC Route Server を使用した動的ルーティング: BGP ベースの動的ルーティングを使用して VPC とインターネットゲートウェイのルートテーブルを自動的に更新することで、ワークロードに耐障害性と自動フェイルオーバーを提供します。

内容

- インターネットトラフィックを単一のネットワークインターフェイスにルーティングする
- VPC Route Server を使用する VPC 内の動的ルーティング

インターネットトラフィックを単一のネットワークインターフェイスにルーティングする

パブリック IP アドレスの大規模なプール宛てのインバウンドインターネットトラフィックを、VPC 内にある単一の Elastic Network Interface (ENI) にルーティングできます。

これまで、インターネットゲートウェイが受け入れるのは、VPC 内のネットワークインターフェイスに直接関連付けられたパブリック IP アドレス宛てのトラフィックのみでした。インスタンスタイプに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る IP アドレス数の制限があるため、これらの制限を超える規模の IP プールのトラフィックを処理する必要がある電気通信やモノのインターネット (IoT) といった業界の課題となっていました。

このルーティングを使用すると、インバウンドインターネット接続での複雑なアドレス変換が不要になります。独自のパブリック IP プールを持ち込み (BYOIP)、プール全体のトラフィックを受け入れて単一のネットワークインターフェイスにルーティングするように VPC インターネットゲートウェイを設定できます。この機能は、次の場合に特に有用です。

- 電気通信: アドレス変換のオーバーヘッドを発生させることなく大規模なサブスクライバー IP プールを管理
- IoT アプリケーション: 何千ものデバイス IP アドレスからのトラフィックを統合
- すべてのシナリオ: ENI の関連付け制限を超えるトラフィックルーティングを要求

このルーティングは、フェイルオーバーシナリオ時の動的なルート更新のために VPC Route Server に統合できます。

主な利点

このルーティングアプローチには以下の利点があります。

- アドレス変換が不要 – 直接的なルーティングが NAT の複雑性を解消
- ENI 制限を回避 – インスタンスの関連付け制限を超える IP プールを処理
- 業界最適化 – 電気通信と IoT の要件に特化した設計
- 動的フェイルオーバー – 自動更新のために Route Server と統合

可用性

この機能は、すべての AWS 商用リージョン、AWS 中国リージョン、および AWS GovCloud リージョンで使用できます。

内容

- 開始する前に
- この機能の仕組み
- ステップ 1: VPC を作成する
- ステップ 2: インターネットゲートウェイを作成してアタッチする
- ステップ 3: ターゲットインスタンスのサブネットを作成する
- ステップ 4: サブネットのルートテーブルを作成する
- ステップ 5: ターゲットインスタンスのセキュリティグループを作成する
- ステップ 6: ターゲット EC2 インスタンスを起動する
- ステップ 7: インターネットゲートウェイのルートテーブルを作成する
- ステップ 8: ルートテーブルをインターネットゲートウェイに関連付ける
- ステップ 9: BYOIP プールをインターネットゲートウェイに関連付ける
- ステップ 10: インスタンスをターゲットにする静的ルートを追加する
- ステップ 11: ターゲットインスタンスを設定する
- ステップ 12: トラフィック処理用のインスタンスを設定する
- ステップ 13: 接続をテストする
- トラブルシューティング
- 高度なオプション: 動的ルーティングのための Route Server 統合
- クリーンアップ

開始する前に

このチュートリアルを開始する前に、以下を確認してください。

1. BYOIP プール: 独自の IP アドレス範囲が既に AWS に持ち込まれている必要があります。「Amazon EC2 に自分の IP アドレスを持ち込む (BYOIP)」の手順を完了してください。
2. BYOIP プールの検証: 以下を実行して、プールの使用準備が整っていることを確認します。

```
aws ec2 describe-public-ipv4-pools --region us-east-1
```

   出力内でプールを見つけて、PoolAddressRanges に Available アドレスが表示されていることを確認します。
3. 適切な許可: AWS VPC リソースと EC2 インスタンスを作成し、BYOIP プールを管理する許可がアカウントにあることを確認します。

この機能の仕組み

このセクションでは、インターネットゲートウェイのイングレスルーティングの背景にある技術的な概念と、トラフィックがインターネットからターゲットインスタンスに流れる仕組みについて説明します。

インターネットゲートウェイのイングレスルーティングを使用する理由

これまでは、ENI の関連付けに制限があったことから、多数の IP アドレスのトラフィックを統合するためのアドレス変換を実行する必要がありました。この機能強化は、ターゲットインスタンスへの BYOIP プールの直接的なルーティングを可能にすることで、このような複雑性を解消します。

ルーティングの仕組み

この機能は、BYOIP プロセスに従ってユーザーが AWS に持ち込むパブリック IP CIDR でのみ動作します。BYOIP プロセスは、アカウントがパブリック IP CIDR を所有していることを確実にします。BYOIP パブリック CIDR を取得したら、以下を実行します。

1. このパブリック IP アドレスプールをインターネットゲートウェイのルートテーブルに関連付けます。インターネットゲートウェイは、既に VPC に関連付けられている必要があります。この関連付けにより、VPC が IP CIDR 宛てトラフィックを受け入れることが可能になります。インターネットゲートウェイに、サブネットと共有されていない専用のルートテーブルがあることを確認してください。
2. BYOIP プールをインターネットゲートウェイのルートテーブルに関連付けたら、送信先が IP CIDR と等しいルート、またはそのサブセットをインターネットゲートウェイのルートテーブルに入力できるようになります。このルートのターゲットは、トラフィックをルーティングする ENI になります。

3. BYOIP CIDR 宛てのトラフィックが AWS 内に入ると、AWS がインターネットゲートウェイのルートテーブルを調べ、それに応じてトラフィックを関連する VPC にルーティングします。
4. VPC 内では、インターネットゲートウェイがトラフィックをターゲット ENI にルーティングします。
5. ターゲット (ワークロードに関連付けられた Elastic Network Interface) はトラフィックを処理します。

ベストプラクティス:

- ルートテーブルを分離する: インターネットゲートウェイのルートテーブルは、インターネットゲートウェイ専用である必要があります。このルートテーブルはどの VPC サブネットにも関連付けないでください。サブネットルーティングには別のルートテーブルを使用します。
- BYOIP IP を直接割り当てない: BYOIP プールからのパブリック IP アドレスを EC2 インスタンスやネットワークインターフェイスに直接関連付けないでください。インターネットゲートウェイのイングレスルーティング機能は、直接的な IP の関連付けを必要とすることなく、トラフィックをインスタンスにルーティングします。

> **Important**
>
> VPC ブロックパブリックアクセス (BPA) を使用している場合に BPA を有効にすると、サブネットレベルで BPA 除外を設定している場合でも、イングレスルーティングを使用するサブネットへのトラフィックがブロックされます。イングレスルーティングでは、サブネットレベルでの除外が機能しません。BPA を有効にした状態でイングレスルーティングトラフィックを許可するには、以下を実行します。
>
> - BPA を完全に無効化する、または
> - VPC レベルでの除外を使用する

ステップ 1: VPC を作成する

このステップを実行して、ターゲットインスタンスとインターネットゲートウェイをホストする VPC を作成します。

> **Note**
>
> VPC のクォータ制限に到達していないことを確認してください。詳細については、「Amazon VPC クォータ」を参照してください。

AWS コンソール

1. Amazon VPC コンソール を開きます。
2. VPC ダッシュボードで、[Create VPC (VPC を作成する)] を選択します。
3. [作成するリソース]で、[VPC のみ] を選択します。
4. [名前タグ] には、VPC の名前を入力します (**IGW-Ingress-VPC** など)。
5. [IPv4 CIDR ブロック] には、CIDR ブロックを入力します (**10.0.0.0/16** など)。
6. [Create VPC (VPC の作成)] を選択します。

AWS CLI

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications
 'ResourceType=vpc,Tags=[{Key=Name,Value=IGW-Ingress-VPC}]' --region us-east-1
```

ステップ 2: インターネットゲートウェイを作成してアタッチする

このステップを実行して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、インターネット接続を有効にします。

AWS コンソール

1. Amazon VPC コンソール を開きます。
2. VPC コンソールで [インターネットゲートウェイ] を選択します。
3. [インターネットゲートウェイの作成] を選択します。
4. [名前タグ] には、インターネットゲートウェイの名前を入力します (**IGW-Ingress-Gateway** など)。
5. [インターネットゲートウェイの作成] を選択します。
6. インターネットゲートウェイを選択してから、[アクション]、[VPC にアタッチ] の順に選択します。
7. 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。

AWS CLI

```
aws ec2 create-internet-gateway --tag-specifications 'ResourceType=internet-
gateway,Tags=[{Key=Name,Value=IGW-Ingress-Gateway}]' --region us-east-1

aws ec2 attach-internet-gateway --internet-gateway-id igw-0123456789abcdef0 --vpc-id
 vpc-0123456789abcdef0 --region us-east-1
```

注意: リソース ID を前のステップからの実際の ID に置き換えてください。

ステップ 3: ターゲットインスタンスのサブネットを作成する

このステップを実行して、ターゲットインスタンスのデプロイ先になるサブネットを作成します。

AWS コンソール

1. VPC コンソールのナビゲーションペインで、[Subnets] (サブネット) を選択します。
2. [サブネットの作成] を選択します。
3. [VPC ID] で VPC を選択します。
4. [サブネット名] に名前を入力します (**`Target-Subnet`** など)。
5. [Availability Zone] (アベイラビリティーゾーン) で、サブネットのゾーンを選択するか、デフォルトの [No Preference] (設定なし) のままにして AWS が選択できるようにします。
6. [IPv4 CIDR ブロック] には [手動入力] を選択し、CIDR ブロックを入力します (**`10.0.1.0/24`** など)。
7. [サブネットの作成] を選択します。

AWS CLI

```
aws ec2 create-subnet \
    --vpc-id vpc-0123456789abcdef0 \
    --cidr-block 10.0.1.0/24 \
    --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Target-Subnet}]' \
    --region us-east-1
```

ステップ 4: サブネットのルートテーブルを作成する

このステップを実行してサブネットのルートテーブルを作成し、サブネットに関連付けます。

AWS コンソール

1. VPC コンソールのナビゲーションペインで [ルートテーブル] を選択します。
2. [ルートテーブルの作成] を選択します。
3. [名前] にルートテーブルの名前を入力します (**Target-Subnet-Route-Table** など)。
4. [VPC] で、ユーザーの VPC を選択します。
5. [ルートテーブルの作成] を選択します。
6. ルートテーブルを選択してから、[アクション]、[サブネットの関連付けの編集] の順に選択します。
7. サブネットを選択して、[関連付けを保存] を選択します。

AWS CLI

```
aws ec2 create-route-table \
    --vpc-id vpc-0123456789abcdef0 \
    --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=Target-Subnet-
Route-Table}]' \
    --region us-east-1

aws ec2 associate-route-table \
    --route-table-id rtb-0987654321fedcba0 \
    --subnet-id subnet-0123456789abcdef0 \
    --region us-east-1
```

ステップ 5: ターゲットインスタンスのセキュリティグループを作成する

このステップを実行して、ターゲットインスタンスへのネットワークアクセスを制御するセキュリティグループを作成します。

AWS コンソール

1. VPC コンソールのナビゲーションペインで [セキュリティグループ] を選択します。
2. セキュリティグループの作成 を選択します。
3. [セキュリティグループ名] に名前を入力します (**IGW-Target-SG** など)。
4. [説明] に **Security group for IGW ingress routing target instance** と入力します。
5. [VPC] で、自分の VPC を選択します。
6. インバウンドルールを追加するには、[インバウンドルール] を選択します。ルールごとに [ルールを追加] を選択して、以下を指定します。

- [タイプ]: すべての ICMP – IPv4、[ソース]: 0.0.0.0/0 (ping テスト用) 。
- [タイプ]: SSH、[ポート]: 22、[ソース]: 0.0.0.0/0 (EC2 Instance Connect 用)。

> **Note**
>
> このセキュリティグループは、このチュートリアルのすべてのインターネットトラフィックに対して SSH ポートを開きます。このチュートリアルは教育を目的としているため、本番環境には設定しないでください。本番環境では、SSH アクセスを特定の IP 範囲に制限します。

- [セキュリティグループの作成] を選択してください。

AWS CLI

```
aws ec2 create-security-group \
    --group-name IGW-Target-SG \
    --description "Security group for IGW ingress routing target instance" \
    --vpc-id vpc-0123456789abcdef0 \
    --region us-east-1

aws ec2 authorize-security-group-ingress \
    --group-id sg-0123456789abcdef0 \
    --protocol icmp \
    --port -1 \
    --cidr 0.0.0.0/0 \
    --region us-east-1

aws ec2 authorize-security-group-ingress \
    --group-id sg-0123456789abcdef0 \
    --protocol tcp \
    --port 22 \
    --cidr 0.0.0.0/0 \
    --region us-east-1
```

ステップ 6: ターゲット EC2 インスタンスを起動する

このステップを実行して、BYOIP プールからのトラフィックを受信する EC2 インスタンスを起動します。

AWS コンソール

1. Amazon EC2 コンソールを開きます。
2. [インスタンスを起動] を選択してください。
3. [名前] にインスタンスの名前を入力します (**IGW-Target-Instance** など)。
4. [アプリケーションと OS イメージ (Amazon マシンイメージ)] には、[Amazon Linux 2023 AMI] を選択します。
5. [インスタンスタイプ] には [t2.micro] (無料利用枠対象) を選択します。
6. [キーペア (ログイン)] には、既存のキーペアを選択するか、新しいキーペアを作成します。
7. [ネットワーク設定] で [編集] を選択します。
   - [VPC]: 使用する VPC を選択します
   - [サブネット]: 使用するサブネットを選択します
   - [パブリック IP の自動割り当て]: 有効にします
   - [ファイアウォール (セキュリティグループ)]: 既存のセキュリティグループを選択するか、独自のセキュリティグループを選択します
8. [インスタンスを起動] を選択してください。
9. 重要: 起動後はインスタンスの詳細に移動して、[ネットワークインターフェイス ID] (「eni-」で始まるもの) を書き留めてください。この ID はステップ 10 で必要になります。

AWS CLI

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --count 1 \
    --instance-type t2.micro \
    --key-name your-key-pair \
    --security-group-ids sg-0123456789abcdef0 \
    --subnet-id subnet-0123456789abcdef0 \
    --associate-public-ip-address \
    --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=IGW-Target-
Instance}]' \
    --region us-east-1
```

コンソールで ENI ID を見つける:

1. EC2 コンソールでインスタンスを選択します。

2. [ネットワーク] タブに移動します。
3. [ネットワークインターフェイス ID] (`eni-0abcdef1234567890` など) を書き留めます。

を使用して ENI ID を見つける:AWS CLI

```
aws ec2 describe-instances --instance-ids i-0123456789abcdef0 --query
 'Reservations[0].Instances[0].NetworkInterfaces[0].NetworkInterfaceId' --output text
 --region us-east-1
```

ステップ 7: インターネットゲートウェイのルートテーブルを作成する

このステップを実行して、イングレスルーティングを処理するインターネットゲートウェイ専用のルートテーブルを作成します。

AWS コンソール

1. VPC コンソールで [ルートテーブル] を選択します。
2. [ルートテーブルの作成] を選択します。
3. [名前] にルートテーブルの名前を入力します (**`IGW-Ingress-Route-Table`** など)。
4. [VPC] で、ユーザーの VPC を選択します。
5. [ルートテーブルの作成] を選択します。
6. ルートテーブルを選択してから、[エッジの関連付け] タブを選択します。
7. [エッジの関連付けを編集] を選択します。
8. インターネットゲートウェイを選択してから、[変更を保存] を選択します。

AWS CLI

```
aws ec2 create-route-table \
    --vpc-id vpc-0123456789abcdef0 \
    --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=IGW-Ingress-
Route-Table}]' \
    --region us-east-1
```

ステップ 8: ルートテーブルをインターネットゲートウェイに関連付ける

このステップを実行してルートテーブルをインターネットゲートウェイに関連付け、イングレスルーティング機能を有効にします。

AWS コンソール

1. VPC コンソールのナビゲーションペインで [ルートテーブル] を選択してから、作成したルートテーブルを選択します。
2. [Edge associations] (エッジの関連付け) タブから、[Edit edge associations] (Edge の関連付けを編集) を選択します。
3. インターネットゲートウェイのチェックボックスにチェックを入れます。
4. [Save changes] (変更の保存) をクリックします。

AWS CLI

```
aws ec2 associate-route-table \
    --route-table-id rtb-0123456789abcdef0 \
    --gateway-id igw-0123456789abcdef0 \
    --region us-east-1
```

ステップ 9: BYOIP プールをインターネットゲートウェイに関連付ける

このステップを実行して BYOIP プールをインターネットゲートウェイのルートテーブルに関連付け、VPC が IP 範囲内のトラフィックを受け入れるようにします。

AWS コンソール

1. VPC コンソールのナビゲーションペインで [ルートテーブル] を選択してから、作成したインターネットゲートウェイのルートテーブルを選択します。
2. [IPv4 プールの関連付け] タブを選択します。
3. [関連付けを編集] ボタンを選択します。
4. BYOIP プールを選択します (`pool-12345678901234567` など)。
5. [関連付けを保存] ボタンを選択します。

AWS CLI

```
aws ec2 associate-route-table \
    --route-table-id rtb-0123456789abcdef0 \
    --public-ipv4-pool pool-12345678901234567 \
    --region us-east-1
```

注意: rtb-0123456789abcdef0 を独自のインターネットゲートウェイルートテーブル ID に置き換えて、pool-12345678901234567 を独自の BYOIP プール ID に置き換えてください。

ステップ 10: インスタンスをターゲットにする静的ルートを追加する

このステップを実行して、BYOIP 範囲からのトラフィックをターゲットインスタンス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追加します。

AWS コンソール

1. VPC コンソールのナビゲーションペインで [ルートテーブル] を選択してから、作成したインターネットゲートウェイのルートテーブルを選択します。
2. [アクション]、[ポリシーの編集] の順に選択します。
3. [Add Rule (ルートの追加)] を選択します。
4. [送信先] には、BYOIP CIDR またはサブセット (**203.0.113.0/24** など) を入力します。プレフィックスは /23 と /28 の間のものにする必要があります。
5. [ターゲット] に [ネットワークインターフェイス] を選択し、インスタンスの ENI ID (eni-0abcdef1234567890 など) を入力します。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

```
aws ec2 create-route \
    --route-table-id rtb-0123456789abcdef0 \
    --destination-cidr-block 203.0.113.0/24 \
    --network-interface-id eni-0abcdef1234567890 \
    --region us-east-1
```

ステップ 11: ターゲットインスタンスを設定する

このステップを実行して、BYOIP アドレス宛てのトラフィックを適切に処理するようにターゲットインスタンスを設定します。

重要: このインスタンス設定ステップは、接続をテスト (ステップ 12) する前に完了してください。イングレスルーティングを正しく機能させるには、BYOIP アドレスに応答するようにインスタンスを設定する必要があります。

AWS コンソール

1. EC2 Instance Connect を使用してターゲットインスタンスに接続します。

   - EC2 コンソールでインスタンスを選択します。
   - [アクション] > [接続] の順に選択します。
   - [EC2 Instance Connect] タブを選択します。
   - 接続 を選択します。

2. インスタンスインターフェイスに特定の BYOIP IP アドレスを追加します。

   まず、ネットワークインターフェイス名を見つけます。

   ```
   ip link show
   ```

   次に、IP アドレスを追加します (203.0.113.10 を BYOIP 範囲からの IP に置き換えてください)。

   ```
   sudo ip addr add 203.0.113.10/32 dev eth0
   ```

   注意: 203.0.113.10 は、テストする BYOIP 範囲からの IP アドレスに置き換えてください。インターフェイス名は、インスタンスタイプに応じて eth0、ens5、または同様の名前になっている場合があります。

3. EC2 コンソールで、送信元/送信先チェックを無効にします。

   - インスタンスを選択してください。
   - [ネットワーク] タブに移動して、ネットワークインターフェイスを選択します。
   - [アクション]、[送信元/送信先チェックを変更]、[無効化] の順にクリックします。

AWS CLI

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-0abcdef1234567890 \
    --no-source-dest-check \
    --region us-east-1
```

ステップ 12: トラフィック処理用のインスタンスを設定する

このステップを実行してインスタンスに BYOIP アドレスを追加し、送信元/送信先チェックを無効にして、適切なトラフィック処理を有効にします。

AWS コンソール

1. EC2 Instance Connect を使用してターゲットインスタンスに接続します。
   - EC2 コンソールでインスタンスを選択します。
   - [アクション] > [接続] の順に選択します。
   - [EC2 Instance Connect] タブを選択します。
   - 接続 を選択します。
2. インスタンスインターフェイスに特定の BYOIP IP アドレスを追加します。

   まず、ネットワークインターフェイス名を見つけます。

   ```
   ip link show
   ```

   次に、IP アドレスを追加します (ens5 を実際のインターフェイス名に置き換えます)。

   ```
   sudo ip addr add 203.0.113.10/32 dev ens5
   ```

   注意: 203.0.113.10 は、テストする BYOIP 範囲からの IP アドレスに置き換えてください。インターフェイス名は、インスタンスタイプに応じて eth0、ens5、または同様の名前になっている場合があります。
3. EC2 コンソールで、送信元/送信先チェックを無効にします。
   - インスタンスを選択してください。
   - [ネットワーク] タブに移動して、ネットワークインターフェイスを選択します。
   - [アクション]、[送信元/送信先チェックを変更]、[無効化] の順にクリックします。

AWS CLI

```
aws ec2 modify-network-interface-attribute \
    --network-interface-id eni-0abcdef1234567890 \
    --no-source-dest-check \
    --region us-east-1
```

ステップ 13: 接続をテストする

このステップを実行して、インターネットトラフィックが BYOIP アドレスを経由してターゲットインスタンスに適切にルーティングされていることを確認します。

1. ターゲットインスタンスで、tcpdump を使用して受信トラフィックを監視します。

```
sudo tcpdump -i any icmp
```

2. 別のターミナルまたはコンピュータから、BYOIP IP アドレスへの接続をテストします。

```
ping 203.0.113.10
```

3. 予想される結果:

   - Ping が成功し、BYOIP IP アドレスからのレスポンスが表示されます。
   - tcpdump には、以下のような BYOIP アドレスの受信パケットが表示されます。

```
12:34:56.789012 IP 203.0.113.100 > 203.0.113.10: ICMP echo request, id 1234, seq
 1, length 64
12:34:56.789123 IP 203.0.113.10 > 203.0.113.100: ICMP echo reply, id 1234, seq 1,
 length 64
```

   - トラフィックは外部 IP アドレスからのものであるように表示されます。これは、インターネットゲートウェイのイングレスルーティングがインスタンスにインターネットトラフィックを配信していることを証明しています。

トラブルシューティング

このセクションを使用して、インターネットゲートウェイのイングレスルーティングの設定時に発生する可能性のある一般的な問題を解決します。

トラフィックがインスタンスに到達しない

- ルートテーブルのターゲットとして正しい ENI ID が設定されていることを確認します。
- BYOIP プールがインターネットゲートウェイのルートテーブルに関連付けられていることを確認します。
- インスタンスで送信元/送信先チェックが無効になっていることを確認します。
- セキュリティグループがテスト対象のトラフィックタイプを許可していることを確認します。

ルートの作成が失敗する

- BYOIP プールがルートテーブルに適切に関連付けられていることを確認します。
- 送信先 CIDR が BYOIP 範囲内であることを確認します。
- ターゲット ENI が存在しており、実行中のインスタンスにアタッチされていることを確認します。
- BYOIP プレフィックスが /23 から /28 の間のものであることを確認します (この範囲外のプレフィックスはサポートされません)。

Ping/接続が失敗する

- IP アドレスがインスタンスインターフェイスに追加されていることを確認します。
- セキュリティグループが ICMP (ping 用) または関連するポートを許可していることを確認します。
- インスタンスが実行状態になっていることを確認します。
- 外部にある複数の場所からテストします。

高度なオプション: 動的ルーティングのための Route Server 統合

自動フェイルオーバーを必要とする環境では、この機能が VPC Route Server と統合して以下を実行します。

- インスタンス障害が発生した時にルートを動的に更新します。
- ルート管理における手動介入を排除します。
- 重要なワークロードにエンタープライズグレードの可用性を提供します。

これは、高可用性が不可欠な電気通信や IoT のユースケースにとって特に重要です。

> **Note**
>
> 複数の BGP ピアで Route Server を使用する場合は、最大 32 の BGP ピアが、ルートサーバーを使用して同じルートテーブルに同じプレフィックスをアドバタイズできることに注意してください。

動的ルーティング、自動フェイルオーバー、複数のインスタンス全体での負荷分散を必要とする環境では、AWS Route Server との統合を検討してください。Route Server は、静的ルートの代わりに BGP ベースの動的ルーティングを有効にすることで、以下を実現します。

- BGP 経由でのインスタンスからの動的ルートアドバタイズ。
- 複数のターゲットインスタンス間での自動フェイルオーバー。
- 複数のエンドポイント全体での負荷分散。
- BGP プロトコルを使用した一元的なルート管理。

これは、高可用性と動的ルーティング機能が必要とされるエンタープライズデプロイの重要なユースケースです。Route Server の詳しい設定手順については、「AWS Route Server ドキュメント」を参照してください。

クリーンアップ

このチュートリアルで作成したリソースは、継続的な料金の発生を回避するために削除してください。

ステップ 1: EC2 インスタンスを終了する

このステップを実行して EC2 インスタンスを終了し、コンピューティングリソースの料金が発生しないようにします。

AWS コンソール

1. Amazon EC2 コンソールを開きます。
2. EC2 コンソールの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してから、[インスタンスの状態]、[インスタンスの終了] の順に選択します。
4. [終了] を選択して確認します。

AWS CLI

```
aws ec2 terminate-instances --instance-ids i-0123456789abcdef0 --region us-east-1
```

ステップ 2: VPC からインターネットゲートウェイをデタッチする

このステップを実行して、VPC からインターネットゲートウェイをデタッチし、削除します。

AWS コンソール

1. Amazon VPC コンソール を開きます。
2. VPC コンソールのナビゲーションペインで [インターネットゲートウェイ] を選択します。
3. インターネットゲートウェイを選択してから、[アクション]、[VPC からデタッチ] の順に選択します。
4. [インターネットゲートウェイをデタッチ] を選択します。
5. デタッチしたら、[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。
6. [インターネットゲートウェイを削除] を選択します。

AWS CLI

```
aws ec2 detach-internet-gateway --internet-gateway-id igw-0123456789abcdef0 --vpc-id
 vpc-0123456789abcdef0 --region us-east-1

aws ec2 delete-internet-gateway --internet-gateway-id igw-0123456789abcdef0 --region
 us-east-1
```

ステップ 3: VPC を削除する

このステップを実行して VPC とすべての関連リソースを削除し、クリーンアッププロセスを完了します。

AWS コンソール

1. VPC コンソールで [VPC] を選択します。
2. VPC を選択してから、[アクション]、[VPC の削除] の順に選択します。
3. **delete** と入力して確定し、[削除] を選択します。

AWS CLI

```
aws ec2 delete-vpc --vpc-id vpc-0123456789abcdef0 --region us-east-1
```

> **Note**
>
> VPC を削除すると、関連付けられたサブネット、ルートテーブル、セキュリティグループも削除されます。

> **Note**
>
> BYOIP プールはこの後も引き続き使用でき、このクリーンアッププロセスの一環として削除されることはありません。

VPC Route Server を使用する VPC 内の動的ルーティング

Amazon VPC Route Server は、VPC とそのインターネットゲートウェイ内にデプロイされたワークロード間のトラフィックのルーティングを簡略化します。この機能を使用すると、VPC Route Server は VPC とインターネットゲートウェイのルートテーブルを任意の IPv4 または IPv6 ルートで動的に更新し、それらのワークロードのルーティングの耐障害性を実現します。これにより、VPC 内のトラフィックを自動的に再ルーティングできるため、VPC ルーティングの管理の容易性とサードパーティーワークロードとの相互運用性が向上します。

ルートサーバーは、次のルートテーブルタイプに対応しています:

- サブネットに関連付けられていない VPC ルートテーブル
- サブネットのルートテーブル
- インターネットゲートウェイのルートテーブル

ルートサーバーは、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に対応していません。トランジットゲートウェイのルートテーブルにルートを伝達するには、Transit Gateway Connect を使用します。

クォータ

Amazon VPC Route Server に関連付けられたクォータについては、「Route server quotas」を参照してください。

料金

Amazon VPC Route Server に関連するコストについては、Amazon VPC 料金ページの VPC Route Server タブを参照してください。

内容

- 用語
- Amazon VPC Route Server の仕組み

- ルートサーバーピアのログ記録
- 使用開始チュートリアル

用語

このガイドでは次の用語が使用されます:

- FIB: Forwarding Information Base (FIB) は、利用可能なすべてのルーティング情報とポリシーを評価した後、RIB のベストパスルートであるとサーバーが判断したルートの転送テーブルとして機能します。ルートテーブルにインストールされている FIB ルート。FIB は、RIB に変更があるたびに再計算されます。
- RIB: Routing Information Base (RIB) は、BGP ピアから学習されたルートなど、ルーターまたはルーティングシステムが収集したすべてのルーティング情報とネットワークトポロジーデータを保存するデータベースとして機能します。RIB は、新しいルーティング情報を受信したり、既存のルートが変更されたりすると、常に更新されます。これにより、ルートサーバーには常にネットワークトポロジーの最新のビューがあり、最適なルーティング決定を行うことができます。
- ルートサーバー: ルートサーバーコンポーネントは、Forwarding Information Base (FIB) の IPv4 または IPv6 ルートで VPC およびインターネットゲートウェイルートテーブルを更新します。ルートサーバーは、単一の FIB と Routing Information Base (RIB) を表します。
- ルートサーバーの関連付け: ルートサーバーの関連付けは、ルートサーバーと VPC の間で確立された接続です。
- ルートサーバーエンドポイント: ルートサーバーエンドポイントは、ルートサーバーと BGP ピア間の BGP (ボーダーゲートウェイプロトコル) 接続を容易にするサブネット内の AWS マネージドコンポーネントです。
- ルートサーバーピア: ルートサーバーピアとは、ルートサーバーエンドポイントと AWS にデプロイされたデバイス (ファイアウォール機器や EC2 インスタンスで実行されているその他のネットワークセキュリティ機能など) との間のセッションです。デバイスは、以下の条件を満たす必要があります:
  - VPC に Elastic Network Interface がある
  - BGP (ボーダーゲートウェイプロトコル) をサポート
  - BGP セッションを開始できる
- ルートサーバーの伝播: 有効にすると、ルートサーバーの伝播は指定したルートテーブルの FIB にルートをインストールします。ルートサーバーは IPv4 および IPv6 のルート伝達に対応しています。

Amazon VPC Route Server の仕組み

このセクションでは、Amazon VPC Route Server の仕組みについて説明しており、サブネットで実行されているワークロードのルーティングの耐障害性を実現する方法を理解するのに役立ちます。

内容

- 概要
- 図表

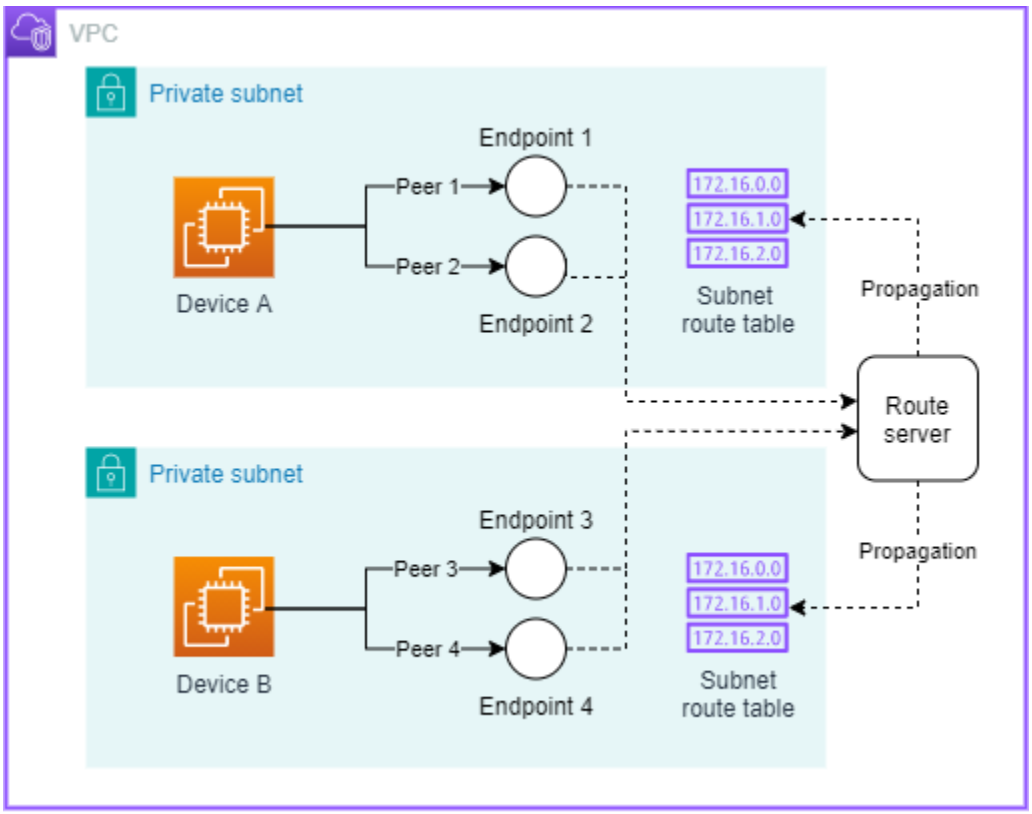
概要

Amazon VPC Route Server の仕組み:

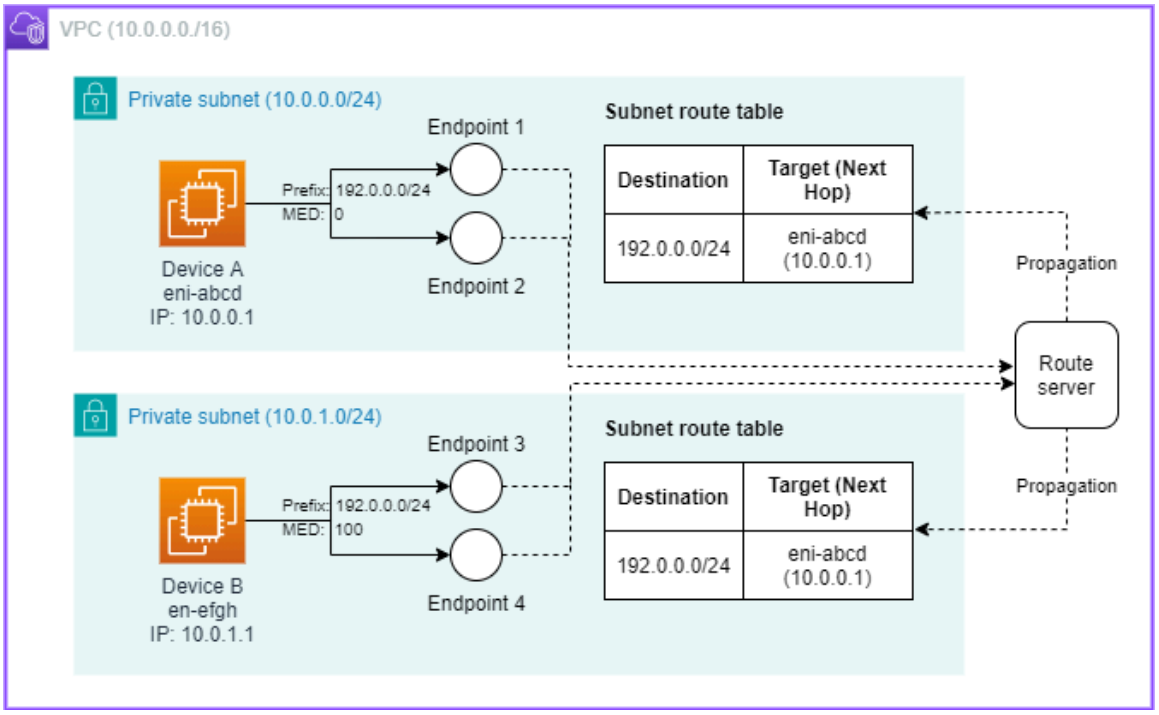
1. Amazon VPC Route Server を使用するように、ネットワークデバイス (VPC の EC2 インスタンスで実行されているファイアウォールなど) を設定します。
2. ネットワークデバイスに障害が発生します。
3. ルートサーバーエンドポイントは、ルートサーバーピアで構成された BFD Bidirectional Forwarding Detection) を通じて障害を検出します。
4. ルートサーバーエンドポイントは、ルートサーバーを更新して、障害が発生したデバイスがネクストホップであるRouting Information Base (RIB) 内のルートを取り消します。
5. ルートサーバーは、RIB から Forwarding Information Base (FIB) を計算し、利用可能な最適なルートを選択します。
6. Route Server は、構成済みのルートテーブルを FIB からのルートで更新します。
7. すべての新しいトラフィックはスタンバイデバイスに転送されます。

図表

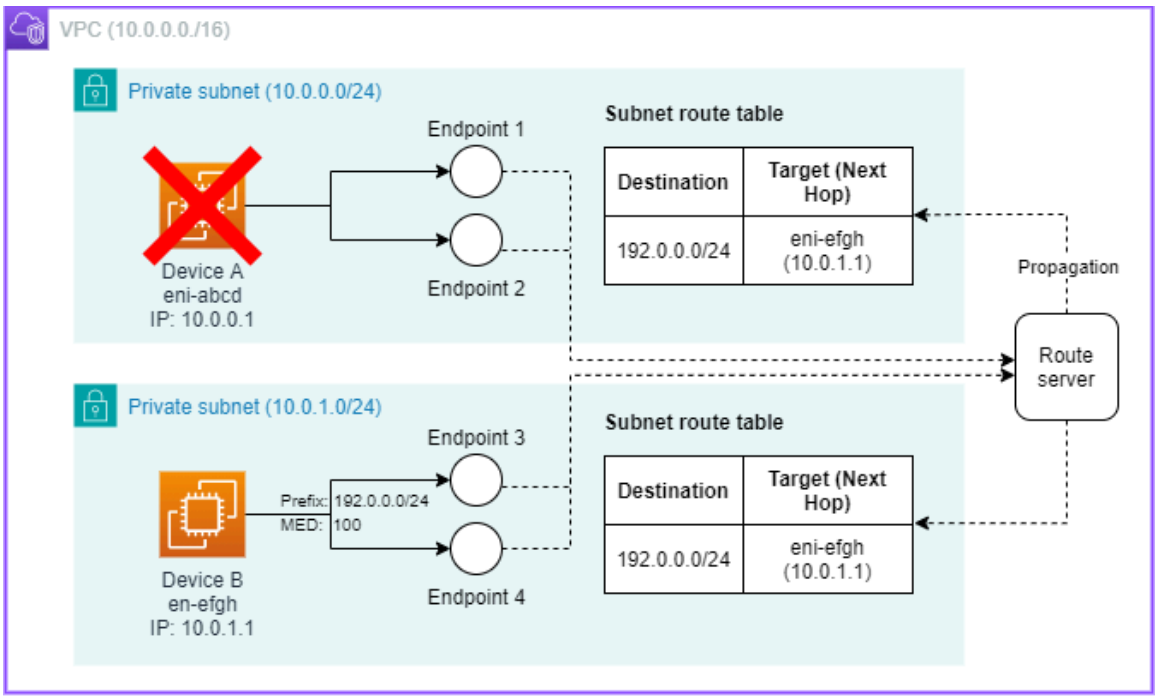
2 つのサブネット内のデバイス用に構成されたルートサーバーエンドポイントを持つ VPC ルートサーバーの図の例を次に示します。



上記の例をベースラインとして開始すると、以下の例はより詳細な設計を示しています。ここでは、デバイス A とデバイス B の両方が、送信先 IP が 192.0.0.0/24 (192.0.0.0 から 192.0.0.255) の範囲のトラフィックを受け入れることができることを BGP 経由でアドバタイズしています。MED (Multi-Exit Discriminator) 属性が 0 の場合、デバイス A がデバイス B よりも優先されるべきであることがルートサーバーに通知されます。ルートサーバーは、デバイス A からルートと MED 属性を受け取り、そのルートをサブネットルートテーブルにインストールします。その際、デバイス A のネットワークインターフェイスを「次のホップ」として使用します。その結果、送信先 IP が 192.0.0.0/24 の範囲にあるサブネット内のトラフィックは、デバイス A に送信されます。その時デバイス A はトラフィックを処理し、その先に送信します。192.0.0.0/24 にバインドされているサブネット (10.0.0.0/24 または 10.0.1.0/24) 内のトラフィックは、ネクストホップとしてデバイス A eni-abcd (10.0.0.1) にルーティングされます。



以下の最後の例は、ルートサーバーがフェイルオーバーを処理する方法を示しています。MED 属性が高いほど、デバイス B がデバイス A よりも優先されていないことがルートサーバーに通知されますが、デバイス A eni-abcd (10.0.0.1) がダウンした場合、ルートサーバーはサブネットルートテーブルを更新し、192.0.0.0/24 へのトラフィックはネクストホップとしてデバイス B eni-efgh (10.0.1.1) にルーティングされます。



ルートサーバーピアのログ記録

次を実行する必要がある場合は、VPC Route Server ピアのログ記録を使用します。

- BGP セッションと BFD セッションの状態を監視する
- 接続問題をトラブルシューティングする
- セッションの変更履歴を確認する
- ネットワークステータスを追跡する

料金

- CloudWatch: ルートサーバーピアログを CloudWatch Logs に発行すると、vended log に対するデータインジェスト料金とアーカイブ料金が適用されます。
- S3: ルートサーバーピアログを Amazon S3 に発行すると、vended log に対するデータインジェスト料金とアーカイブ料金が適用されます。
- Data Firehose: 標準の取り込み料金と配信料金が適用されます。

vended log とは、ボリューム階層化料金で利用できる特定の AWS サービスからのログで、CloudWatch Logs、Amazon S3、または Amazon Data Firehose に配信されます。詳細については、「Amazon CloudWatch 料金表」を開いて [ログ] を選択し、[Vended Logs] を参照してください。

ログ形式の例

```
{
    "resource_arn": "arn:aws:ec2:us-east-1:111122223333:route-server-peer/
rsp-1234567890abcdef0",
    "event_timestamp": 1746643505367,
    "type": "RouteStatus",
    "status": "ADVERTISED",
    "message": {
        "prefix": "10.24.34.0/32",
        "asPath": "65000",
        "med": 100,
        "nextHopIp": "10.24.34.1"
    }
}

{
```

```
    "resource_arn": "arn:aws:ec2:us-east-1:111122223333:route-server-peer/
rsp-1234567890abcdef0",
    "event_timestamp": 1746643490000,
    "type": "BGPStatus",
    "status": "UP",
    "message": null
}
```

説明は以下のとおりです。

- resource_arn はルートサーバーピアの ARN です。
- event_timestamp はイベントのタイムスタンプです。
- 生成するログイベントの type (RouteStatus、BGPStatus、BFDStatus)。
- status フィールドはステータス更新です。
  - RouteStatus タイプのメッセージの場合
    - ADVERTISED (ルートがピアによってアドバタイズされた)
    - UPDATED (既存のルートがピアによって更新された)
    - WITHDRAWN (ルートがピアによって取り消された)
  - BFDStatus および BGPStatus 更新の場合
    - UP, DOWN.
- 現在、message フィールドは RouteStatus メッセージタイプのルート属性のみに使用されていますが、任意のタイプの関連情報を入力することができます。

AWS Management Console

ルートサーバーピアログの作成方法:

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [仮想プライベートクラウド] で、[ルートサーバー] を選択します。
3. [ルートサーバー] ページで、[ルートサーバーピア] を選択します。
4. [ログ配信] タブを選択します。
5. [ログ配信を追加] を選択します。
6. 送信先を選択し、設定を行います。

   - Amazon CloudWatch Logs

- ログタイプ: 配信するログのタイプ。サポートされているログタイプは EVENT_LOGS のみです。
- 配信先のロググループ: ログが送信される CloudWatch ロググループ。既存のロググループを選択するか、新しいロググループを作成できます (例: /aws/vpc/route-server-peers)。
- フィールド選択: ログに含めるデータフィールド。
- 出力形式: ログのフォーマット方法:
  - JSON: コンピュータ処理用に構造化された形式
  - テキスト: プレーンテキスト形式
- フィールド区切り文字: テキスト形式を使用する場合、これがフィールドを区切る文字になります (例: カンマ、タブ、スペース)。

- Amazon S3
  - クロスアカウント - 異なる AWS アカウントにログを送信
    - ログタイプ: 配信するログのタイプ。サポートされているログタイプは EVENT_LOGS のみです。
    - 配信先 ARN: ログの送信先である別の AWS アカウント内の S3 バケットの Amazon リソースネーム。
    - フィールド選択: ログに含めるデータフィールド。
    - サフィックス: ログファイル名の末尾に追加する要素 (例: .log、.txt)。
    - Hive 互換: オンにすると、Hive ベースのツールで機能するフォルダ構造でログが整理され、Amazon Athena などのサービスでの検索が容易になります。
    - フィールド区切り文字: テキスト形式を使用する場合、これがフィールドを区切る文字になります。
  - 現行のアカウント
    - ログタイプ: 配信するログのタイプ。サポートされているログタイプは EVENT_LOGS のみです。
    - 送信先 S3 バケット: ログが送信されるアカウント内の S3 バケット。サブフォルダパスを指定できます。
    - フィールド選択: ログに含めるデータフィールド。
    - サフィックス: ログファイル名の末尾に追加する要素 (例: .log、.txt)。
    - Hive 互換: オンにすると、Hive ベースのツールで機能するフォルダ構造でログが整理され、検索しやすくなります。

        - フィールド区切り文字: テキスト形式を使用する場合、これがフィールドを区切る文字になります。
    - Amazon Data Firehose
        - クロスアカウント
            - ログタイプ: 配信するログのタイプ。サポートされているログタイプは EVENT_LOGS のみです。
            - 配信先 ARN: 別の AWS アカウント内にある Firehose 配信ストリームの Amazon リソースネーム。
            - フィールド選択: ログに含めるデータフィールド。
            - フィールド区切り文字: テキスト形式を使用する場合、これがフィールドを区切る文字になります。
        - 現行のアカウント
            - ログタイプ: 配信するログのタイプ。サポートされているログタイプは EVENT_LOGS のみです。
            - 配信先ストリーム: ログが送信されるアカウント内の Firehose 配信ストリーム。ストリームは「Direct Put」ソースタイプを使用する必要があります。
            - フィールド選択: ログに含めるデータフィールド。
            - 出力形式: ログのフォーマット方法:
                - JSON: コンピュータ処理用に構造化された形式
                - テキスト: プレーンテキスト形式
            - フィールド区切り文字: テキスト形式を使用する場合、これがフィールドを区切る文字になります。

Command line

このセクションのコマンドは、AWS CLI コマンドリファレンスに関連しています。ドキュメントには、コマンドの実行時に使用できるオプションの詳しい説明が記載されています。

ルートサーバーピアログの作成方法:

1. put-delivery-source コマンドを使用します。
    - リクエスト例

```
aws logs put-delivery-source --name "source-rsp-1234567890abcdef0" --
resource-arn "arn:aws:ec2:us-east-1:111122223333:route-server-peer/
rsp-1234567890abcdef0" --log-type "EVENT_LOGS"
```

- レスポンス例

```
{
     "deliverySource": {
        "name": "source-rsp-1234567890abcdef0",
        "arn": "arn:aws:logs:us-east-1:111122223333:delivery-source:source-
rsp-1234567890abcdef0",
        "resourceArns": [
            "arn:aws:ec2:us-east-1:111122223333:route-server-peer/
rsp-1234567890abcdef0"
        ],
        "service": "ec2",
        "logType": "EVENT_LOGS"
    }
}
```

2. put-delivery-destination コマンドを使用します。

- 次の AWS CLI 例は、ルートサーバーログを作成します。ログは、指定されたロググループに配信されます。
- リクエスト例

```
aws logs put-delivery-destination --name "destination-rsp-abcdef01234567890"
 --destination-resource-arn "arn:aws:logs:us-east-1:111122223333:log-group:/
aws/vendedlogs/ec2/route-server-peer/EVENT_LOGS/rsp-abcdef01234567890"
```

- レスポンス例

```
{
     "deliveryDestination": {
        "name": "destination-rsp-abcdef01234567890",
        "arn": "arn:aws:logs:us-east-1:111122223333:delivery-
destination:destination-rsp-abcdef01234567890",
        "deliveryDestinationType": "CWL",
        "deliveryDestinationConfiguration": {
```

```
            "destinationResourceArn": "arn:aws:logs:us-
east-1:111122223333:log-group:/aws/vendedlogs/ec2/route-server-peer/
EVENT_LOGS/rsp-abcdef01234567890"
        }
    }
}
```

3. create-delivery コマンドを使用します。

   - リクエスト例

```
aws logs create-delivery --delivery-source-name "source-rsp-1234567890abcdef0"
 --delivery-destination-arn "arn:aws:logs:us-east-1:111122223333:delivery-
destination:destination-rsp-abcdef01234567890"
```

   - レスポンス例

```
{
     "delivery": {
        "id": "1234567890abcdef0",
        "arn": "arn:aws:logs:us-
east-1:111122223333:delivery:1234567890abcdef0",
        "deliverySourceName": "source-rsp-1234567890abcdef0",
        "deliveryDestinationArn": "arn:aws:logs:us-
east-1:111122223333:delivery-destination:destination-rsp-abcdef01234567890",
        "deliveryDestinationType": "CWL",
        "recordFields": [
            "resource_arn",
            "event_timestamp",
            "type",
            "status",
            "message"
        ]
    }
}
```

使用開始チュートリアル

本チュートリアルでは、VPC で動的ルーティングを有効にするように VPC Route Server を設定および設定するプロセスについて説明します。すべての必要なコンポーネントを作成および設定し、BGP ピアリングを確立して、適切なオペレーションを確認する方法について説明します。本

チュートリアルでは、初期 IAM セットアップからテストやクリーンアップまで、すべてについて説明します。

本チュートリアルを開始する前に、以下を確認してください:

- AWS アカウントへの管理アクセス権。
- 動的ルーティングを有効にするサブネットが 2 つ以上ある VPC
- BGP をサポートし、ルートサーバーのピアデバイスとして機能できるネットワークデバイス (EC2 インスタンスで実行されているファイアウォールなど)
- BGP の概念と AWS ネットワークに関する基本的な知識

ステップは、AWS マネジメントコンソールまたは AWS CLI を使用して完了できます。どちらの方法もステップごとに提供されています。

推定完了時間: 15 ~ 30 分

Steps

- ステップ 1: 必要な IAM ロールのアクセス許可を設定する
- ステップ 2: ルートサーバーを作成する
- ステップ 3: ルートサーバーを VPC に関連付ける
- ステップ 4: ルートサーバーエンドポイントを作成する
- ステップ 5: ルートサーバーの伝播を有効にする
- ステップ 6: ルートサーバーピアを作成する
- ステップ 7: デバイスから BGP セッションを開始する
- ステップ 8: クリーンアップ

ステップ 1: 必要な IAM ロールのアクセス許可を設定する

VPC Route Server を使用するには、使用している IAM ユーザーまたはロールに必要な IAM アクセス許可があることを確認します。以下は、各 API ごとに必要なアクセス許可に対するガイドです。

JSON

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

```
        {
            "Sid": "CreateRouteServer",
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DeleteRouteServer",
            "Effect": "Allow",
            "Action": [
                "sns:DeleteTopic"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreateRouteServerEndpoint",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:CreateSecurityGroup",
                "ec2:DescribeSecurityGroups",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateTags",
                "ec2:DeleteTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DeleteRouteServerEndpoint",
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteSecurityGroup",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:CreateTags",
                "ec2:DeleteTags"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CreateRouteServerPeer",
```

```
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupIngress"
            ],
            "Resource": "*"
        },
        {
            "Sid": "DeleteRouteServerPeer",
            "Effect": "Allow",
            "Action": [
                "ec2:RevokeSecurityGroupIngress"
            ],
            "Resource": "*"
        }
    ]
}
```

ステップ 2: ルートサーバーを作成する

このセクションのステップを実行してルートサーバーを作成します。

ルートサーバーコンポーネントは、Forwarding Information Base (FIB) の IPv4 または IPv6 ルートで VPC およびインターネットゲートウェイルートテーブルを更新します。ルートサーバーは、単一の FIB と Routing Information Base (RIB) を表します。

AWS Management Console

ルートサーバーを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [仮想プライベートクラウド] で、[ルートサーバー] を選択します。
3. [ルートサーバー] ページで、[ルートサーバーを作成] を選択します。
4. [ルートサーバーを作成] ページで、次の設定を行います:

   - [名前] には、ルートサーバーの名前を入力します (例: "my-route-server-01")。名前は 255 文字未満にする必要があります。
   - [Amazon Side ASN] には、BGP ASN 値を入力します。この値は 1 ~ 4294967295 の範囲で指定する必要があります。プライベート ASN は、64512 ~ 65534 (16 ビット ASN) または 4200000000 ~ 4294967294 (32 ビット ASN) の範囲で使用することをお勧めします。

- [ルートを永続化] には、[有効] または [無効] を選択します。このオプションは、すべての BGP セッションが終了した後にルートを維持するかどうかを決定します:
  - 有効にした場合: すべての BGP セッションが終了しても、ルートはルートサーバーのルーティングデータベースに保持されます
  - 無効にした場合: すべての BGP セッションが終了すると、ルーティングデータベースからルートが削除されます
- 永続的ルートを有効にした場合は、[永続時間] に 1～5 分の値を入力します。この時間は、BGP が再確立されてルートが非永続化されるまで、ルートサーバーが待機する時間を指定します。たとえば、1 分に設定すると、BGP を再確立してから、ルートサーバーが通常の機能を再開するまで、デバイスは 1 分間でルートを再学習してアドバタイズします。通常は 1 分で十分ですが、BGP ネットワークがすべてのルートを完全に再確立して再学習するのにもっと時間が必要な場合は、最大 5 分まで設定できます。
- (オプション) BGP 状態変更の SNS 通知を有効にするには、[SNS 通知を有効にする] スイッチをオンにします。SNS 通知を有効にすると、ルートサーバーピアの BGP または BFD セッションステータスの変更と、AWS がプロビジョニングした SNS トピックへのルートサーバーエンドポイントのメンテナンス通知が保持されます。これらの通知の詳細については、以下の SNS 通知の詳細テーブルを参照してください。

5. (オプション) ルートサーバーにタグを追加するには、[タグ – オプション] セクションまでスクロールし、[新しいタグを追加] を選択します。各タグのキーとオプションの値を入力します。最大 50 個のタグを追加できます。
6. 設定を確認し、[ルートサーバーを作成] を選択します。
7. ルートサーバーが作成されるまで待機します。完了すると、[ルートサーバー] ページにリダイレクトされ、新しいルートサーバーのステータスが利用可能と表示されます。

Command line

次の手順を使用して、VPC 内の動的ルーティングを管理するための新しいルートサーバーを作成します。

`--amazon-side-asn` に、BGP ASN 値を入力します。この値は 1～4294967295 の範囲で指定する必要があります。プライベート ASN は、64512～65534 (16 ビット ASN) または 4200000000～4294967294 (32 ビット ASN) の範囲で使用することをお勧めします。

1. コマンド:

```
aws ec2 create-route-server --amazon-side-asn 65000
```

レスポンス:

```
{
    "RouteServer": {
        "RouteServerId": "rs-1",
        "AmazonSideAsn": 65000,
        "State": "pending"
    }
}
```

2. ルートサーバーが利用可能になるまで待機します。

コマンド:

```
aws ec2 describe-route-servers
```

レスポンス:

```
{
    "RouteServer": {
        "RouteServerId": "rs-1",
        "AmazonSideAsn": 65000,
        "State": "available"
    }
}
```

SNS 通知の詳細

次の表は、Amazon VPC Route Server が Amazon SNS を使用して送信するメッセージの詳細を示しています:

標準フィールド		メッセージ属性 (メタデータ)			
メッセージ	送信時	timestamp	eventCode	routeServerEndpointId	affectedRouteServerPeerIds
ルートサーバーエンドポイント [ENDPOINT ID] は現在メンテナンス中です。BFD セッションと BGP セッションが影響を受ける可能性があります。	ルートサーバーエンドポイントのメンテナンス	形式: 2025-02-17T15:55:00Z	ROUTE_SERVER_ENDPOINT_MAINTENANCE	影響を受けるエンドポイント ID	影響を受けるピア ID のリスト
メッセージ	送信時	timestamp	eventCode	routeServerPeerId	newBgpStatus
Route Server Peer [PEER ID] の BGP が [UP/DOWN] になりました。	ルートサーバーのピア BGP ステータスの変更	形式: 2025-02-17T15:55:00Z	ROUTE_SERVER_PEER_BGP_STATUS_CHANGE	影響を受けるピア ID	UP または DOWN
メッセージ	送信時	timestamp	eventCode	routeServerPeerId	newBfdStatus
Route Server Peer [PEER ID] の BFD が	ルートサーバーピア	形式: 2025-02-17T15:55:00Z	ROUTE_SERVER_PEER_	影響を受けるピア ID	UP または DOWN

標準フィールド		メッセージ属性 (メタデータ)			
[UP/DOWN] になりました。	BFD ステータスの変更		BFD_STATUS_CHANGE		

ステップ 3: ルートサーバーを VPC に関連付ける

このセクションのステップを実行して、ルートサーバーを VPC に関連付けます。

ルートサーバーの関連付けは、ルートサーバーと VPC の間で確立された接続です。これは、ルートサーバーが VPC 内のアプライアンスと連携できるようにする基本的な設定ステップです。

ルートサーバーの関連付けを作成する場合:

- ルートサーバーを特定の VPC にリンクします。
- これにより、ルートサーバーは VPC のサブネット内のルートテーブルとやり取りできます。
- これにより、ルートサーバーは関連付けられた VPC 内でルートを受信して伝播できます。
- これにより、ルートサーバーが動作できる範囲が確立されます。

ルートサーバーの関連付けの主な側面:

- 各ルートサーバーは、1 つの VPC に関連付けることができます。各 VPC には、デフォルトで最大 5 つまで個別のルートサーバーの関連付けを設定できます。クォータの詳細については、「Route server quotas」を参照してください。
- ルートサーバーがルートを管理する前に、関連付けを作成する必要があります。
- 関連付けをモニタリングして、その状態 (未関連付けや関連付け済みなど) を追跡できます。
- ルートサーバーがその VPC で動作しなくなった場合は、関連付けを削除 (関連付け解除) できます。

AWS Management Console

VPC をルートテーブルに関連付ける

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。

2. ナビゲーションペインの [仮想プライベートクラウド] で、[ルートサーバー] を選択します。
3. VPC に関連付けるルートサーバーを選択します。
4. [関連付けタブ] で、[ルートサーバーの関連付け] を選択します。
5. [ルートサーバーの関連付け] ダイアログボックスで、次の操作を行います:
   - [ルートサーバー ID] フィールドには、選択したルートサーバーが自動的に入力されます。
   - [VPC ID] には、ドロップダウンリストから関連付ける VPC を選択します。
6. [ルートサーバーの関連付け] を選択します。
7. 関連付けが完了するのを待機します。完了すると、[関連付け] タブに [状態] が [関連付け済み] として表示されます。

Command line

VPC をルートサーバーに関連付けるには、次の手順を実行します。

1. コマンド:

```
aws ec2 associate-route-server --route-server-id rs-1 --vpc-id vpc-1
```

   レスポンス:

```
{
    "RouteServerAssociation": {
        "RouteServerId": "rs-1",
        "VpcId": "vpc-1",
        "State": "associating"
    }
}
```

2. 関連付けが完了するのを待機します。

   コマンド:

```
aws ec2 get-route-server-associations --route-server-id rs-1
```

   レスポンス:

```
{
```

```
    "RouteServerAssociation": {
        "RouteServerId": "rs-1",
        "VpcId": "vpc-1",
        "State": "associated"
    }
}
```

ステップ 4: ルートサーバーエンドポイントを作成する

このセクションのステップを実行して、ルートサーバーエンドポイントを作成します。冗長性を確保するために、サブネットごとに 2 つのエンドポイントを作成します。

ルートサーバーエンドポイントとは、ルートサーバーと BGP ピア間の BGP (ボーダーゲートウェイプロトコル) 接続を容易にするサブネット内の AWS マネージドコンポーネントです。

ルートサーバーエンドポイントは、ネットワークデバイスがルートサーバーとの BGP セッションを確立する「コンタクトポイント」です。ルートサーバー自体がルーティングの決定とルート伝播を管理するのに対して、これらは実際に BGP 接続を処理するコンポーネントです。

> **Note**
>
> ルートサーバーエンドポイントには、1 時間あたり 0.75 USD が課金されます。

AWS Management Console

ルートサーバーエンドポイントを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [仮想プライベートクラウド] で、[ルートサーバー] を選択します。
3. エンドポイントを作成するルートサーバーを選択します。
4. 下部のペインで、[ルートサーバーエンドポイント] タブを選択します。
5. [ルートサーバーエンドポイントを作成] を選択します。
6. [ルートサーバーエンドポイントを作成] ページで、次の設定を行います:
   - [名前] に、エンドポイントのわかりやすい名前を入力します。
   - [ルートサーバー] で、正しいルートサーバーが選択されていることを確認します。
   - [サブネット] で、エンドポイントを作成するサブネットを選択してください。

7. (オプション) ルートサーバーエンドポイントにタグを追加するには、[タグ – オプション] セクションまでスクロールし、[新しいタグを追加] を選択します。各タグのキーとオプションの値を入力します。
8. 設定を確認したら、[ルートサーバーエンドポイントを作成] を選択してください。
9. エンドポイントが作成されるまで待ちます。完了すると、成功メッセージが表示されます。
10. ステップ 5～9 を繰り返して、同じサブネットに別の名前を使用して 2 番目のエンドポイントを作成します。
11. ルートサーバーエンドポイントが必要なサブネットごとに、ステップ 5～10 を繰り返します。
12. エンドポイントを作成したら、ルートサーバーの [ルートサーバーエンドポイント] タブに戻ります。
13. 各サブネットごとに 2 つのエンドポイントがリストされていることを確認します。
14. 各エンドポイントの [状態] が利用可能であることを確認します。

Command line

以下の手順に従って、ルートサーバーエンドポイントを作成します。

1. コマンド:

```
aws ec2 create-route-server-endpoint --route-server-id rs-1 --subnet-id subnet-1
```

レスポンス:

```
{
    "RouteServerEndpoint": {
        "RouteServerId": "rs-1",
        "RouteServerEndpointId": "rse-1",
        "VpcId": "vpc-1",
        "SubnetId": "subnet-1",
        "State": "pending"
    }
}
```

2. 作成後にエンドポイントが完全に利用可能になるまで数分待機する必要がある場合があります。

コマンド:

```
aws ec2 describe-route-server-endpoints
```

レスポンス:

```
{
    "RouteServerEndpoint": {
        "RouteServerId": "rs-1",
        "RouteServerEndpointId": "rse-1",
        "VpcId": "vpc-1",
        "SubnetId": "subnet-1",
        "EniId": "eni-123",
        "EniAddress": "10.1.2.3",
        "State": "available"
    }
}
```

ステップを繰り返して、異なる名前を使用して同じサブネットに 2 番目のエンドポイントを作成し、ルートサーバーエンドポイントが必要なサブネットごとにエンドポイントを作成します。

ステップ 5: ルートサーバーの伝播を有効にする

ルートサーバーの伝播を有効にするには、このステップを完了します。

有効すると、ルートサーバーの伝播は、指定したルートテーブルの FIB にルートをインストールします。ルートサーバーは IPv4 および IPv6 のルート伝達に対応しています。

ルートサーバーの伝播とは、ルートテーブルの更新を自動化するメカニズムです。ルートテーブルを手動で更新する代わりに、ルートサーバーは FIB からのルートを使用して、構成済みのルートテーブルに適切なルートを自動的に伝播します。

ルートサーバーの伝播の主な側面:

- 設定
  - ルートサーバーを特定のルートテーブルにリンクします
  - 動的ルート更新を受信するルートテーブルを決定します
  - ルートテーブルごとに有効または無効にすることができます
- 機能

- BGP ピアから学習したルートでルートテーブルを自動的に更新します
- BGP 属性に基づいて利用可能な最適なルートを伝播します
- 指定されたルートテーブル間でルートの一貫性を維持します
- ネットワーク条件が変更されたときにルートを動的に更新します

- 状態
  - 有効にできます (ルートは伝播されています)
  - 無効にできます (ルートは伝播されません)

AWS Management Console

ルートサーバーの伝播を有効にするには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 伝播を有効にするルートサーバーを選択します。
3. ルートサーバーの詳細パネルの [伝播] タブを選択します。
4. [伝播を有効にする] を選択します。
5. [伝播を有効にする] ダイアログで、次の操作を行います:
   - [ルートサーバー ID] は事前入力されます。
   - [ルートテーブル] で、新しく伝播されたルートのドロップダウンメニューから送信先ルートテーブルを選択します。
6. [伝播を有効にする] を選択して確認します。
7. 伝播ステータスが [伝播] リストで「利用可能」に変わるまで待ちます。
8. 選択したルートテーブルが利用可能状態になっている [伝播] リストに表示されていることを確認します。

Command line

ルートサーバーの伝播を有効にするには、次の手順に従います。

1. コマンド:

```
aws ec2 enable-route-server-propagation --route-table-id rtb-1 --route-server-id
 rs-1
```

レスポンス:

```
{
    "RouteServerRoutePropagation": {
        "RouteServerId": "rs-1",
        "RouteTableId": "rtb-1",
        "State": "pending"
    }
}
```

2. 伝播の状態が利用可能に変わるまで待機します。

コマンド:

```
aws ec2 get-route-server-propagations --route-server-id rs-1
```

レスポンス:

```
{
    "RouteServerRoutePropagation": {
        "RouteServerId": "rs-1",
        "RouteTableId": "rtb-1",
        "State": "available"
    }
}
```

ステップ 6: ルートサーバーピアを作成する

ルートサーバーピアとは、ルートサーバーエンドポイントと AWS にデプロイされたデバイス (ファイアウォール機器や EC2 インスタンスで実行されているその他のネットワークセキュリティ機能など) との間のセッションです。デバイスは、以下の条件を満たす必要があります:

- VPC に Elastic Network Interface がある
- BGP (ボーダーゲートウェイプロトコル) をサポート
- BGP セッションを開始できる

> **Note**
>
> 冗長性を確保するために、ルートサーバーエンドポイントごとに 1 つのルートサーバーピアを作成することをお勧めします。

AWS Management Console

ルートサーバーピアを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションパスで、[VPC] > [ルートサーバーピア] > [ルートサーバーピアを作成] の順で選択します。
3. [詳細] で、以下を設定します:
   - 名前: ルートサーバーピアの名前を入力します (最大 255 文字)。例: my-route-server-peer-01
   - ルートサーバーエンドポイント ID: ドロップダウンからルートサーバーエンドポイントを選択します。オプションで、[ルートサーバーエンドポイントを作成] を選択して新しいエンドポイントを作成します。
   - ピアアドレス: ピアの IPv4 アドレスを入力します。有効な IP アドレスである必要があります。ピアアドレスは、ルートサーバーエンドポイントから到達可能である必要があります。
   - ピア ASN: BGP ピアの ASN (自律システム番号) を入力します。値は 1 ~ 4294967295 の範囲で指定する必要があります。ASN は通常、プライベート範囲 (16 ビットの場合は 64512-65534、32 ビットの場合は 4200000000-4294967294) を使用する必要があります
   - ピアの活性検出:
     - BGP キープアライブ (デフォルト): 標準 BGP キープアライブメカニズム
     - BFD: Bidirectional Forwarding Detection によるフェイルオーバーの高速化
   - (オプション) [タグ] で、[新しいタグを追加] を選択してキーと値のペアタグを追加します。タグにより、AWS リソースの識別と追跡が可能になります。
4. 設定を確認し、[ルートサーバーピアを作成] を選択します。

Command line

次の手順を使用してルートサーバーピアを作成します。

1. コマンド:

```
aws ec2 create-route-server-peer --route-server-endpoint-id rse-1 --peer-address
 10.0.2.3 --bgp-options PeerAsn=65001,PeerLivenessDetection=bfd
```

レスポンス:

レスポンスでは、状態値は pending|available|deleting|deleted にすることができます。

```
{
    "RouteServerPeer": {
        "RouteServerPeerId": "rsp-1",
        "RouteServerId": "rs-1",
        "VpcId": "vpc-1",
        "SubnetId": "subnet-1",
        "State": "pending",
        "EndpointEniId": "eni-2,
        "EndpointEniAddress": "10.0.2.4",
        "PeerEniId": "eni-1",
        "PeerAddress": "10.0.2.3",
        "BgpOptions": {
            "PeerAsn": 65001,
   "PeerLivenessDetection": "bfd"
        },
        "BgpStatus": {
            "Status": "Up"
        }
    }
}
```

2. 伝播の状態が利用可能に変わるまで待機します。

コマンド:

```
aws ec2 describe-route-server-peers
```

レスポンス:

```
{
    "RouteServerPeer": {
```

```
        "RouteServerPeerId": "rsp-1",
        "RouteServerId": "rs-1",
        "VpcId": "vpc-1",
        "SubnetId": "subnet-1",
        "State": "available",
        "EndpointEniId": "eni-2,
        "EndpointEniAddress": "10.0.2.4",
        "PeerEniId": "eni-1",
        "PeerAddress": "10.0.2.3",
        "BgpOptions": {
            "PeerAsn": 65001,
   "PeerLivenessDetection": "bfd"
        },
        "BgpStatus": {
            "Status": "down"
        }
    }
}
```

ステップ 7: デバイスから BGP セッションを開始する

ルートサーバーピアのステータスが利用可能になったら、ルートサーバーエンドポイントとの BGP セッションを開始するようにワークロードを設定します。

サブネット内のデバイスから BGP セッションを開始することは、本ガイドの範囲外です。ルートサーバーエンドポイントは BGP セッションを開始しません。

VPC ルートサーバー機能が動作していることを確認するには、ルートテーブルにルートサーバーが伝播した最適なルートがあることを確認します。

ステップ 8: クリーンアップ

チュートリアルの構築部分が完了しました。このセクションのステップを完了して、作成した VPC Route Server コンポーネントを削除します。

7.1: デバイスで BGP アドバタイズを取り消す

サブネット内のデバイスでの BGP アドバタイズの取り消しは、本ガイドの範囲外です。必要に応じて、BGP 設定については、サードパーティーベンダーを参照してください。

7.2: ルートサーバーの伝播を無効にする

ルートサーバーの伝播を無効にするには、次の手順に従います。

AWS Management Console

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 伝播を無効にするルートサーバーを選択します。
3. [アクション > ルートサーバーを変更] の順で選択します。
4. ルートサーバーの詳細パネルの [伝播] タブを選択します。
5. 無効にする伝播を選択し、[伝播を無効にする] を選択します。
6. ダイアログボックスで、[ルートサーバーの伝播を無効にする] を選択します。

Command line

1. 伝播を無効にします:

```
aws ec2 disable-route-server-route-propagation --route-table-id rtb-1 --route-server-id rs-1
```

2. 伝播が削除されたことを確認します:

```
aws ec2 get-route-server-route-propagations --route-server-id rs-1 [--route-table-id rtb-1]
```

7.3: ルートサーバーピアを削除する

以下の手順に従って、ルートサーバーピアを削除します。

AWS Management Console

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションパスで、[ルートサーバー] > [ルートサーバーピア]の順で選択します。
3. ルートサーバーピアを選択します。
4. [アクション] > [ルートサーバーピアを削除] の順で選択します。

Command line

1. ピアを削除します:

```
aws ec2 delete-route-server-peer --route-server-peer-id rsp-1
```

2. 削除を確定します:

```
aws ec2 describe-route-server-peers [--route-server-peer-ids rsp-1] [--filters
 Key=RouteServerId|RouteServerEndpointId|VpcId]
```

7.4: ルートサーバーエンドポイントを削除する

次の手順に従って、ルートサーバーエンドポイントを削除します。

AWS Management Console

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. エンドポイントを削除するルートサーバーを選択します。
3. [ルートサーバーエンドポイント] を選択します。
4. エンドポイントを選択し、[アクション] > [ルートサーバーエンドポイントを削除] の順に選択します。
5. 「削除」と入力し、[削除] を選択します。

Command line

1. エンドポイントを記述します:

```
aws ec2 describe-route-server-endpoints
```

2. ルートサーバーエンドポイントを削除します:

```
aws ec2 delete-route-server-endpoint --route-server-endpoint-id rse-1
```

3. エンドポイントが削除されていることを確認します:

```
aws ec2 describe-route-server-endpoints [--route-server-endpoint-ids rsp-1] [--
filters Key=RouteServerId|VpcId|SubnetId]
```

7.5: VPC からルートサーバーの関連付けを解除する

次の手順に従って、ルートサーバーと VPC の関連付けを解除します。

AWS Management Console

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 関連付けを解除するルートサーバーを選択します。
3. [関連付け] を選択します。
4. [ルートサーバーの関連付けを解除] を選択します。
5. 変更点を確認し、[ルートサーバーの関連付けを解除] を選択します。

Command line

1. VPC からルートサーバーの関連付けを解除します。

```
aws ec2 disassociate-route-server --route-server-id rs-1 --vpc-id vpc-1
```

2. 関連付け解除を確認します:

```
aws ec2 get-route-server-associations --route-server-id rs-1
```

7.6 ルートサーバーを削除する

以下の手順に従って、ルートサーバーを削除します。

AWS Management Console

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 削除するルートサーバーを選択します。
3. [アクション] > [ルートサーバーを削除] の順で選択します。
4. 「削除」と入力し、[削除] を選択します。

Command line

1. ルートサーバーを削除します:

```
aws ec2 delete-route-server --route-server-id rs-1
```

2. 削除を確定します:

```
aws ec2 describe-route-servers [--route-server-ids rs-1] [--filters Key=VpcId]
```

Amazon VPC Route Server のチュートリアルが完了しました。

VPC 内の到達可能性問題をトラブルシューティングする

Reachability Analyzer は静的な設定分析ツールです。Reachability Analyzer を使用して、VPC 内の 2 つのリソース間のネットワーク到達可能性を分析およびデバッグできます。Reachability Analyzer は、これらのリソースに到達可能な場合は、リソース間にある仮想パスのホップバイホップの詳細を生成し、そうでない場合はブロッキングコンポーネントを識別します。例えば、見失ったルートテーブルルート、または正しく設定されていないルートテーブルルートを特定できます。

詳細については、「Reachability Analyzer Guide」(到達可能性アナライザーガイド) を参照してください。

ミドルボックスルーティングウィザード

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 “ミドルボックスシナリオ”」を参照してください。

内容

- ミドルボックスルーティングウィザードの前提条件
- VPC 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
- 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
- ミドルボックスシナリオ

ミドルボックスルーティングウィザードの前提条件

確認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”。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。

VPC 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

ミドルボックスのルーティングウィザードは Amazon VPC コンソールで利用できます。

内容

- 1. 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
- 2. ミドルボックスルートの変更
- 3. 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

1. 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. ナビゲーションペインで、[Your VPCs (お使いの VPC)] 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. [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。

2. 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. ナビゲーションペインで、[Your VPCs (お使いの VPC)] 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. [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。

3. 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。

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。

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、[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”」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、ルートの送信先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。

- ミドルボックスルーティングを使用してセキュリティアプライアンス経由でルーティングする場合、検査後の送信元と最終的な送信先の間のセキュリティグループの参照はサポートされません。

ミドルボックスシナリオ

Amazon Virtual Private Cloud (VPC) には、仮想ネットワーク内のトラフィックのルーティングをカスタマイズし制御するための、幅広いネットワーク機能が用意されています。こうした機能の 1 つにミドルボックスルーティングウィザードがあります。これを使用すると、VPC に出入りするトラフィックのルーティングパスを詳細に制御することができます。

検査、モニタリング、最適化のためにトラフィックをセキュリティアプライアンス、ロードバランサー、またはその他ネットワークデバイスにリダイレクトする必要がある場合は、ミドルボックスルーティングウィザードを使用することでプロセスを簡素化できます。このウィザードにより、指定されたトラフィックをリダイレクトするために必要な、ルートテーブルとルート (ホップ) が自動で作成されるため、複雑なルーティング構成をマニュアルで設定する手間を省けます。

ミドルボックスルーティングウィザードは複数のシナリオに対応しています。例えば、特定のサブネット宛てのトラフィックを検査したり、ミドルボックストラフィックのルーティングを設定し VPC 全体を検査したり、特定のサブネット間のトラフィックを選択的に検査したりできます。トラフィックルーティングを細かく制御することで、高度なセキュリティポリシーを実装したり、一元化されたネットワークモニタリングを有効にしたり、クラウドベースのアプリケーションのパフォーマンスを最適化したりすることができます。

次の例で、ミドルボックスルーティングウィザードのシナリオを説明しています。

内容

- サブネット宛てのトラフィックを検査する
- VPC でミドルボックストラフィックのルーティングと検査を設定する
- サブネット間のトラフィックを検査する

サブネット宛てのトラフィックを検査する

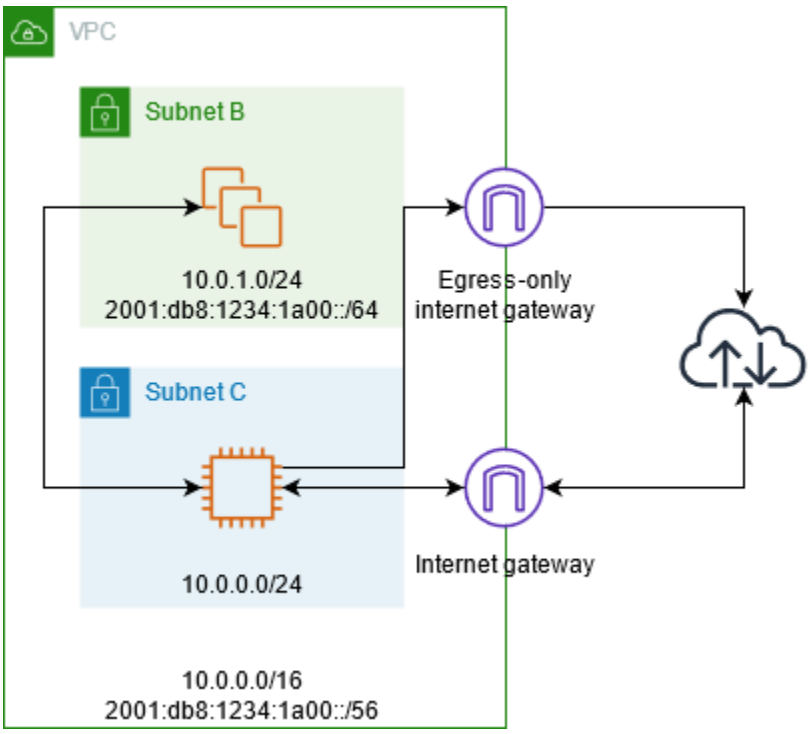
インターネットゲートウェイを介して VPC にトラフィックが着信しており、EC2 インスタンスにインストールされたファイアウォールアプライアンスを使用して、サブネットを送信先 (サブネット B など) とするすべてのトラフィックを検査するシナリオを考えてみます。ファイアウォールアプライアンスは、VPC のサブネット B (サブネット C) とは別のサブネットにある EC2 インスタンスにインストールおよび設定する必要があります。その後、ミドルボックスルーティングウィザードを使用して、サブネット B とインターネットゲートウェイ間のトラフィックのルートを設定できます。

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。

- 次のルートテーブルを作成します。
  - インターネットゲートウェイのルートテーブル
  - 宛先サブネットのルートテーブル
  - ミドルボックスサブネットのルートテーブル
- 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 [Source] (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[Middlebox] (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[Destination] (宛先)) 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次のルートを追加します。

ルーティング先	Target	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	*appliance-eni*	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	*appliance-eni*	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。

- キーは「Origin」で、値は「ミドルボックスウィザード」です
- キーは「date_created」で、値は作成時刻です (「2021-02-18T22:25:49.137Z」など)。

宛先サブネットのルートテーブル

宛先サブネット (図のサブネット B) のルートテーブルに次のルートを追加します。

ルーティング先	Target	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	*appliance-eni*	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	*appliance-eni*	インターネット宛ての IPv6 トラフィックをミドルボックスにルーティングする

ミドルボックスサブネット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。

- キーは「Origin」で、値は「ミドルボックスウィザード」です
- キーは「date_created」で、値は作成時刻です (「2021-02-18T22:25:49.137Z」など)。

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 (例のサブネット C) のルートテーブルに次のルートを追加します。

ルーティング先	Target	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

ルーティング先	Target	目的
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

宛先サブネット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。

- キーは「Origin」で、値は「ミドルボックスウィザード」です
- キーは「date_created」で、値は作成時刻です (「2021-02-18T22:25:49.137Z」など)。

VPC でミドルボックストラフィックのルーティングと検査を設定する

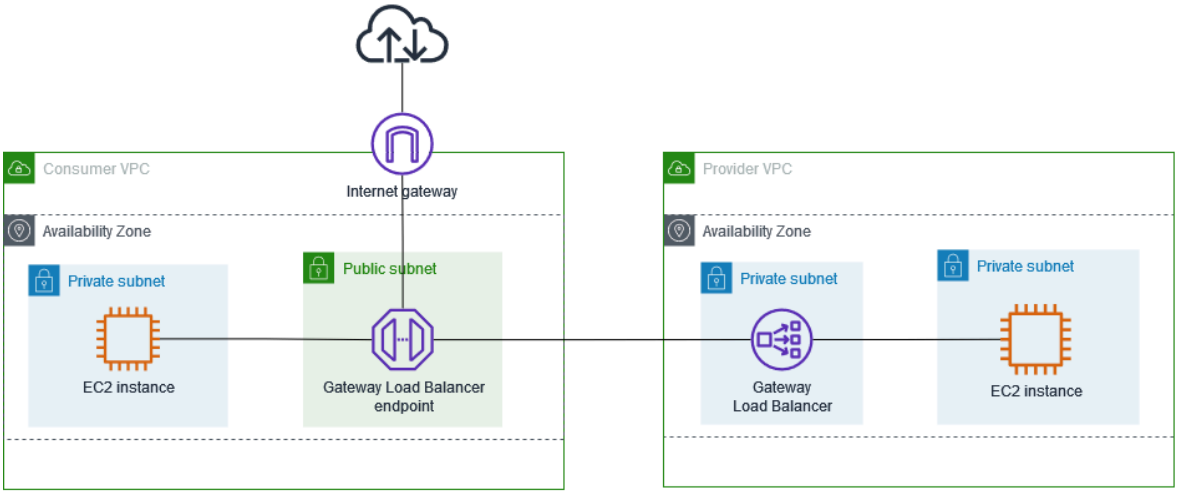
Gateway Load Balancer の背後に設定されたセキュリティアプライアンスのフリートを使用して、インターネットゲートウェイから VPC に入り、サブネットを送信先とするトラフィックを検査するシナリオを考えてみます。サービスコンシューマー VPC の所有者は、VPC 内のサブネットに Gateway Load Balancer エンドポイントを作成します (エンドポイントネットワークインターフェイスで表されます)。インターネットゲートウェイを経由して VPC に入るすべてのトラフィックは、まず検査のために Gateway Load Balancer エンドポイントにルーティングされ、その後にアプリケーションサブネットにルーティングされます。同様に、アプリケーションサブネットから出るすべてのトラフィックは、まず検査のために 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。



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、次のルートがあります。

ルーティング先	Target	目的
####### VPC CIDR	ローカル	ローカルルート
############ CIDR	*endpoint-id*	アプリケーションサブネット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。

- キーは「Origin」で、値は「ミドルボックスウィザード」です

- キーは「date_created」で、値は作成時刻です (「2021-02-18T22:25:49.137Z」など)。

アプリケーションサブネットのルートテーブル

アプリケーション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。

ルーティング先	Target	目的
####### VPC CIDR	ローカル	ローカルルート
0.0.0.0/0	*endpoint-id*	アプリケーションサーバーからのトラフィックはを Gateway Load Balancer エンドポイントにルーティングされてから、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。

- キーは「Origin」で、値は「ミドルボックスウィザード」です
- キーは「date_created」で、値は作成時刻です (「2021-02-18T22:25:49.137Z」など)。

プロバイダーサブネットのルートテーブル

プロバイダー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。

ルーティング先	Target	目的
###### VPC CIDR	ローカル	ローカルルート インターネットから発信されるトラフィックがアプリケーションサーバーにルーティングされるようにします。
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

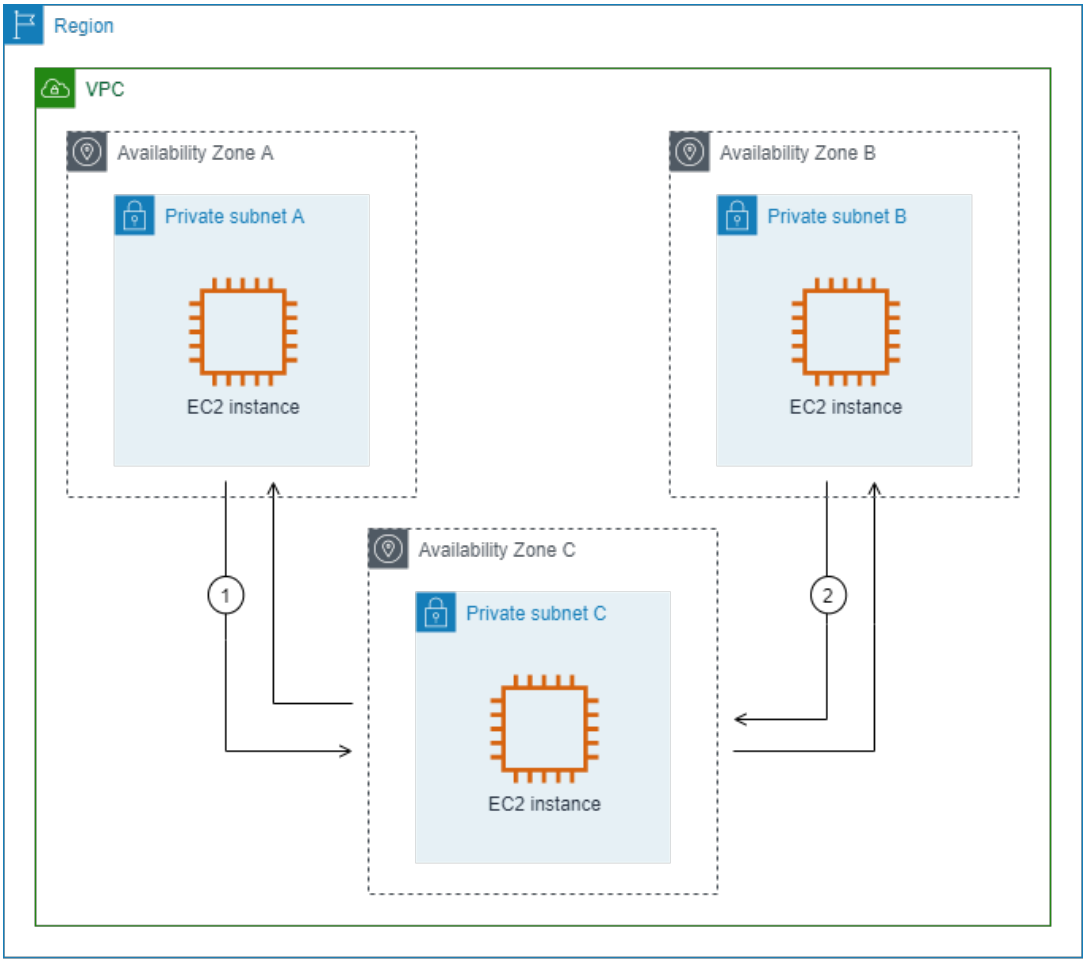
ミドルボックスルーティングウィザードを使用すると、次のタグがルートテーブルに関連付けられます。

- キーは「Origin」で、値は「ミドルボックスウィザード」です
- キーは「date_created」で、値は作成時刻です (「2021-02-18T22:25:49.137Z」など)。

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、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。

サブネット C の 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次の図に示します。このアプライアンスは、サブネット A からサブネット B (「1」を参照) およびサブネット B からサブネット A (「2」を参照) に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テーブルを使用します。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。

サブネット A のカスタムルートテーブル

サブネット A のルートテーブルには、次のルートがあります。

ルーティング先	Target	目的
VPC CIDR	ローカル	ローカルルート
Subnet B CIDR	*appliance-eni*	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

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。

- キーは「Origin」で、値は「ミドルボックスウィザード」です
- キーは「date_created」で、値は作成時刻です (「2021-02-18T22:25:49.137Z」など)。

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。

ルーティング先	Target	目的
VPC CIDR	ローカル	ローカルルート
Subnet A CIDR	*appliance-eni*	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

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。

- キーは「Origin」で、値は「ミドルボックスウィザード」です
- キーは「date_created」で、値は作成時刻です (「2021-02-18T22:25:49.137Z」など)。

メインルートテーブル

Subnet C は、メインルートテーブルを使用します。メインルートテーブルには以下のルートがあります。

ルーティング先	Target	目的
VPC CIDR	ローカル	ローカルルート

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。

- キーは「Origin」で、値は「ミドルボックスウィザード」です
- キーは「date_created」で、値は作成時刻です (「2021-02-18T22:25:49.137Z」など)。

サブネットを削除する

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。サブネットにネットワークインターフェイスが含まれている場合は、そのサブネットを削除できません。例えば、サブネットを削除する前に、サブネット内のインスタンスを終了する必要があります。

サブネットを削除すると、そのサブネットに関連付けられた CIDR ブロックは VPC の使用可能な IP アドレスプールに返されます。つまり、そのサブネットの CIDR 範囲に含まれる IP アドレスは、同じ VPC 内の他のサブネットまたはリソースに再割り当てすることができます。

サブネットを削除しても、そのサブネット内にあるリソースは自動では削除されませんのでご注意ください。サブネットの削除を続けるときは、先に EC2 インスタンスを終了し、ネットワークインターフェイスを削除し、そのサブネットに関連付けられている他のリソースを削除します。

コンソールを使用してサブネットを削除するには

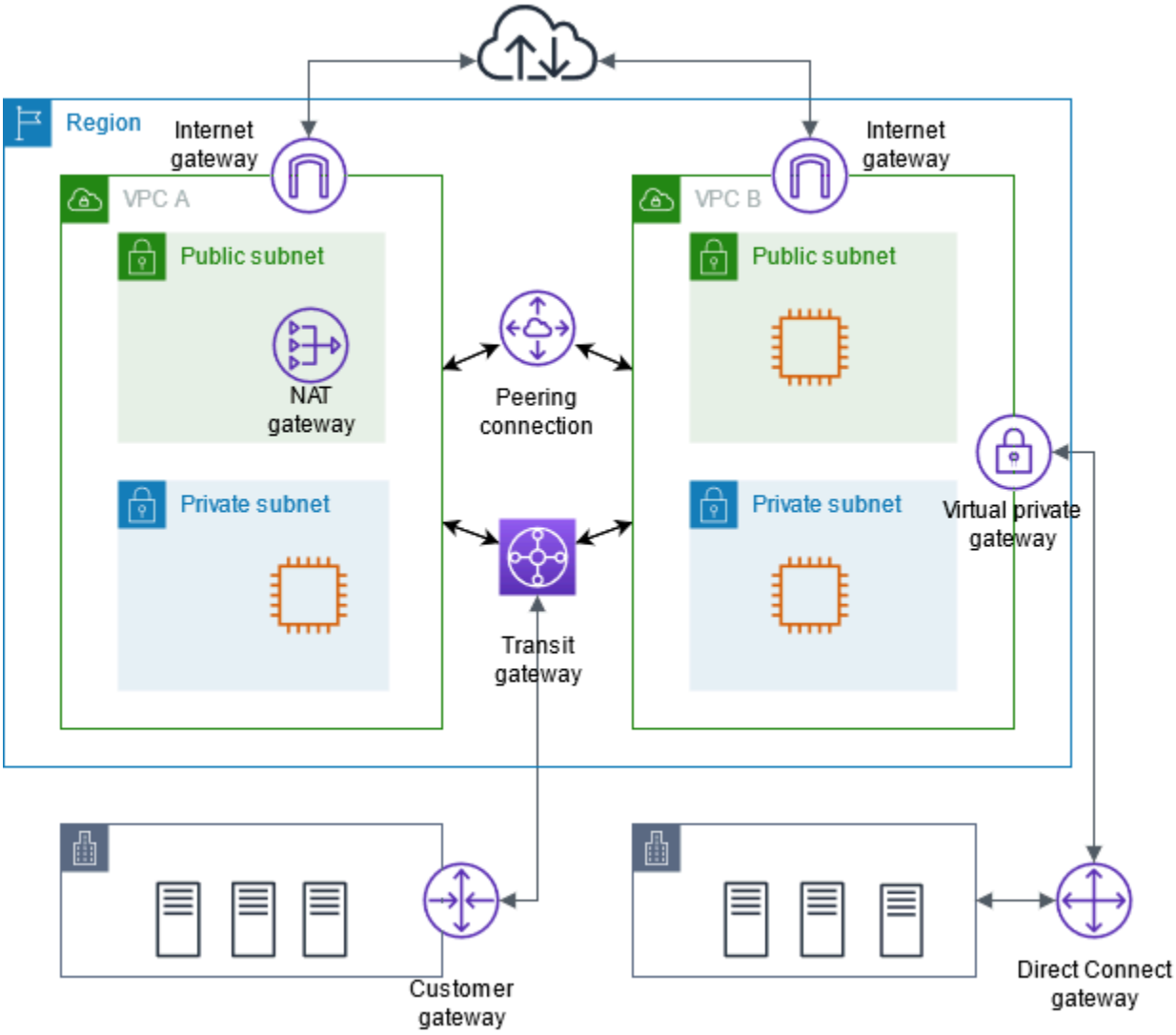
1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. サブネットのすべてのインスタンスを終了します。詳細については、「Amazon EC2 ユーザーガイド」の「インスタンスの終了」を参照してください。
3. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
4. ナビゲーションペインで、[サブネット] を選択してください。
5. サブネットを選択して、[Actions] (アクション)、@Delete subnet] (サブネットの削除) の順に選択します。
6. 確認を求められたら、**`delete`** と入力し、[Delete] (削除) を選択します。

AWS CLI を使用してサブネットを削除するには

delete-subnet コマンドを使用してください。

VPC を他のネットワークに接続する

仮想プライベートクラウド (VPC) は、他の VPC、インターネット、自社のオンプレミスネットワークなど、他のネットワークに接続することができます。



仮想プライベートクラウド (VPC) は、他の VPC、インターネット、自社のオンプレミスネットワークなど、他のネットワークに接続することができます。

次の図は、これらの接続オプションの一部を示しています。VPC A 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接続され、プライベートサブネット内の EC2 インスタンスは、パブリックサブネット内の NAT ゲートウェイを使用してインターネットに接続することができます。VPC B もインターネットに接続されていますが、ダイレクトインターネットゲートウェイを介して接続されるため、パブリックサブネットの EC2 インスタンスはインターネットにアクセスすることが可能です。

さらに、VPC A と VPC B は、VPC ピアリング接続とトランジットゲートウェイを通じて相互に接続されます。トランジットゲートウェイはデータセンターに VPN アタッチメントを使用し、VPC B は同じデータセンターに Direct Connect 接続を使用します。この相互接続により、組織は、クラウドリソースをオンプレミスインフラストラクチャに統合し、ハイブリッドクラウド環境を構築することができます。

VPC を他のネットワークに接続することは、AWS 内でクラウドインフラストラクチャを構築する上で重要な側面の 1 つです。それにより組織は、ネットワーク設定を柔軟に制御することが可能になり、ビジネス要件やセキュリティニーズに合う VPC アーキテクチャを設計できるようになります。こうした接続のオプションは、クラウド内であれオンプレミスであれ、分散型 IT 環境のさまざまな要素間での効率的なデータフローに役立ちます。

AWS には、こうした VPC 接続を可能にする、インターネットゲートウェイ、NAT ゲートウェイ、VPC ピアリング、トランジットゲートウェイ、 を含む幅広いツールや機能が用意されています。Direct Connectこれらの機能を使用することで、既存の IT インフラストラクチャとシームレスに統合した安全で一体的なクラウド環境を構築することができます。

仮想プライベートクラウド (VPC) を他のネットワークに接続できます。他の VPC、インターネット、オンプレミスのネットワークはその一例です。

詳細については、「Amazon Virtual Private Cloud Connectivity Options」(Amazon Virtual Private Cloud の接続オプション) を参照してください。

内容

- インターネットゲートウェイを使用して VPC のインターネットアクセスを有効にする
- Egress-Only インターネットゲートウェイを使用してアウトバウンド IPv6 トラフィックを有効にする
- NAT デバイスを使用してインターネットまたは他のネットワークに接続する
- Elastic IP アドレスを VPC 内のリソースに関連付ける
- トランジットゲートウェイを使用して VPC を他の VPC およびネットワークに接続する
- AWS Virtual Private Network を使用して VPC をリモートネットワークに接続する
- VPC ピアリングを使用して VPC を接続する

インターネットゲートウェイを使用して VPC のインターネットアクセスを有効にする

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。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。

インターネットゲートウェイを使用すると、リソースにパブリック IPv4 アドレスまたは IPv6 アドレスがある場合、パブリックサブネット内のリソース (EC2 インスタンスなど) がインターネットに接続できるようになります。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サブネット内のリソースへの接続を開始できます。例えば、インターネットゲートウェイを使用すると、ローカルコンピュータを使用して AWS の EC2 インスタンスに接続できます。

インターネットゲートウェイは、インターネットルーティング可能なトラフィックの VPC ルートテーブル内のターゲットを提供します。IPv4 を使用した通信の場合、インターネットゲートウェイは、ネットワークアドレス変換 (NAT) も実行します。詳細については、「IP アドレスおよび NAT」を参照してください。

料金

インターネットゲートウェイには課金されませんが、インターネットゲートウェイを使用する EC2 インスタンスにはデータ転送料金が発生します。詳細については、「Amazon EC2 オンデマンド料金」を参照してください。

内容

- インターネットゲートウェイの基本
- サブネットへのインターネットアクセスを追加する
- 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の基本

インターネットゲートウェイを使用するには、インターネットゲートウェイを VPC にアタッチしてルーティングを設定する必要があります。

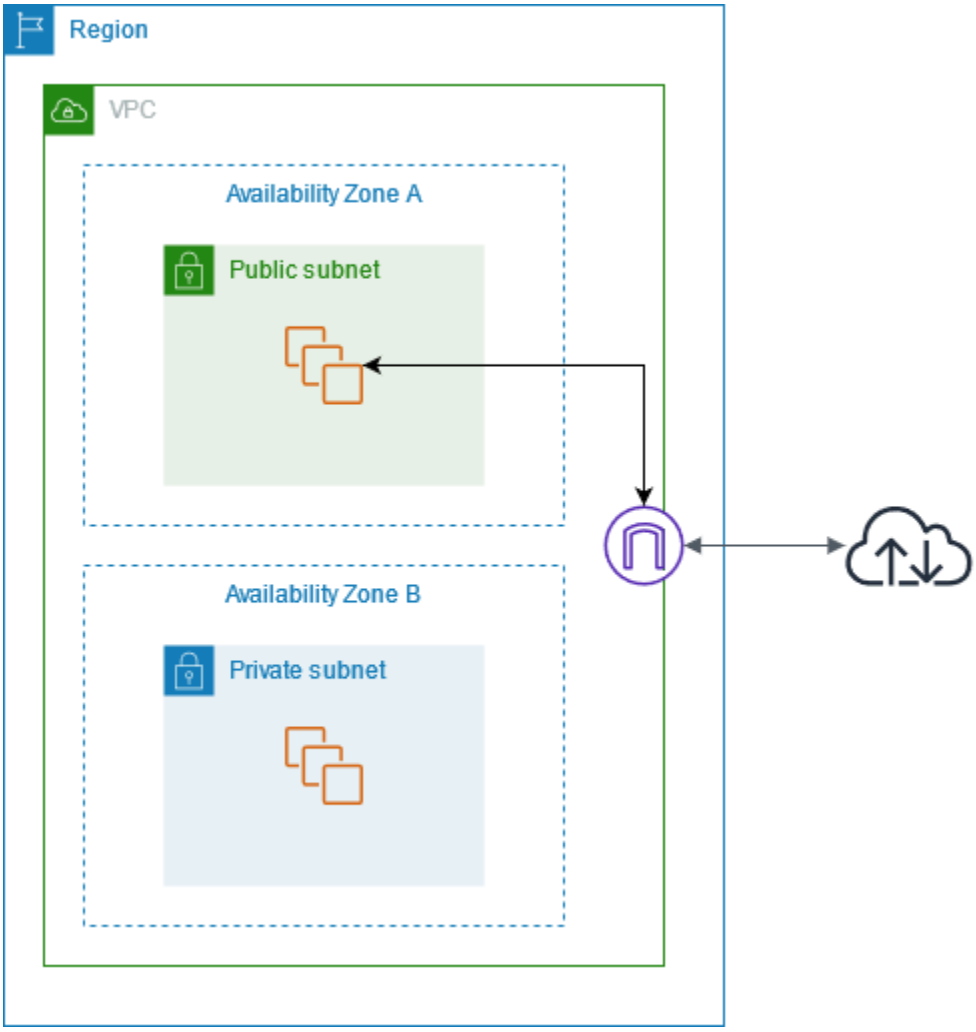
ルーティング設定

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 (IPv4 の場合は 0.0.0.0/0、IPv6 の場合は ::/0) を指定することができます。また、より狭い範囲の IP アドレス (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) にルートを絞り込むこともできます。

インターネットゲートウェイの図

次の図では、すべてのインターネットバウンド IPv4 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がルートテーブルにあるため、アベイラビリティーゾーン A のサブネットは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は、インターネットゲートウェイを経由してインターネットとの通信を有効にするために、パブリック IP アドレスまたは Elastic IP アドレスが必要です。比較として、アベイラビリティーゾーン B 比較として、アベイラビリティーゾーン B のサブネットは、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ないため、プライベートサブネットとなります。インターネットゲートウェイへのルートがないため、プライベートサブネット内のインスタンスは、パブリック IP アドレスが付与されている場合でもインターネットと通信できません。



IP アドレスおよび NAT

IPv4 でインターネット経由の通信ができるようにするには、インスタンスにパブリック IPv4 アドレスが必要です。インスタンスにパブリック IPv4 アドレスが自動的に割り当てられるように VPC を設定するか、インスタンスに Elastic IP アドレスを割り当てることができます。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。

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	いいえ
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	いいえ
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	いいえ
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)

サブネットへのインターネットアクセスを追加する

次に、インターネットゲートウェイを使用してデフォルトでない VPC のサブネットからインターネットへアクセスする方法について説明します。インターネットゲートウェイを作成し、VPC にアタッチして、サブネットのルーティングを設定する必要があります。

サブネットのインターネットアクセスを設定したら、サブネット内のリソースがインターネットにアクセスできることを確認する必要があります。例えば、EC2 インスタンスにはパブリック IPv4 または IPv6 アドレスが必要で、インスタンスのセキュリティグループではインターネットとの間の特定のトラフィックを許可する必要があります。

NAT デバイスを使用して、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こともできます。詳細については、「NAT デバイス」を参照してください。

インターネットアクセスを削除する場合は、VPC からインターネットゲートウェイをデタッチして削除します。詳細については、「the section called “インターネットゲートウェイを削除する”」を参照してください。

タスク

- ステップ 1: インターネットゲートウェイを作成する
- ステップ 2: インターネットゲートウェイを VPC にアタッチする
- ステップ 3: サブネットルートテーブルにルートを追加する

ステップ 1: インターネットゲートウェイを作成する

インターネットゲートウェイを作成するには、以下の手順を実行します。

コンソールを使用してインターネットゲートウェイを作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[Internet gateways] (インターネットゲートウェイ) を選択します。
3. [インターネットゲートウェイの作成] を選択します。
4. (オプション) インターネットゲートウェイの名前を入力します。
5. (オプション) タグを追加するには、[Add new tag] (新しいタグを追加) を選択し、そのタグのキーと値を入力します。
6. [インターネットゲートウェイの作成] を選択します。
7. (オプション) インターネットゲートウェイを今すぐ VPC に接続するには、画面上部のバナーから [VPC に接続] を選択し、使用可能な VPC を選択してから [インターネットゲートウェイに接続] を選択します。それ以外の場合は、別の機会にインターネットゲートウェイを VPC にアタッチできます。

コマンドラインを使用してインターネットゲートウェイを作成するには

- create-internet-gateway（AWS CLI）
- New-EC2InternetGateway（AWS Tools for Windows PowerShell）

ステップ 2: 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を使用するには、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。

コンソールを使用してインターネットゲートウェイを VPC にアタッチするには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[Internet gateways] (インターネットゲートウェイ) を選択します。
3. [インターネットゲートウェイ] のチェックボックスを選択します。
4. アタッチするには、[アクション]、[VPC にアタッチ] の順に選択し、使用可能な VPC を選択して、[インターネットゲートウェイのアタッチ] を選択します。
5. デタッチするには、[アクション]、[VPC からデタッチ] の順に選択し、[インターネットゲートウェイのデタッチ] を選択します。確認を求められたら、[インターネットゲートウェイをデタッチ] を選択します。

コマンドラインを使用してインターネットゲートウェイを VPC にアタッチするには

- attach-internet-gateway（AWS CLI）
- Add-EC2InternetGateway（AWS Tools for Windows PowerShell）

ステップ 3: サブネットルートテーブルにルートを追加する

サブネットのルートテーブルには、インター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するルートが必要です。

コンソールを使用してサブネットルートテーブルを設定するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[Route tables] (ルートテーブル) を選択します。

3. サブネットのルートテーブルを選択します。デフォルトでは、サブネットは VPC のメインルートテーブルを使用します。代替手段として、ルートテーブルを作成し、サブネットを新しいルートテーブルに関連付けることもできます。
4. [ルート] タブで、[ルートの編集] 、[ルートの追加] の順に選択します。
5. [送信先] に 0.0.0.0/0 と入力し、[ターゲット] のインターネットゲートウェイを選択します。
6. [Save changes] (変更の保存) をクリックします。

コマンドラインを使用してサブネットルートテーブルを設定するには

- create-route (AWS CLI)
- New-EC2Route (AWS Tools for Windows PowerShell)

インターネットゲートウェイを削除する

VPC のインターネットアクセスが不要になった場合、VPC からインターネットゲートウェイをデタッチして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。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。

コンソールを使用して VPC から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 コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[Internet gateways] (インターネットゲートウェイ) を選択します。
3. [インターネットゲートウェイ] のチェックボックスを選択します。
4. アタッチするには、[アクション]、[VPC にアタッチ] の順に選択し、使用可能な VPC を選択して、[インターネットゲートウェイのアタッチ] を選択します。
5. デタッチするには、[アクション]、[VPC からデタッチ] の順に選択し、[インターネットゲートウェイのデタッチ] を選択します。確認を求められたら、[インターネットゲートウェイをデタッチ] を選択します。

コマンドラインを使用して、添付ファイルを含むインターネットゲートウェイを記述するには

- describe-internet-gateways (AWS CLI)
- Get-EC2InternetGateway (AWS Tools for Windows PowerShell)

コマンドラインを使用して VPC からインターネットゲートウェイをデタッチするには

- detach-internet-gateway (AWS CLI)
- Dismount-EC2InternetGateway (AWS Tools for Windows PowerShell)

コンソールを使用してインターネットゲートウェイを削除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[Internet gateways] (インターネットゲートウェイ) を選択します。
3. [インターネットゲートウェイ] のチェックボックスを選択します。
4. [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。
5. 確認を求められたら、「**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。

コマンドラインを使用してインターネットゲートウェイを削除するには

- 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を使用してアウトバウンド IPv6 トラフィックを有効にする

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。

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、「NAT ゲートウェイ」を参照してください。

料金

Egress-only インターネットゲートウェイには課金されませんが、インターネットゲートウェイを使用する EC2 インスタンスにはデータ転送料金が発生します。詳細については、「Amazon EC2 オンデマンド料金」を参照してください。

内容

- Egress-Only インターネットゲートウェイの基本
- サブネットへの Egress-Only インターネットアクセスの追加

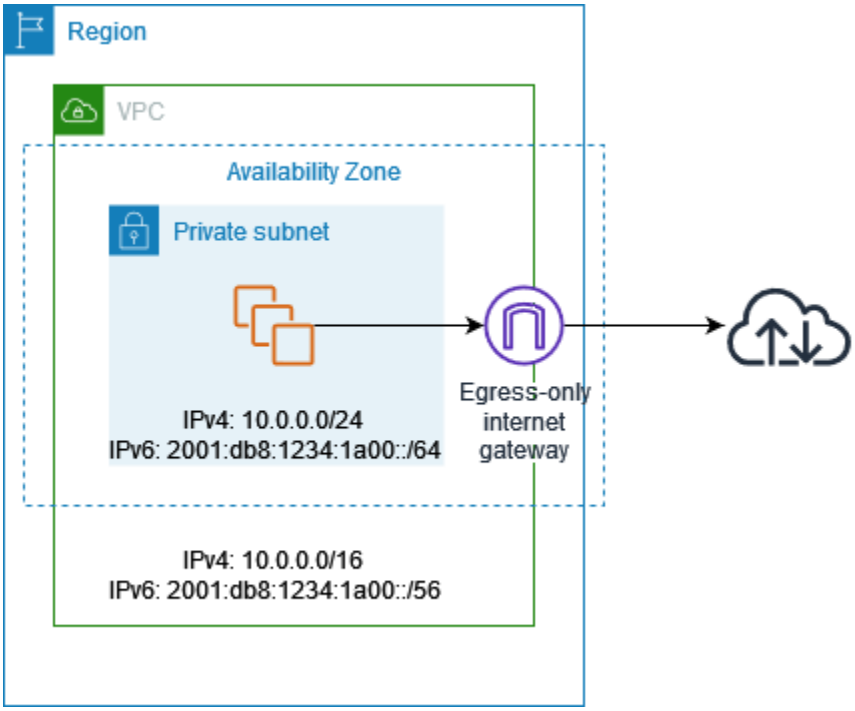
Egress-Only インターネットゲートウェイの基本

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。

セキュリティグループを Egress-Only インターネットゲートウェイに関連付けて、Egress-Only インターネットゲートウェイへの出入りが許可されるトラフィックを制御することはできません。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。

次の図表では、VPC に IPv4 と IPv6 の両方の CIDR ブロックがあり、サブネットにも IPv4 と IPv6 の両方の CIDR ブロックがあります。VPC には、エグレス専用のインターネットゲートウェイがあります。



サブネットに関連付けられているルートテーブルの例を次に示します。インターネットにバインドされたすべての IPv6 トラフィック (::/0) をエグレス専用のインターネットゲートウェイに送信するルートがあります。

ルーティング先	ターゲット
10.0.0.0/16	ローカル
2001:db8:1234:1a00:/64	ローカル
::/0	*eigw-id*

サブネットへの Egress-Only インターネットアクセスの追加

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

- 1. Egress-Only インターネットゲートウェイを作成する
- 2. カスタムルートテーブルを作成する
- 3. Egress-Only インターネットゲートウェイを削除する

- コマンドラインの概要

1. 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. [作成] を選択します。

2. 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、「ルートテーブルを設定する」を参照してください。

3. 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。

コマンドラインの概要

このページで説明しているタスクは、コマンドラインを使用して実行できます。

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)

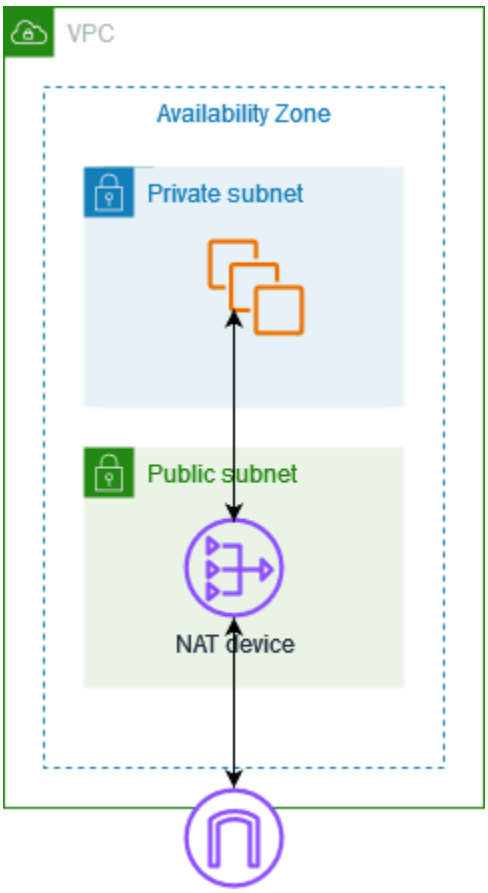
Egress-Only インターネットゲートウェイを削除する

- delete-egress-only-internet-gateway (AWS CLI)
- Remove-EC2EgressOnlyInternetGateway (AWS Tools for Windows PowerShell)

NAT デバイスを使用してインターネットまたは他のネットワークに接続する

NAT デバイスを使用して、プライベートサブネット内のリソー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。

例えば、次の図は、プライベートサブネットの EC2 インスタンス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接続できるようにするパブリックサブネットの NAT デバイスを示しています。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。



Important

- 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。
- 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。

内容

- NAT ゲートウェイ
- NAT インスタンス
- NAT ゲートウェイと NAT インスタンスの比較

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。

- 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、このインスタンスは、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ます。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。
- 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。ただし、このインスタンスは他の VPC またはオンプレミスのネットワーク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。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 IPv4 または IPv6 トラフィック (DNS64 と NAT64 を使用) で使用されます。IPv6 経由での送信専用のインターネット通信を有効にするもう 1 つのオプションは、送信専用インターネットゲートウェイを使用することです。

プライベートとパブリックの両方の NAT ゲートウェイは、インスタンスの送信元プライベート IPv4 アドレスを NAT ゲートウェイのプライベート IPv4 アドレスにマッピングします。パブリック NAT ゲートウェイの場合、インターネットゲートウェイはパブリック NAT ゲートウェイのプライベート IPv4 アドレスを NAT ゲートウェイに関連付けられた Elastic IP アドレスにマッピングします。

インスタンスに応答トラフィックを送信するとき、パブリック NAT ゲートウェイであってもプライベート NAT ゲートウェイであっても、NAT ゲートウェイはアドレスを元の送信元 IP アドレスに変換します。

考慮事項

- 接続は常に NAT ゲートウェイを含む VPC 内から開始する必要があります。
- トラフィックをトランジットゲートウェイと仮想プライベートゲートウェイにルーティングするときは、パブリック NAT ゲートウェイまたはプライベート NAT ゲートウェイのいずれかを使用します。
- プライベート NAT ゲートウェイを使用してトランジ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接続する場合、宛先へのトラフィックはプライベート NAT ゲートウェイのプライベート IP アドレスから送信されます。
- パブリック NAT ゲートウェイを使用して Transit Gateway または仮想プライベートゲートウェイに接続する場合、送信先へのトラフィックはパブリック NAT ゲートウェイのプライベート IP アドレスから送信されます。パブリック NAT ゲートウェイは、同じ VPC 内のインターネットゲートウェイと組み合わせて使用する場合、その Elastic IP アドレスのみを送信元 IP アドレスとして使用します。

内容

- NAT ゲートウェイの基本
- NAT ゲートウェイの使用
- 自動マルチ AZ 拡張用のリージョン NAT ゲートウェイ
- NAT ゲートウェイのユースケース
- DNS64 と NAT64
- NAT ゲートウェイからのトラフィックを検査する
- Amazon CloudWatch による NAT ゲートウェイのモニタリング
- NAT ゲートウェイのトラブルシューティング
- NAT ゲートウェイの料金

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、「ゲートウェイ」を参照してください。

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。耐障害性を高めるに

は、各アベイラビリティーゾーンに 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 ゲートウェイがリソースで使用され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 IPv4 または IPv6 トラフィックでサポートされます。IPv6 トラフィックの場合、NAT ゲートウェイは NAT64 を実行します。これを DNS64 (Route 53 Resolver で利用可能) と組み合わせて使用することで、Amazon VPC のサブネット内の IPv6 ワークロードが IPv4 リソースと通信できます。これらの IPv4 サービスは、オンプレミス環境またはインターネット上の、同じ VPC (別のサブネット内) または別の VPC に存在することがあります。
- NAT ゲートウェイは 5 Gbps の帯域幅をサポートし、100 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 1,000 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。
- 各 IPv4 アドレスは、固有の送信先それぞれに対して最大 55,000 の同時接続をサポートできます。固有の送信先は、送信先 IP アドレス、送信先ポート、およびプロトコル (TCP/UDP/ICMP) の一意の組み合わせで識別されます。この制限は、NAT ゲートウェイに最大 8 個の IPv4 アドレス (1 個のプライマリ IPv4 アドレスと 7 個のセカンダリ IPv4 アドレス) を関連付けることで、引き上げることができます。デフォルトで、パブリック NAT ゲートウェイに関連付ける Elastic IP アドレスは 2 個に制限されています。この制限は、クォータの調整をリクエストすることで引き上げることができます。詳細については、「Elastic IP アドレス」を参照してください。
- NAT ゲートウェイを作成するときに、NAT ゲートウェイに割り当てられているプライマリプライベート IPv4 アドレスを選択できます。選択しない場合は、サブネットの IPv4 アドレス範囲から自動的に選択されます。プライマリプライベート IPv4 アドレスを変更または削除することはできません。必要に応じて、セカンダリプライベート IPv4 アドレスを追加できます。
- 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 ゲートウェイのリクエスタマネージドネットワークインターフェイスが作成されます。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。説明の中から NAT ゲートウェイの ID を検索します。ネットワークインターフェイスにタグを追加できますが、このネットワークインターフェイスの他のプロパティを変更することはできません。

- 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、「ネットワーク ACL」を参照してください。
- VPC ピアリング接続を経由して 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。ただし、VPC ピア接続を介した NAT ゲートウェイからピア接続された VPC の送信先へのトラフィックは、「送信者に戻る」動作をサポートします。リターントラフィックは、送信先 VPC でリターンルートが設定されていなくても、送信元 NAT ゲートウェイに自動的にルーティングされて戻ります。この動作は NAT ゲートウェイに固有であり、標準の EC2 インスタンスには適用されません。これを避けるには、NACL を使用してリターントラフィックをブロックします。

  サポート外:

  ```
  Client # Peering # NAT # Internet
  ```

  サポート対象:

  ```
  Client # NAT # Peering # Destination
  ```

- 仮想プライベートゲートウェイを使用して Site-to-Site VPN または Direct Connect から NAT ゲートウェイへの方向のトラフィックをルーティングすることはできません。仮想プライベートゲートウェイではなく、トランジットゲートウェイを使用する場合は、Site-to-Site VPN または Direct Connect から NAT ゲートウェイにトラフィックをルーティングできます。
- NAT ゲートウェイは最大伝送単位 (MTU) が 8500 のトラフィックをサポートしますが、以下の点に注意する必要があります。
  - MTU とは、接続を介して渡すことができる最大許容パケットサイズ (バイト) です。接続の 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。
  - NAT ゲートウェイに到達した 8,500 バイトを超えるパケットはドロップ (または該当する場合はフラグメント化) されます。
  - パブリック NAT ゲートウェイを使用してインターネット経由でリソースと通信する際にパケットロスが発生するのを防ぐため、EC2 インスタンスの MTU 設定は 1500 バイトを超えないようにしてください。インスタンスの MTU の確認と設定については、「Amazon EC2 ユーザーガイド」の「EC2 インスタンスのネットワーク MTU」を参照してください。
  - NAT ゲートウェイは、FRAG_NEEDED ICMPv4 パケットとパケット・トゥー・ビッグ (PTB) ICMPv6 パケットによるパス MTU ディスカバリー (PMTUD) をサポートします。
  - NAT ゲートウェイは、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。

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。

タスク

- NAT ゲートウェイの使用を制御する
- NAT ゲートウェイを作成する
- セカンダリ IP アドレスの関連付けを編集する
- NAT ゲートウェイのタグ付け
- NAT ゲートウェイの削除
- コマンドラインの概要

NAT ゲートウェイの使用を制御する

デフォルトでは、 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権限がありません。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ポリシーがアタッチされた IAM ロールを作成できます。詳細については、「Amazon VPC の Identity and Access Management」を参照してください。

NAT ゲートウェイを作成する

NAT ゲートウェイを作成するには、以下の手順を実行します。

関連クォータ

- アカウントに割り当てられた Elastic IP アドレスの数を使い果たすと、パブリック NAT ゲートウェイを作成できなくなります。詳細については、「Elastic IP アドレス」を参照してください。
- プライベート NAT ゲートウェイには、最大 8 個のプライベート IPv4 アドレスを割り当てることができます。この制限は調整できません。
- デフォルトで、パブリック NAT ゲートウェイに関連付ける Elastic IP アドレスは 2 個に制限されています。この制限は、クォータの調整をリクエストすることで引き上げることができます。詳細については、「Elastic IP アドレス」を参照してください。

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 ゲートウェイを作成] を選択します。
4. (オプション) 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。
5. 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。
6. [接続タイプ] で、パブリック NAT ゲートウェイを作成する場合はデフォルトの [パブリック] が選択されたままにしておき、プライベート NAT ゲートウェイを作成する場合は [プライベート] を選択します。パブリック NAT ゲートウェイとプライベート NAT ゲートウェイの違いに関する詳細については、「NAT ゲートウェイ」を参照してください。
7. [パブリック] を選択した場合は、以下の手順に従うか、ステップ 8 に進みます。

    1. [Elastic IP 割り当て ID] を選択して NAT ゲートウェイに Elastic IP アドレスを割り当てるか、[Elastic IP を割り当て] を選択してパブリック NAT ゲートウェイのアドレスを自動的に割り当てます。デフォルトで、パブリック NAT ゲートウェイに関連付ける Elastic IP アドレスは 2 個に制限されています。この制限は、クォータの調整をリクエストすることで引き上げることができます。詳細については、「Elastic IP アドレス」を参照してください。

    > **Important**
    >
    > Elastic IP アドレスをパブリック NAT ゲートウェイに割り当てる場合、EIP のネットワークボーダーグループは、パブリック NAT ゲートウェイを起動するアベイラビリティーゾーン (AZ) のネットワークボーダーグループと一致する必要があります。同じでない場合、NAT ゲートウェイは起動に失敗します。サブネットの詳細を表示すると、サブネットの AZ のネットワークボーダーグループを確認できます。同様に、EIP アドレスの詳細を表示することで EIP のネットワークボーダーグループを表示できます。詳細については、「1. Elastic IP アドレスを割り当てる」を参照してください。

    2. (オプション) [追加設定] を選択して、[プライマリプライベート IP アドレス - オプション] で NAT ゲートウェイのプライベート IPv4 アドレスを入力します。アドレスを入力しない場合は、AWS が自動的に、NAT ゲートウェイがあるサブネットからのプライベート IPv4 アドレスを NAT ゲートウェイにランダムに割り当てます。
    3. ステップ 11 に進みます。

8. [プライベート] を選択した場合は、[追加設定] の [プライベート IPv4 アドレスの割り当て方法] で以下のいずれかを選択します。

- [自動的に割り当て]: AWS は、NAT ゲートウェイのプライマリプライベート IPv4 アドレスを選択します。[自動割り当てのプライベート IPv4 アドレス数] には、NAT ゲートウェイのセカンダリプライベート IPv4 アドレスの数をオプションで指定できます。AWS は、NAT ゲートウェイのサブネットからこれらの IP アドレスをランダムに選択します。
- [カスタム]: [プライマリプライベート IPv4 アドレス] で NAT ゲートウェイのプライマリプライベート IPv4 アドレスを選択します。[セカンダリプライベート IPv4 アドレス] には、オプションで NAT ゲートウェイに最大 7 つのセカンダリプライベート IPv4 アドレスを指定できます。

9. ステップ 8 で [カスタム] を選択した場合は、このステップをスキップしてください。[自動割り当て] を選択した場合は、[自動で割り当てられたプライベート IP アドレスの数] で、AWS がこのプライベート NAT ゲートウェイに割り当てるセカンダリ IPv4 アドレスの数を選択します。IPv4 アドレスは、最大 7 個選択できます。

> **Note**
>
> セカンダリ IPv4 アドレスはオプションであり、NAT ゲートウェイを使用するワークロードの単一の送信先 (同じ送信先 IP、送信先ポート、およびプロトコル) への同時接続数が 55,000 個を超える場合に割り当てる、または割り振られる必要があります。セカンダリ IPv4 アドレスは利用可能なポート数を増やすことから、ワークロードが NAT ゲートウェイを使用して確立できる同時接続数の上限も増加します。

10. ステップ 9 で [自動割り当て] を選択した場合は、このステップをスキップしてください。[カスタム] を選択した場合は、以下を実行します。

    1. [プライマリプライベート IPv4 アドレス] に、プライベート IPv4 アドレスを入力します。
    2. [セカンダプライベート IPv4 アドレス] に、最大 7 個のセカンダリプライベート IPv4 アドレスを入力します。

11. (オプション) タグを NAT ゲートウェイに追加するには、[Add new tag] (新しいタグを追加) を選択し、キー名と値を入力します。最大 50 個のタグを追加できます。
12. [NAT ゲートウェイを作成] を選択します。
13. 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。必要に応じて、ルートテーブルを更新するようにしてください。例については、the section called “ユースケース” 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ています。詳細については、「NAT ゲートウェイの作成に失敗する」を参照してください。

セカンダリ IP アドレスの関連付けを編集する

各 IPv4 アドレスは、固有の送信先それぞれに対して最大 55,000 の同時接続をサポートできます。固有の送信先は、送信先 IP アドレス、送信先ポート、およびプロトコル (TCP/UDP/ICMP) の一意の組み合わせで識別されます。この制限は、NAT ゲートウェイに最大 8 個の IPv4 アドレス (1 個のプライマリ IPv4 アドレスと 7 個のセカンダリ IPv4 アドレス) を関連付けることで、引き上げることができます。デフォルトで、パブリック NAT ゲートウェイに関連付ける Elastic IP アドレスは 2 個に制限されています。この制限は、クォータの調整をリクエストすることで引き上げることができます。詳細については、「Elastic IP アドレス」を参照してください。

NAT ゲートウェイの CloudWatch メトリクスである ErrorPortAllocation と PacketsDropCount を使用して、NAT ゲートウェイがポート割り当てエラーを生成しているかどうか、またはパケットをドロップしているかどうかを判断できます。この問題を解決するには、NAT ゲートウェイにセカンダリ IPv4 アドレスを追加します。

考慮事項

- セカンダリプライベート IPv4 アドレスは、プライベート NAT ゲートウェイの作成時、またはこのセクションの手順を使用して NAT ゲートウェイを作成した後で追加できます。Elastic IP アドレスをパブリック NAT ゲートウェイに追加できるのは、このセクションの手順を使用して NAT ゲートウェイを作成した後のみです。
- NAT ゲートウェイには、最大 8 個の IPv4 アドレス (1 個のプライマリ IPv4 アドレスと 7 個のセカンダリ IPv4 アドレス) を関連付けることができます。プライベート NAT ゲートウェイには、最大 8 個のプライベート IPv4 アドレスを割り当てることができます。デフォルトで、パブリック NAT ゲートウェイに関連付ける Elastic IP アドレスは 2 個に制限されています。この制限は、クォータの調整をリクエストすることで引き上げることができます。詳細については、「Elastic IP アドレス」を参照してください。

セカンダリ IPv4 アドレスの関連付けを編集する

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [NAT ゲートウェイ] を選択します。
3. セカンダリ IPv4 アドレスの関連付けを編集する NAT ゲートウェイを選択します。
4. [アクション] を選択してから、[セカンダリ IP アドレスの関連付けを編集] を選択します。

5. プライベート NAT ゲートウェイのセカンダリ IPv4 アドレスの関連付けを編集している場合は、[アクション] で [新しい IPv4 アドレスの割り当て] または [既存の IPv4 アドレスの割り当てを解除する] を選択します。パブリック NAT ゲートウェイのセカンダリ IPv4 アドレスの関連付けを編集している場合は、[アクション] で [新しい IPv4 アドレスの関連付け] または [既存の IPv4 アドレスの関連付けを解除する] を選択します。
6. 次のいずれかを行います。

   - 新しい IPv4 アドレスの割り当てまたは関連付けを選択した場合は、以下を実行します。

     1. このステップは必須です。プライベート IPv4 アドレスを選択する必要があります。[プライベート IPv4 アドレスの割り当て方法] を選択します。

        - [自動割り当て]: AWS がプライマリプライベート IPv4 アドレスを自動的に選択します。AWS が NAT ゲートウェイに割り当てるセカンダリプライベート IPv4 アドレス (最大 7 個) を割り当てるかどうかはユーザーが選択します。AWS は自動的に、NAT ゲートウェイがあるサブネットからのアドレスをランダムに選択して割り当てます。
        - [カスタム]: NAT ゲートウェイに割り当てるプライマリプライベート IPv4 アドレスと、最大 7 個のセカンダリプライベート IPv4 アドレスを選択します。

     2. [Elastic IP 割り当て ID] で、セカンダリ IPv4 アドレスとして追加する Elastic IP アドレスを選択します。このステップは必須です。プライベート IPv4 アドレスとともに、Elastic IP アドレスも選択する必要があります。[プライベート IP アドレスの割り当て方法] で [カスタム] を選択した場合は、追加する Elastic IP アドレスごとにプライベート IPv4 アドレスを入力する必要もあります。

        > **Important**
        >
        > セカンダリ EIP をパブリック NAT ゲートウェイに割り当てる場合、EIP のネットワークボーダーグループは、パブリック NAT ゲートウェイを起動するアベイラビリティーゾーン (AZ) のネットワークボーダーグループと一致する必要があります。同じでない場合、EIP は割り当てに失敗します。サブネットの詳細を表示すると、サブネットの AZ のネットワークボーダーグループを確認できます。同様に、EIP アドレスの詳細を表示することで EIP のネットワークボーダーグループを表示できます。詳細については、「1. Elastic IP アドレスを割り当てる」を参照してください。

   NAT ゲートウェイには、最大 8 個の IP アドレスを関連付けることができます。これがパブリック NAT ゲートウェイである場合、リージョンあたりの Elastic IP アドレス数にデフォル

トのクォータ制限があります。詳細については、「Elastic IP アドレス」を参照してください。

- 新しい IPv4 アドレスの割り当ての解除、または関連付けの解除を選択した場合は、以下を実行します。
  1. [割り当てを解除する既存のセカンダリ IP アドレス] で、割り当てを解除するセカンダリ IP アドレスを選択します。
  2. (オプション) [接続ドレイン期間] には、接続がまだ進行中の場合に IP アドレスを強制的に解放するまでの最大待機時間 (秒単位) を入力します。値を指定しない場合のデフォルト値は 350 秒です。

7. [Save changes] (変更の保存) をクリックします。

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、「NAT ゲートウェイの作成に失敗する」を参照してください

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、「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 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 ゲートウェイにタグを付ける

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [NAT ゲートウェイ] を選択します。
3. タグ付けする NAT ゲートウェイを選択し、[アクション] を選択します。次に、[タグを管理] を選択します。
4. [新しいタグを追加] を選択して、タグの [キー] と [値] を定義します。最大 50 個のタグを追加できます。
5. [保存] を選択します。

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、「5. Elastic IP アドレスをリリース」を参照してください。

コマンドラインの概要

このページで説明しているタスクは、コマンドラインを使用して実行できます。

プライベート NAT ゲートウェイにプライベート IPv4 アドレスを割り当てる

- assign-private-nat-gateway-address (AWS CLI)
- Register-EC2PrivateNatGatewayAddress (AWS Tools for Windows PowerShell)

Elastic IP アドレスとプライベート IPv4 アドレスをパブリック NAT ゲートウェイに関連付ける

- associate-nat-gateway-address (AWS CLI)
- Register-EC2NatGatewayAddress (AWS Tools for Windows PowerShell)

NAT ゲートウェイを作成する

- create-nat-gateway (AWS CLI)

- New-EC2NatGateway (AWS Tools for Windows PowerShell)

NAT ゲートウェイを削除する

- delete-nat-gateway (AWS CLI)
- Remove-EC2NatGateway (AWS Tools for Windows PowerShell)

NAT ゲートウェイを記述する

- describe-nat-gateways (AWS CLI)
- Get-EC2NatGateway (AWS Tools for Windows PowerShell)

パブリック NAT ゲートウェイからセカンダリ Elastic IP アドレスの関連付けを解除する

- disassociate-nat-gateway-address (AWS CLI)
- Unregister-EC2NatGatewayAddress (AWS Tools for Windows PowerShell)

NAT ゲートウェイにタグを付ける

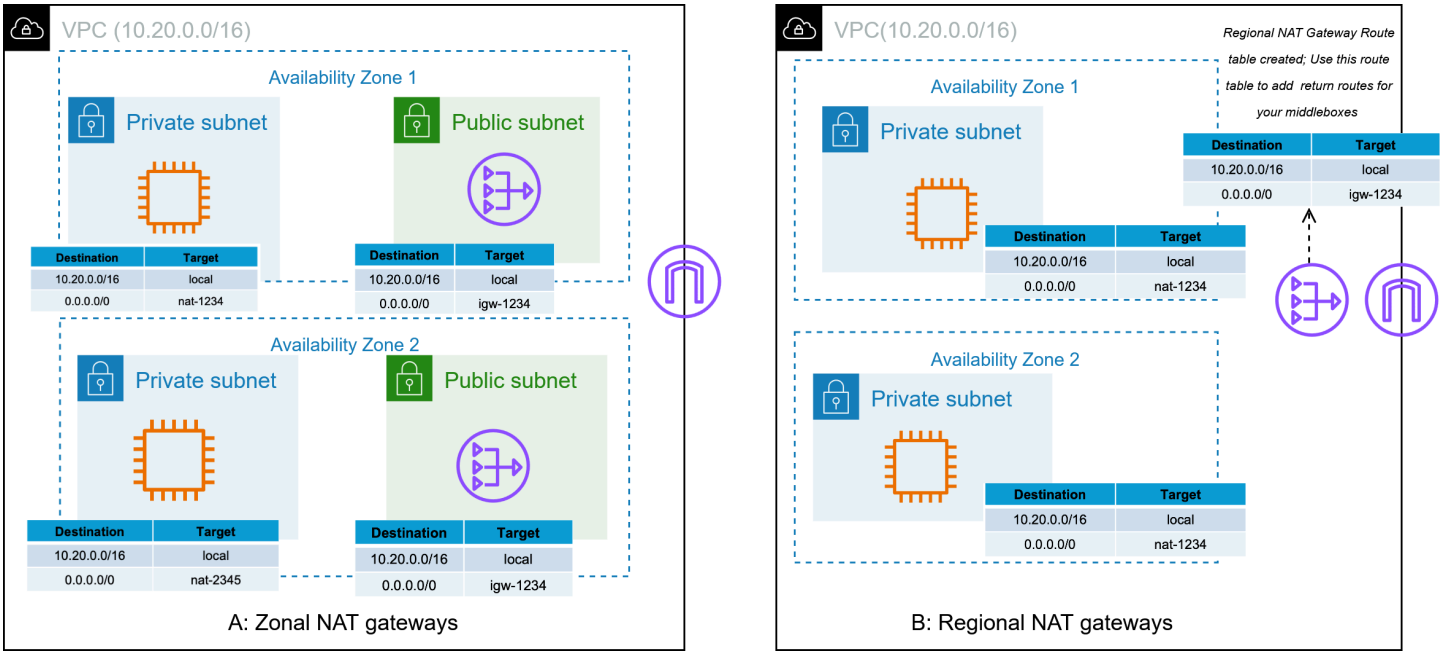
- create-tags (AWS CLI)
- New-EC2Tag (AWS Tools for Windows PowerShell)

プライベート NAT ゲートウェイからセカンダリ IPv4 アドレスの割り当てを解除する

- unassign-private-nat-gateway-address (AWS CLI)
- Unregister-EC2PrivateNatGatewayAddress (AWS Tools for Windows PowerShell)

自動マルチ AZ 拡張用のリージョン NAT ゲートウェイ

ネットワークアーキテクチャを簡素化し、セキュリティ体制を改善し、デフォルトで高可用性を設定する場合は、リージョン NAT ゲートウェイを使用します。リージョン NAT ゲートウェイは、ワークロードの存在に基づいてアベイラビリティーゾーン間で自動的に拡張されます。単一のアベイラビリティーゾーンで動作する標準 NAT ゲートウェイ (ゾーン NAT ゲートウェイと呼ばれます) とは異なり、リージョン NAT ゲートウェイはワークロードに従って自動高可用性を提供します。



左側の図 A は、ゾーン NAT ゲートウェイを使用した現在のセットアップを示しています。まず、アベイラビリティーゾーンごとにゾーン NAT ゲートウェイを作成し、パブリックサブネットで NAT をホストします。次に、アベイラビリティーゾーンごとにプライベートサブネットからそのアベイラビリティーゾーンの NAT への個別のルートを設定します。高可用性を確保するため、ワークロードが新しいアベイラビリティーゾーンに拡張されるたびに、このステップを繰り返します。さらに、アベイラビリティーゾーンごとに NAT サブネットのルートテーブルにインターネットゲートウェイのルートを追加する必要があります。

一方、リージョン NAT ゲートウェイでは、パブリックサブネットを作成してホストする必要はありません。また、ワークロードが新しいアベイラビリティーゾーンに拡張されるたびに、NAT ゲートウェイを作成および削除したり、ルートテーブルを編集したりする必要はありません。代わりに、リージョンモードの NAT ゲートウェイを作成し、VPC を選択するだけで、ワークロードの存在に基づいてすべての AZ 間で自動的に拡張および縮小され、高可用性が提供されます。図 B に示すように、プライベートサブネット内のリソースからのトラフィックをすべての AZ 間でこの単一のリージョン NAT ゲートウェイ ID にルーティングすることも、AZ 内のサブネット間で同じルートテーブルを使用してネットワークアドレス変換を実行することもできます。リージョン NAT ゲートウェイを作成すると、AWS は自動的にそのルートテーブルを作成します。このテーブルには、インターネットゲートウェイへの事前設定されたルートが付属しています。このルートテーブルを使用して、ミドルボックスに戻りルートを追加できます。

利点

リージョン NAT ゲートウェイには以下の利点があります。

- セットアップの簡素化 – ネットワークインターフェイスを持つすべてのアベイラビリティーゾーンで単一の NAT ID を使用するため、異なるアベイラビリティーゾーンのサブネットに同じルートエントリを使用できます。
- セキュリティの強化 – パブリックサブネットは必要ありません。リージョン NAT ゲートウェイは、独自のルートテーブルを持つスタンドアロンリソースであり、リージョン NAT ゲートウェイをホストするために VPC 内のパブリックサブネットを必要としないため、パブリック接続を使用してサブネット内のプライベートリソースを誤って設定する可能性が低くなります。
- 自動高可用性 – ワークロードフットプリントを自動的に拡張および縮小し、デフォルトで高可用性を提供するゾーンアフィニティを維持します。
- ポートと IP の制限の引き上げ – リージョン NAT ゲートウェイは、アベイラビリティーゾーンごとに最大 32 個の IP アドレスをサポートします (ゾーン NAT ゲートウェイの場合は 8 個)。各 IP アドレスは、一般的な送信先 (送信先 IP、送信先ポート、プロトコルの一意の組み合わせで識別) への同時接続の制限を 55,000 引き上げます。

リージョン NAT ゲートウェイを使用するタイミング

プライベート接続を必要とするユースケースを除くすべてのユースケースで、リージョン NAT ゲートウェイを使用することを検討してください。リージョン NAT ゲートウェイはプライベート接続を提供しないため、プライベート NAT ユースケースではゾーンアベイラビリティーモードで NAT ゲートウェイを使用することをお勧めします。

リージョン NAT ゲートウェイの仕組み

新しいアベイラビリティーゾーンでリソースを起動すると、リージョン NAT ゲートウェイはそのアベイラビリティーゾーン内のネットワークインターフェイス (ENI) の存在を検出し、自動的にそのゾーンに拡張します。同様に、NAT ゲートウェイはアクティブなワークロードがないアベイラビリティーゾーンから縮小します。

リソースがインスタンス化された後、リージョン NAT ゲートウェイが新しいアベイラビリティーゾーンに拡張されるまでに最大 60 分かかる場合があります。この拡張が完了するまで、このリソースからの関連するトラフィックは、既存のアベイラビリティーゾーンのいずれかのリージョン NAT ゲートウェイによってゾーン間で処理されます。

リージョン NAT ゲートウェイは 2 つのモードをサポートします。

- 自動モード – このモードでは、AWS は IP アドレスとアベイラビリティーゾーンの拡張を自動的に管理します (推奨)。このモードで独自の IP アドレスを使用し、Amazon VPC IPAM を使用する

場合は、「Amazon VPC IPAM ユーザーガイド」の「IPAM ポリシーを使用してパブリック IPv4 割り当て戦略を定義する」を参照してください。

- 手動モード – このモードでは、各アベイラビリティーゾーンの IP アドレスを手動で管理し、ネットワークアドレス変換を制御します。手動モードでは、アベイラビリティーゾーン間で NAT ゲートウェイを拡張および縮小する責任がユーザーにあります。

> **Important**
>
> リージョン NAT ゲートウェイは、リージョン NAT ゲートウェイルートテーブルで有効なルートとして AWS Transit Gateway をサポートします。リージョン NAT ゲートウェイはプライベート NAT をサポートしていません。プライベート NAT が必要な場合は、代わりにゾーン NAT ゲートウェイを使用してください。リージョン NAT ゲートウェイは、制約のあるアベイラビリティーゾーンではサポートされていません。

料金

料金については、「Amazon VPC の料金」を参照してください。

リージョン NAT ゲートウェイを作成する

コンソールを使用する

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [NAT ゲートウェイ] を選択します。
3. [NAT ゲートウェイを作成] を選択します。
4. [アベイラビリティーモード] で、[リージョン] を選択します。リージョンの可用性を選択するときは、サブネットを指定する必要はありません。
5. VPC を選択してください。
6. 残りの設定を完了し、[NAT ゲートウェイの作成] を選択します。

AWS CLI の使用

リージョン NAT ゲートウェイを作成する

```
aws ec2 create-nat-gateway --vpc-id vpc-12345678 --availability-mode regional
```

NAT ゲートウェイの詳細を表示する

```
aws ec2 describe-nat-gateways --nat-gateway-ids nat-12345678
```

IP アドレスを追加する (手動モード)

```
aws ec2 associate-nat-gateway-address --nat-gateway-id nat-12345678 --availability-zone
 us-east-1b --allocation-ids eipalloc-12345678
```

IP アドレスを削除する

```
aws ec2 disassociate-nat-gateway-address --nat-gateway-id nat-12345678 --association-
ids eipassoc-12345678
```

リージョン NAT ゲートウェイを削除する

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-12345678
```

ゾーン NAT ゲートウェイからリージョン NAT ゲートウェイへの変換

> **Important**
>
> これにより、既存の接続がリセットされます。メンテナンスウィンドウでこれらのステップを完了することをお勧めします。

既存のゾーン NAT ゲートウェイをリージョン NAT ゲートウェイに変換するには、次の 2 つの方法のいずれかを使用します。

新しい IP アドレスでリージョン NAT ゲートウェイを使用しても問題ない場合:

1. 新しいリージョン NAT ゲートウェイを作成する
2. リージョン NAT ゲートウェイを参照するようにルートテーブルを更新する
3. 古いゾーン NAT ゲートウェイを削除する

このアプローチでは、新しい IP アドレスを使用し、ルートが更新されると既存の接続をリセットします。

リージョン NAT ゲートウェイで既存の IP アドレスを再利用する場合:

1. 既存のゾーン NAT ゲートウェイを削除して IP アドレスを解放する
2. リリースされた IP アドレスを使用してリージョン NAT ゲートウェイを作成する
3. リージョン NAT ゲートウェイを参照するようにルートテーブルを更新する

このアプローチでは IP アドレスは保持されますが、移行中にトラフィックが中断されるため、メンテナンスウィンドウが必要です。

NAT ゲートウェイのユースケース

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

- 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
- 許可リストに含まれる IP アドレスを使用してネットワークにアクセスする
- 重複するネットワーク間の通信を有効にする

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がアウトバウンドトラフィックをインターネットへの送信を有効にすると同時に、インターネットからインスタンスへの接続の確立を防ぎます。

内容

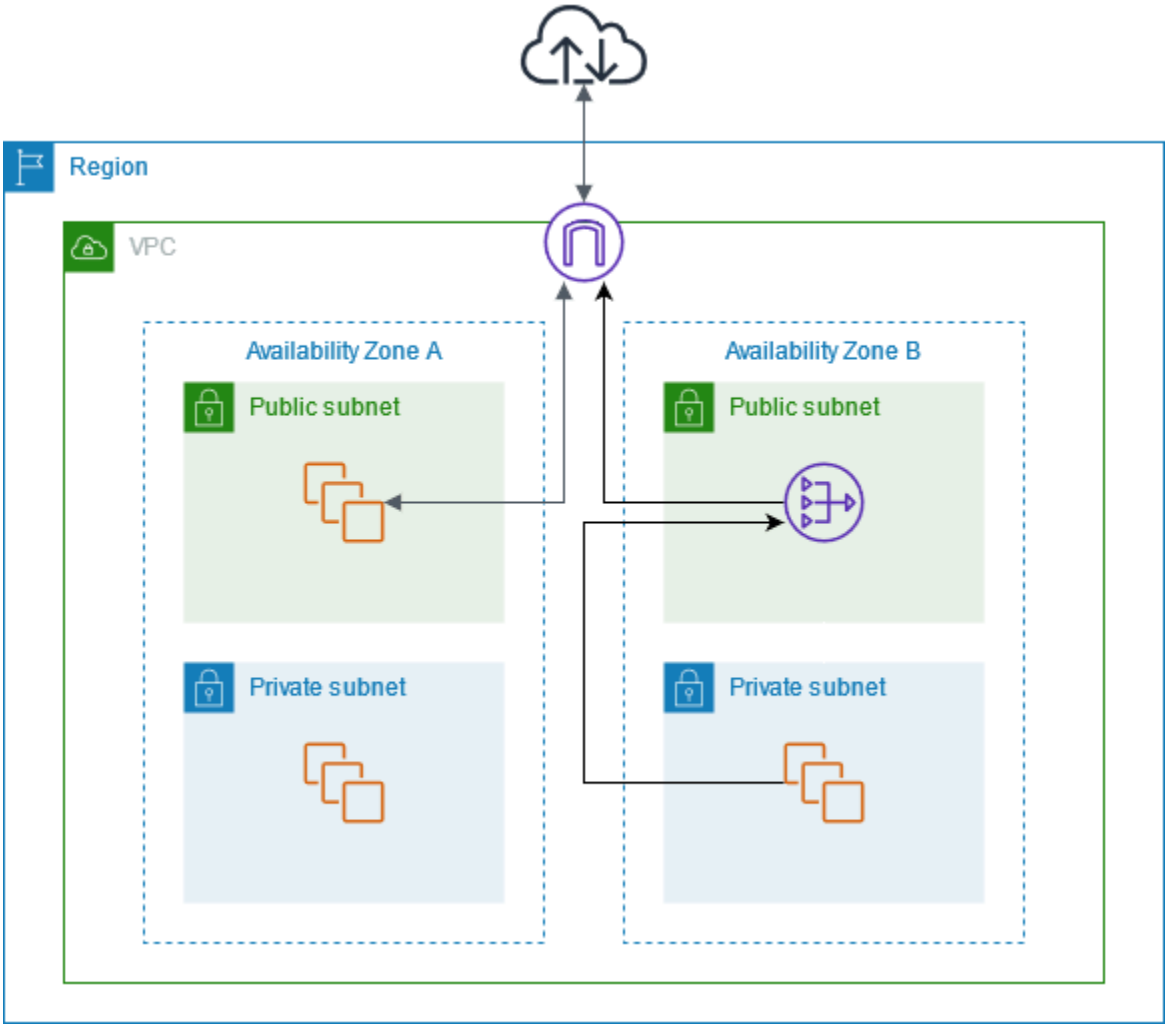
- 概要
- ルーティング
- パブリック NAT ゲートウェイのテスト

概要

次の図表は、このユースケースを示しています。アベイラビリティーゾーンが 2 つあり、それぞれのアベイラビリティーゾーンに 2 つのサブネットがあります。各サブネットのルートテーブルは、トラフィックのルーティング方法を決定します。アベイラビリティーゾーン A では、パブリックサブネットのインスタンスはインターネットゲートウェイへのルートを介してインターネットに到達で

きますが、プライベートサブネットのインスタンスにはインターネットへのルートがありません。アベイラビリティーゾーン B では、パブリックサブネットに NAT ゲートウェイが含まれており、プライベートサブネット内のインスタンスは、パブリックサブネット内の NAT ゲートウェイへのルートを介してインターネットに到達できます。

プライベート NAT ゲートウェイとパブリック NAT ゲートウェイはどちらも、インスタンスの送信元プライベート IPv4 アドレスをプライベート NAT ゲートウェイのプライベート IPv4 アドレスにマッピングしますが、パブリック NAT ゲートウェイの場合、インターネットゲートウェイはパブリック NAT ゲートウェイのプライベート IPv4 アドレスを NAT ゲートウェイに関連付けられた Elastic IP アドレスにマッピングします。インスタンスに応答トラフィックを送信するとき、パブリック NAT ゲートウェイであってもプライベート NAT ゲートウェイであっても、NAT ゲートウェイはアドレスを元の送信元 IP アドレスに変換します。



アベイラビリティーゾーン A のプライベートサブネットにあるインスタンスもインターネットにアクセスする必要がある場合は、このサブネットからアベイラビリティーゾーン B の NAT ゲートウェイへのルートを作成することができます。または、インターネットアクセスを必要とするリソースが

含まれる各アベイラビリティーゾーンに NAT ゲートウェイを作成することで、耐障害性を向上させることができます。図の例については、「the section called “プライベートサーバー”」を参照してください。

ルーティング

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たルートテーブルです。最初のエントリはローカルルートで、サブネット内のインスタンスがプライベート IP アドレスを使用して VPC 内の他のインスタンスとの通信を有効にします。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、サブネット内のインスタンスのインターネットアクセスを有効にします。

ルーティング先	Target
VPC CIDR	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 A 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。エントリはローカルルートで、サブネット内のインスタンスはプライベート IP アドレスを使用して VPC 内の他のインスタンスとの通信を有効にします。このサブネットのインスタンスはインターネットにアクセスできません。

ルーティング先	Target
VPC CIDR	ローカル

以下は、アベイラビリティーゾーン B 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。最初のエントリはローカルルートで、サブネット内のインスタンスがプライベート IP アドレスを使用して VPC 内の他のインスタンスとの通信を有効にします。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、サブネット内の NAT ゲートウェイのインターネットアクセスを有効にします。

ルーティング先	Target
VPC CIDR	ローカル

ルーティング先	Target
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。最初のエントリはローカルルートで、サブネット内のインスタンスがプライベート IP アドレスを使用して VPC 内の他のインスタンスとの通信を有効にします。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 ゲートウェイに送信します。

ルーティング先	Target
VPC CIDR	ローカル
0.0.0.0/0	*nat-gateway-id*

詳細については、「the section called “サブネットルートテーブルを管理する”」を参照してください。

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」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 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)。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。

   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 トラフィックとプライベートサブネットの IP アドレス範囲への SSH 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。このテストのインバウンド SSH トラフィックとアウトバウンド 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」または「Windows 用に SSH エージェント転送を設定するには」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、「インターネット接続をテストするには」を参照してください

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 用に SSH エージェント転送を設定するには

Windows で利用可能な OpenSSH クライアントを使用するか、希望する SSH クライアント (PuTTY など) をインストールできます。

OpenSSH

「Getting started with OpenSSH for Windows」の説明に従って、Windows 用 OpenSSH をインストールします。次に、認証エージェントにキーを追加します。詳細については、「Key-based authentication in OpenSSH for Windows」を参照してください。

PuTTY

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、「Amazon EC2 ユーザーガイド」の「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) のコンテキストメニューを開き、[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、「PuTTY を使用した 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」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、「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。

許可リストに含まれる IP アドレスを使用してネットワークにアクセスする

プライベート NAT ゲートウェイを使用することで、許可リストに含まれるアドレスのプールを使用して、VPC からオンプレミスネットワークへの通信を有効にすることができます。

各インスタンスに許可リストの IP アドレス範囲から個別の IP アドレスを割り当てる代わりに、許可リストの IP アドレス範囲の IP アドレスを持つプライベート NAT ゲートウェイを介してサブネットからのトラフィックをルーティングできます。

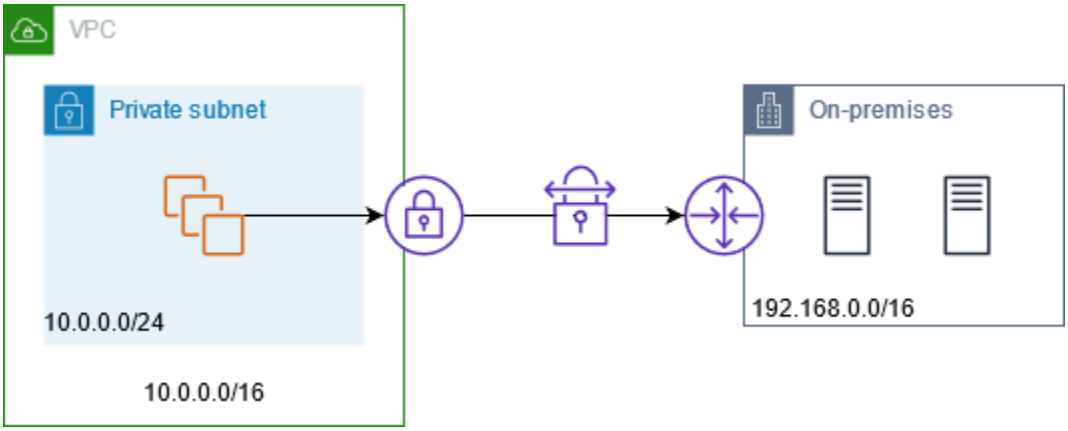
内容

- 概要
- リソース
- ルーティング

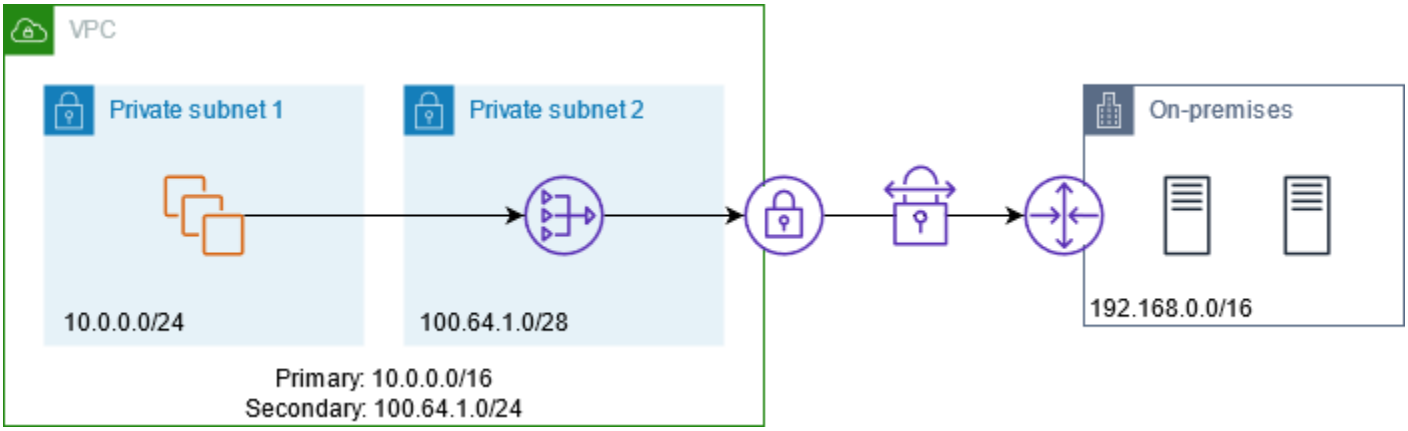
概要

次の図表は、インスタンスが Site-to-Site VPN を介してオンプレミスリソースにアクセスする方法を示しています。インスタンスからのトラフィックは、仮想プライベートゲートウェイから VPN 接続を介して、カスタマーゲートウェイにルーティングされ、そしてオンプレミスネットワークの宛先

にルーティングされます。ただし、宛先が特定の IP アドレス範囲 (100.64.1.0/28 など) からのトラフィックのみを許可するとします。これにより、これらのインスタンスからのトラフィックがオンプレミスネットワークに到達するのを防ぐことができます。



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。VPC には、元の IP アドレス範囲に加え、許可された IP アドレス範囲があります。VPC には、プライベート NAT ゲートウェイを使用して許可された IP アドレス範囲のサブネットがあります。インスタンスからのオンプレミスネットワーク宛のトラフィックは、VPN 接続にルーティングされる前に、NAT ゲートウェイに送信されます。オンプレミスネットワークは、NAT ゲートウェイの送信元 IP アドレスが許可された IP アドレスの範囲のインスタンスからのトラフィックを受信します。



リソース

次のようにリソースを作成または更新します。

- 許可された IP アドレス範囲を VPC に関連付けます。
- 許可された IP アドレスの範囲から VPC にサブネットを作成します。
- 新しいサブネット内にプライベート NAT ゲートウェイを作成します。

- インスタンスでサブネットのルートテーブルを更新して、オンプレミスネットワーク宛てのトラフィックを NAT ゲートウェイに送信します。オンプレミスネットワーク宛てのトラフィックを、仮想プライベートゲートウェイに送信するプライベート NAT ゲートウェイを使用して、サブネットのルートテーブルにルートを追加します。

ルーティング

以下は、最初のサブネットに関連付けられているルートテーブルです。VPC CIDR ごとにローカルルートが存在します。ローカルルートにより、サブネット内のリソースは、プライベート IP アドレスを使用して VPC 内の他のリソースとの通信が有効化されます。3 番目のエントリは、オンプレミスネットワーク宛てのトラフィックを、プライベート NAT ゲートウェイに送信します。

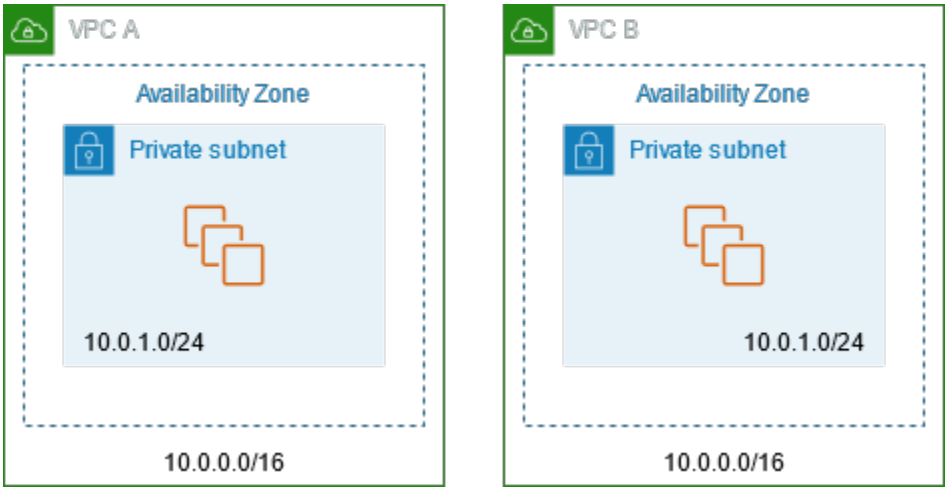
ルーティング先	ターゲット
10.0.0.0/16	local
100.64.1.0/24	ローカル
192.168.0.0/16	*nat-gateway-id*

以下は、2 番目のサブネットに関連付けられたルートテーブルです。VPC CIDR ごとにローカルルートが存在します。ローカルルートにより、サブネット内のリソースは、プライベート IP アドレスを使用して VPC 内の他のリソースとの通信が有効化されます。3 番目のエントリは、オンプレミスネットワーク宛てのトラフィックを、仮想プライベートゲートウェイに送信します。

ルーティング先	ターゲット
10.0.0.0/16	local
100.64.1.0/24	ローカル
192.168.0.0/16	*vgw-id*

重複するネットワーク間の通信を有効にする

プライベート NAT ゲートウェイを使用して、重複する CIDR 範囲がある場合でも、ネットワーク間の通信を有効にできます。例えば、VPC A のインスタンスが VPC B のインスタンスによって提供されるサービスにアクセスする必要があるとします。



内容

- 概要
- リソース
- ルーティング

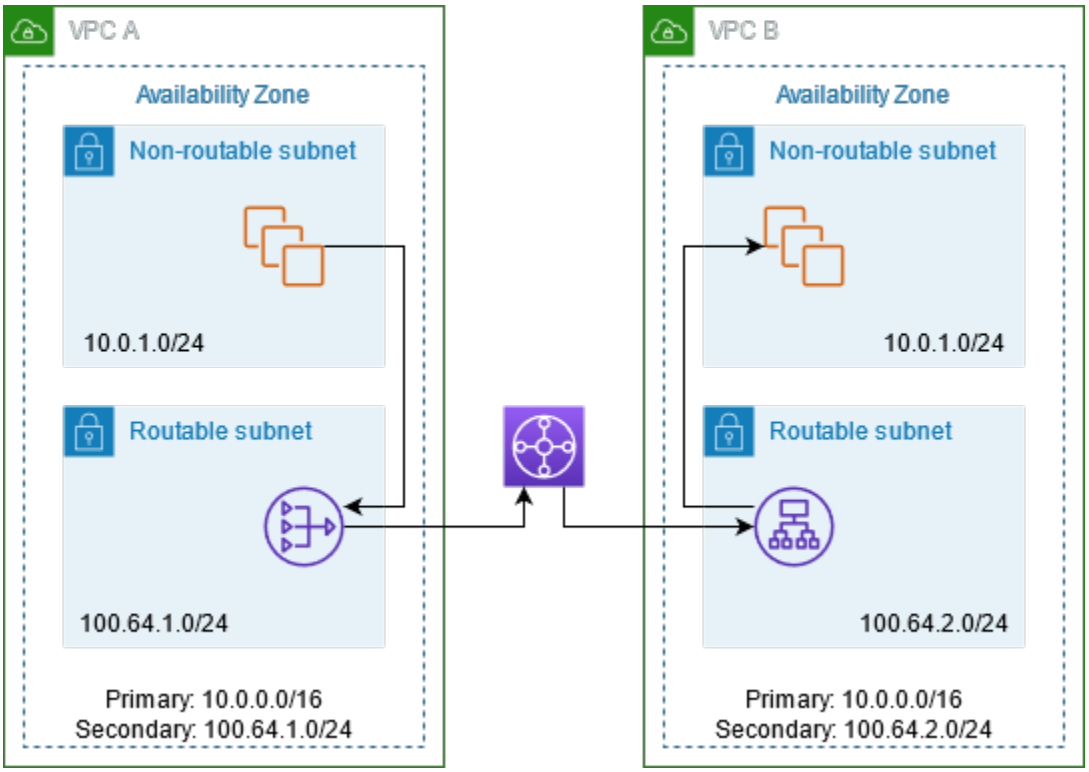
概要

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。まず、IP 管理チームは、重複できるアドレス範囲 (ルーティング不可能なアドレス範囲) と重複できないアドレス範囲 (ルーティング可能なアドレス範囲) を決定します。IP 管理チームは、ルーティング可能なアドレス範囲のプールから、アドレス範囲をリクエストに応じてプロジェクトに割り当てます。

各 VPC には、ルーティング不可能な元の IP アドレス範囲に加えて、IP 管理チームによって割り当てられたルーティング可能な IP アドレス範囲があります。VPC A には、プライベート NAT ゲートウェイを使用して、ルーティング可能な範囲からのサブネットがあります。プライベート NAT ゲートウェイは、サブネットから IP アドレスを取得します。VPC B には、Application Load Balancer を使用して、ルーティング可能な範囲からのサブネットがあります。Application Load Balancer は、サブネットから IP アドレスを取得します。

VPC A のルーティング範囲外のサブネットのインスタンスから VPC B のルーティング範囲外のサブネットのインスタンス宛のトラフィックは、プライベート NAT ゲートウェイを経由して送信されます。その後、トランジットゲートウェイにルーティングされます。

Transit Gateway は、トラフィックを Application Load Balancer に送信します。これにより、Application Load Balancer は、トラフィックを VPC B のルーティング範囲外のサブネットのターゲットインスタンスの 1 つにルーティングします。Transit Gateway からアプリケーションロードバランサーへのトラフィックには、プライベート NAT ゲートウェイの送信元 IP アドレスが含まれます。そのため、ロードバランサーからのレスポンストラフィックは、プライベート NAT ゲートウェイのアドレスを宛先として使用します。レスポンストラフィックは Transit Gateway に送信され、プライベート NAT ゲートウェイにルーティングされます。これにより、VPC A のルーティング範囲外のサブネットにあるインスタンスに宛先が変換されます。



リソース

次のようにリソースを作成または更新します。

- 割り当てられたルーティング可能な IP アドレスの範囲を、それぞれの VPC に関連付けます。
- ルーティング可能な IP アドレスの範囲から VPC A にサブネットを作成し、この新しいサブネットにプライベート NAT ゲートウェイを作成します。

- ルーティング可能な IP アドレスの範囲から VPC B にサブネットを作成し、この新しいサブネットに Application Load Balancer を作成します。ロードバランサーのターゲットグループに、ルーティング範囲外のサブネットのインスタンスを登録します。
- VPC 間を接続する Transit Gateway を作成します。ルート伝達は必ず無効にしてください。各 VPC を Transit Gateway に接続するときは、VPC のルーティング可能なアドレス範囲を使用します。
- VPC A のルーティング範囲外のサブネットのルートテーブルを更新して、VPC B のルーティング可能なアドレス範囲宛てのトラフィックをすべてプライベート NAT ゲートウェイに送信します。VPC A のルーティング可能なサブネットのルートテーブルを更新して、VPC B のルーティング可能なアドレス範囲宛てのトラフィックをすべて Transit Gateway に送信します。
- VPC B のルーティング可能なサブネットのルートテーブルを更新して、VPC A のルーティング可能なアドレス範囲宛てのトラフィックをすべて Transit Gateway に送信します。

ルーティング

以下は、VPC A のルーティング範囲外のサブネットのルートテーブルです。

ルーティング先	ターゲット
10.0.0.0/16	local
100.64.1.0/24	ローカル
100.64.2.0/24	*nat-gateway-id*

以下は、VPC A のルーティング可能なサブネットのルートテーブルです。

ルーティング先	ターゲット
10.0.0.0/16	local
100.64.1.0/24	ローカル
100.64.2.0/24	*transit-gateway-id*

以下は、VPC B のルーティング範囲外のサブネットのルートテーブルです。

ルーティング先	ターゲット
10.0.0.0/16	local
100.64.2.0/24	ローカル

以下は、VPC B のルーティング可能なサブネットのルートテーブルです。

ルーティング先	ターゲット
10.0.0.0/16	local
100.64.2.0/24	ローカル
100.64.1.0/24	*transit-gateway-id*

以下は、Transit Gateway のルートテーブルです。

CIDR	添付ファイル	ルートタイプ
100.64.1.0/24	*VPC A ########*	静的
100.64.2.0/24	*VPC B ########*	静的

DNS64 と NAT64

NAT ゲートウェイは、IPv6 から IPv4 へのネットワークアドレス変換をサポートします (一般的に NAT64 と言う)。NAT64 により、IPv6 AWS リソースが、オンプレミスネットワークまたはインターネット上で、同じ VPC または異なる VPC 内の IPv4 リソースと通信することが可能になります。Amazon Route 53 Resolver の DNS64 で NAT64 を使用することも、独自の DNS64 サーバーを使用することもできます。

内容

- DNS64 とは
- NAT64 とは
- DNS64 と NAT64 を設定する

DNS64 とは

VPC で実行される IPv6 専用ワークロードは、IPv6 ネットワークパケットのみを送受信できます。DNS64 を使用しない場合、IPv4 専用サービスの DNS クエリは IPv4 宛先アドレスを応答として生成するため、IPv6 専用サービスは IPv4 宛先アドレスと通信できません。この通信ギャップを埋めるために、サブネットの DNS64 を有効化して、それをサブネット内のすべての AWS リソースに適用します。DNS64 では、Amazon Route 53 Resolver はクエリしたサービスの DNS レコードを検索し、次のいずれかを実行します。

- レコードに IPv6 アドレスが含まれている場合は、元のレコードが返され、IPv6 を介した変換なしに接続が確立されます。
- DNS レコード内の宛先に関連付けられた IPv6 アドレスがない場合、Route 53 Resolver は、RFC6052 (`64:ff9b::/96`) で定義された既知の `/96` プレフィックスの先頭に付加して IPv6 アドレスを合成し、レコード内の IPv4 アドレスに送信します。

  IPv6 専用サービスは、ネットワークパケットを合成 IPv6 アドレスに送信します。次に、このトラフィックを NAT ゲートウェイ経由でルーティングします。NAT ゲートウェイは、サブネット内の IPv6 サービスがそのサブネット外の IPv4 サービスにアクセスできるように、トラフィックに対して必要な変換を実行します。

サブネットを選択し、[Actions] (アクション) > [Edit subnet settings] (サブネットの設定を編集する) の順に選択して、AWS CLI または VPC コンソールを使用する modify-subnet-attribute によって、サブネット上の DNS64 を有効または無効にすることができます。

NAT64 とは

NAT64 を使用すると、Amazon VPC 内の IPv6 専用サービスが、(異なるサブネット内の) 同じ VPC 内、接続された VPC 内、オンプレミスネットワーク内、またはインターネット経由で、IPv4 専用サービスと通信できます。

NAT64 は、既存の NAT ゲートウェイまたは作成した新しい NAT ゲートウェイで自動的に使用可能です。この機能を有効または無効にすることはできません。NAT ゲートウェイが配置されているサブネットは、NAT64 を機能させるためのデュアルスタックサブネットである必要はありません。

DNS64 を有効化した後、IPv6 専用サービスが NAT ゲートウェイを介して合成された IPv6 アドレスにネットワークパケットを送信すると、次のようになります。

- `64:ff9b::/96` プレフィックスから、NAT ゲートウェイは元の宛先が IPv4 であることを認識し、以下を置き換えて、IPv6 パケットを IPv4 に変換します:

  - インターネットゲートウェイによって Elastic IP アドレスに変換された独自のプライベート IP を持つソース IPv6。
  - `64:ff9b::/96` プレフィックスを切り捨てた IPv6 から IPv4 への宛先。
- NAT ゲートウェイは、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、Transit Gateway を介して変換された IPv4 パケットを宛先に送信し、接続を開始します。
- IPv4 専用ホストは IPv4 応答パケットを送り返します。接続が確立されると、NAT ゲートウェイは外部ホストからの応答 IPv4 パケットを受け入れます。
- 応答 IPv4 パケットの宛先は NAT ゲートウェイで、NAT ゲートウェイはパケットを受信し、その IP (宛先 IP) をホストの IPv6 アドレスに置き換え、`64:ff9b::/96` を送信元 IPv4 アドレスの先頭に付加することによって、パケットの NAT を元に戻します。その後、パケットはローカルルートに従ってホストに流れます。

このようにして、NAT ゲートウェイにより、サブネット内の IPv6 専用ワークロードが、サブネット外の IPv4 専用サービスと通信できるようになります。

DNS64 と NAT64 を設定する

このセクションのステップに従って DNS64 と NAT64 を設定し、IPv4 専用サービスとの通信を有効にします。

内容

- インターネット上の IPv4 専用サービスと AWS CLI の通信を有効にする
- オンプレミス環境で IPv4 専用サービスとの通信を有効にする

インターネット上の IPv4 専用サービスと AWS CLI の通信を有効にする

この例では、サブネット外の IPv4 専用サービスと通信する必要がある IPv6 専用ワークロードを持つサブネットがある場合に、IPv6 専用サービスを有効にして、インターネット上の IPv4 専用サービスと通信ができるかを示しています。

まず、(IPv6専用ワークロードを含むサブネットとは別に) パブリックサブネットに NAT ゲートウェイを設定する必要があります。例えば、NAT ゲートウェイを含むサブネットは、インターネットゲートウェイを指す `0.0.0.0/0` ルートを持つ必要があります。

これらの IPv6 専用サービスをインターネット上の IPv4 専用サービスの接続を有効にするには、次の手順を実行します。

1. IPv6 専用ワークロードを含むサブネットのルートテーブルに次の 3 つのルートを追加します。

   - NAT ゲートウェイを指す IPv4 ルート (存在する場合)。
   - 64:ff9b::/96NAT ゲートウェイを指す ルート。これにより、IPv4 専用サービスを宛先とする IPv6 専用ワークロードからのトラフィックを、NAT ゲートウェイ経由でルーティングできるようになります。
   - 出力専用インターネットゲートウェイ (またはインターネットゲートウェイ) を指す IPv6 ::/0 ルート。

   インターネットゲートウェイを指す ::/0 は、外部 IPv6 ホスト (VPC 外部) の IPv6 経由の接続開始を許可することに注意してください。

```
aws ec2 create-route --route-table-id rtb-34056078 --destination-cidr-block
0.0.0.0/0 --nat-gateway-id nat-05dba92075d71c408
```

```
aws ec2 create-route --route-table-id rtb-34056078 --destination-ipv6-cidr-block
64:ff9b::/96 --nat-gateway-id nat-05dba92075d71c408
```

```
aws ec2 create-route --route-table-id rtb-34056078 --destination-ipv6-cidr-block
::/0 --egress-only-internet-gateway-id eigw-c0a643a9
```

2. IPv6 専用ワークロードを含むサブネットで DNS64 機能を有効にします。

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --enable-dns64
```

これで、プライベートサブネット内のリソースは、インターネット上の IPv4 サービスと IPv6 サービスの両方とのステートフル接続を確立できます。セキュリティグループと NACL を適切に設定して、出力トラフィックと入力トラフィックを 64:ff9b::/96 トラフィックに許可します。

オンプレミス環境で IPv4 専用サービスとの通信を有効にする

Amazon Route 53 Resolver を使用すると、DNS クエリを VPC からオンプレミスネットワークに、またはその逆に転送することができます。これを行うには、次を実行します。

- VPC 内に Route 53 Resolver アウトバウンドエンドポイントを作成し、Route 53 Resolver がクエリを転送する送信元となる IPv4 アドレスを割り当てます。オンプレミス DNS リゾルバーの場合、これらは DNS クエリの送信元の IP アドレスであるため、IPv4 アドレスである必要があります。
- 1 つ以上のルールを作成し、オンプレミスリゾルバーに Route 53 Resolver から転送する DNS クエリのためのドメイン名を指定します。また、オンプレミスリゾルバーの IPv4 アドレスも指定します。
- Route 53 Resolver アウトバウンドエンドポイントを設定したので、IPv6 専用ワークロードを含むサブネットで DNS64 を有効にする必要があります。また、オンプレミスネットワーク宛てのデータを NAT ゲートウェイ経由でルーティングする必要があります。

オンプレミスネットワークの IPv4 専用の宛先に対する DNS64 の仕組み

1. VPC の Route 53 Resolver アウトバウンドエンドポイントに IPv4 アドレスを割り当てます。
2. IPv6 サービスからの DNS クエリは、IPv6 経由で Route 53 Resolver に送信されます。Route 53 Resolver は、クエリを転送ルールと照合し、オンプレミスリゾルバーの IPv4 アドレスを取得します。
3. Route 53 Resolver は、クエリパケットを IPv6 から IPv4 に変換し、アウトバウンドエンドポイントに転送します。エンドポイントの各 IP アドレスは、DNS リゾルバーのオンプレミス IPv4 アドレスにリクエストを転送する 1 つの ENI を表します。
4. オンプレミスリゾルバーは、IPv4 の応答パケットを、アウトバウンドエンドポイントを介して Route 53 Resolver に送信します。
5. Route 53 Resolver はクエリが DNS64 対応のサブネットから作成されたと仮定し、次の 2 つの処理を行います。

    a. 応答パケットの内容をチェックします。レコードに IPv6 アドレスがある場合、Route 53 Resolver はコンテンツをそのまま保持しますが、IPv4 レコードのみを含む場合は、`64:ff9b::/96` を IPv4 アドレスの先頭に付加して、同様に IPv6 レコードを合成します。

    b. コンテンツを再パッケージし、IPv6 経由で VPC 内のサービスに送信します。

NAT ゲートウェイからのトラフィックを検査する

Network Firewall Proxy を NAT ゲートウェイにアタッチして、NAT ゲートウェイのトラフィックを検査およびフィルタリングできます。このセキュリティコントロールにより、信頼できる境界外へのデータ漏洩を防ぎ、望ましくないインバウンドレスポンスをブロックできます。

仕組み

Network Firewall Proxy を作成するときは、プロキシをアタッチする既存の NAT ゲートウェイを選択する必要があります。作成されると、プロキシは以下のようになります。

- プロキシには完全修飾ドメイン名が付属しているため、http および https 接続リクエストをプロキシに送信するようにアプリケーションを設定する必要があります。プロキシはまず、お客様が入力したルールに基づいて接続リクエストのドメイン名をフィルタリングします。お客様が許可した場合、プロキシは DNS クエリを実行してドメインの IP アドレスを取得します。次に、最終送信先との TCP 接続を確立します。TLS 復号が有効かどうかに基づいて、プロキシは IP アドレスとヘッダー属性で TLS 接続をフィルタリングし、IP とヘッダー属性 (ヘッダーアクションと URL パスを含む) がポリシーで許可されている場合にのみ、送信先との TLS 接続を確立します。
- アプライアンスはトラフィックを検査し、フィルタリングします。
- 許可されたトラフィックは送信先 (インターネット、オンプレミス環境、または別の VPC) に継続されます。

アプライアンスのアタッチ

アプライアンスは AWS Network Firewall を介して NAT ゲートウェイにアタッチされます。アプライアンスを作成してアタッチするステップについては、「Network Firewall Proxy デベロッパーガイド」を参照してください。

アタッチされたアプライアンスの表示

NAT ゲートウェイにアタッチされたアプライアンスを表示するには、describe-nat-gateways コマンドを使用します。

```
aws ec2 describe-nat-gateways --nat-gateway-ids nat-1234567890abcdef0
```

レスポンスには、以下の内容を示す `AttachedAppliances` フィールドが含まれます。

- タイプ – アプライアンスタイプ (例: `network-firewall-proxy`)

- ApplianceArn – アタッチされたアプライアンスの ARN
- AttachmentState – 現在のアタッチメントステータス (`attached`、`detaching`、`detached`、`attach_failed`、`detach_failed`)
- ModificationState – 現在の変更ステータス (`modifying`、`completed`、`failed`)
- VpcEndpointId – 検査とフィルタリングのためにアプリケーション VPC からプロキシにトラフィックをルーティングするために使用される VPC エンドポイント ID
- FailureCode – アプライアンスのアタッチメントまたは変更オペレーションが失敗した場合の失敗コード
- FailureMessage – アプライアンスのアタッチメントまたは変更操作が失敗した場合の失敗を説明するメッセージ

Amazon CloudWatch による 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。これらのメトリクスを使用すれば、NAT ゲートウェイの状態とパフォーマンスを可視化し、そのオペレーションを詳細にモニタリングしたり問題が発生したときにはすばやくトラブルシューティングしたりすることができます。

CloudWatch によって収集される NAT ゲートウェイのメトリクスには、処理済みのバイト数、パケット数、接続数、エラー率などのデータポイントが含まれます。これにより、NAT ゲートウェイを通過するトラフィックを詳細に理解し、異常やボトルネックがないか確認することができます。CloudWatch はこのメトリクスデータを 1 分間隔で配信するため、NAT ゲートウェイの動作の最新の状況を画面上で詳細に把握できます。

さらに、CloudWatch ではこの NAT ゲートウェイのメトリクスのデータが 15 か月間にわたって保持されるため、データの傾向やパターンを経時的に分析することができます。この履歴データは、キャパシティプランニング、パフォーマンスの最適化、そして NAT ゲートウェイの使用における長期的な変化の理解に役立ちます。

このような強力なモニタリング機能を使用するには、特定のニーズに合わせて調整された CloudWatch のカスタムのダッシュボードとアラームを作成します。例えば、NAT ゲートウェイのアウトバウンドのデータ転送が特定のしきい値を超えた場合にアラートで通知するように設定すると、潜在的な帯域幅の制約に事前に対処することができます。

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

内容

- 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 CloudWatch メトリクスの表示
- 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

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。説明の列には、各メトリクスの説明に加え、単位と統計が含まれています。

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。

メトリクス	説明
	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。
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。

メトリクス	説明
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。これには、最初の SYN のみが含まれます。場合によっては、SYN の再送信により、ConnectionAttemptCount が ConnectionEstablishedCount よりも低くなることがあります。 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
ConnectionEstablishedCount	NAT ゲートウェイ経由で確立された接続の数。これには、SYN と SYN の再送信が含まれます。 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。
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。

メトリクス	説明
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: 数 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。 ドロップされたパケットの数をパケットトラフィック全体のパーセンテージとして計算するには、PacketsDropCount/(PacketsInFromSource+PacketsInFromDestination)*100 の式を使用します。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、Amazon VPC サービスに問題がある可能性があります。AWS サービスヘルスダッシュボードを使用して、NAT ゲートウェイがパケットをドロップする原因となっているサービスの問題を特定します。 単位: カウント 統計: 最も有用な統計は Sum です。

メトリクス	説明
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: 数 統計: 最も有用な統計は Sum です。
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: 数 統計: 最も有用な統計は Sum です。

メトリクス	説明
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: 数 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: 数 統計: 最も有用な統計は Sum です。

メトリクス	説明
PeakBytesPerSecond	このメトリックは、特定の 1 分間においてバイト数が最大である 10 秒間の 1 秒あたりの平均値を報告します。 単位: カウント 統計: 最も有用な統計は Maximum です。
PeakPacketsPerSecond	このメトリクスは、60 秒間、10 秒ごとに平均パケットレート (1 秒あたりに処理されるパケット) を計算し、その 6 つのレートの最大値 (最高平均パケットレート) を報告します。 単位: カウント 統計: 最も有用な統計は Maximum 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。ゾーン NAT ゲートウェイは、このディメンションのみを使用します。リージョン NAT ゲートウェイは、このディメンションを AvailabilityZone とともに使用します。
AvailabilityZone	アベイラビリティーゾーン別にメトリクスデータをフィルタリングします。リージョン NAT ゲートウェイは、このディメンションを NatGatewayId とともに使用します。ゾーン NAT ゲートウェイは、このディメンションを使用しません。

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 コンソールの https://console.aws.amazon.com/cloudwatch/ を開いてください。
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、[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。[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 コンソールの https://console.aws.amazon.com/cloudwatch/ を開いてください。
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、[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。[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
- NAT ゲートウェイクォータ
- Elastic IP アドレスのクォータ
- アベイラビリティーゾーンがサポートされていない
- NAT ゲートウェイが表示されなくなりました
- NAT ゲートウェイが ping コマンドに応答しない
- インスタンスがインターネットにアクセスできない
- 送信先への TCP 接続が失敗する
- 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
- インターネット接続が 350 秒後に中断される
- IPsec 接続を確立できない
- 追加の接続を開始できない

NAT ゲートウェイの作成に失敗する

問題

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。

原因

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。

ソリューション

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、「サブネットへのインターネットアクセスを追加する」を参照してください。
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

表示されるエラー	原因	ソリューション
		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。(詳しくは、「Elastic IP アドレスの使用を開始する」を参照してください。)

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
```

原因

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。

ソリューション

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、「Amazon VPC クォータ」を参照してください

Elastic IP アドレスのクォータ

問題

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.
```

原因

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。

ソリューション

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

問題

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。

原因

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。

ソリューション

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

問題

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。

原因

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。

ソリューション

「NAT ゲートウェイの作成に失敗する」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

問題

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。

原因

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。

ソリューション

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」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

問題

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。

原因

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。

ソリューション

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、「NAT ゲートウェイの作成に失敗する」を参照してください
- 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、「VPC フローログを使用した IP トラフィックのログ記録」を参照してください
- 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

問題

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。

原因

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、「NAT ゲートウェイと NAT インスタンスの比較」を参照してください
- 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。

ソリューション

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

問題

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。

原因

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。

ソリューション

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

問題

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。

原因

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)。

ソリューション

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

問題

送信先への IPsec 接続を確立できません。

原因

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。

ソリューション

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

問題

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。

原因

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、「NAT ゲートウェイの基本」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。

ソリューション

次のいずれかを行います。

- 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、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` 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。
- 複数の IP アドレスで NAT ゲートウェイを作成するか、既存の NAT ゲートウェイにセカンダリ IP アドレスを追加します。新しい IPv4 アドレスはそれぞれ最大 55,000 の同時接続をサポートできます。詳細については「NAT ゲートウェイを作成する」または「セカンダリ IP アドレスの関連付けを編集する」を参照してください。

NAT ゲートウェイの料金

NAT ゲートウェイをプロビジョンすると、NAT ゲートウェイが使用可能な時間 1 時間ごと、および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全体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してください。または、リソースがある各アベイラビリティーゾーンに NAT ゲートウェイを作成します。
- 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。

NAT インスタンス

NAT インスタンスはネットワークアドレス変換 (NAT) を提供します。NAT インスタンスを使用すると、プライベートサブネット内のリソースが、インターネットやオンプレミスネットワークなどの仮想プライベートクラウド (VPC) 外部の宛先と通信できます。プライベートサブネット内のリソース

は、インターネットへのアウトバウンド IPv4 トラフィックを開始できますが、インターネット上で開始されたインバウンドトラフィックを受信することはできません。

> **Important**
>
> NAT AMI は、2020 年 12 月 31 日に標準サポートが終了し、2023 年 12 月 31 日にメンテナンスサポートが終了した Amazon Linux AMI 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。
> 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。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」を参照してください。
> NAT インスタンスの方が NAT ゲートウェイよりもユースケースに適している場合は、the section called “3. NAT AMI を作成する” で説明されているように Amazon Linux の現行バージョンから独自の NAT AMI を作成できます。

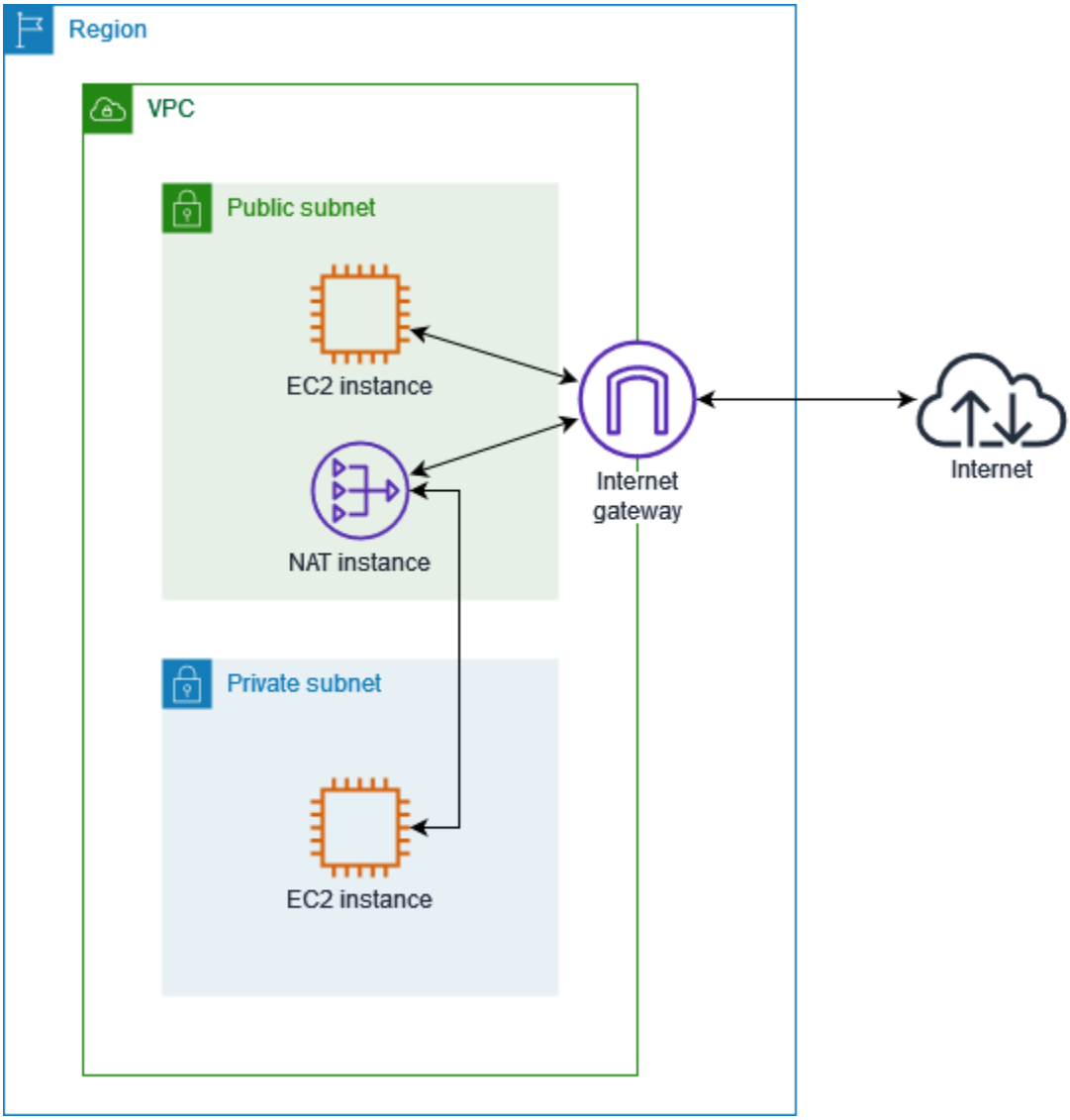
内容

- NAT インスタンスの基本
- プライベートリソースが VPC の外側で通信できるようにする

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。プライベートサブネットと関連付けられたルートテーブルは、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 インスタンスにインターネット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 インスタンスのパブリック IP アドレスに関連付けられます。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。

NAT インスタンスにはインターネットアクセスが必要なため、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がある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 インスタンスを使い始めるには、NAT AMI を作成し、NAT インスタンスのセキュリティグループを作成して、NAT インスタンスを VPC で起動します。

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。詳細については、「AWS 全般のリファレンス」の「Amazon EC2 Service Quotas」を参照してください。

プライベートリソースが VPC の外側で通信できるようにする

このセクションでは、NAT インスタンスを作成し、これを使ってプライベートサブネット内のリソースが仮想プライベートクラウドの外部で通信できるようにする方法について説明します。

タスク

- 1. NAT インスタンス用の VPC を作成する
- 2. NAT インスタンスのセキュリティグループを作成する
- 3. NAT AMI を作成する
- 4. NAT インスタンスの作成
- 5. 送信元/送信先チェックを無効にする
- 6. ルートテーブルを更新する
- 7. NAT インスタンスをテストする

1. NAT インスタンス用の VPC を作成する

次の手順を使用して、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。

VPC を作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. [Create VPC (VPC の作成)] を選択します。
3. [Resources to create] (作成するリソース) で、[VPC and more] (VPC など) を選択します。
4. [名前タグの自動生成] に、VPC の名前を入力します。
5. サブネットを設定するには、次の操作を行います。
    a. [アベイラビリティーゾーンの数] で、ニーズに応じて [1] または [2] を選択します。
    b. [パブリックサブネットの数] で、アベイラビリティーゾーンごとに 1 つのパブリックサブネットがあることを確認します。
    c. [Number of private subnets] (プライベートサブネットの数) で、アベイラビリティーゾーンごとに 1 つのプライベートサブネットがあることを確認します。
6. [Create VPC (VPC の作成)] を選択します。

2. NAT インスタンスのセキュリティグループを作成する

次の表で説明されているルールを使用してセキュリティグループを作成します。これらのルールにより、NAT インスタンスは、ネットワークからの SSH トラフィックだけでなく、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に向かうトラフィックも受信できるようになります。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。

推奨されるインバウンドルールを次に示します。

ソース	プロトコル	ポート範囲	コメント
########## CIDR	TCP	80	プライベートサブネットのサーバーからのインバウンド HTTP トラフィックを許可する
########## CIDR	TCP	443	プライベートサブネットのサーバーからのインバウンド HTTPS トラフィックを許可する
########## IP ##### ##	TCP	22	ネットワークから NAT 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

推奨されるアウトバウンドルールを次に示します。

送信先	プロトコル	ポート範囲	コメント
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

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 [セキュリティグループ] をクリックします。
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。
5. [VPC] には、NAT インスタンスの VPC の ID を選択します。
6. 次のように、[インバウンドルール] でインバウンドトラフィックのルールを追加します。
    a. [ルールの追加] を選択します。[種類] で [HTTP] を選択し、[送信元] に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。

b. [ルールの追加] を選択します。[種類] で [HTTPS] を選択し、[送信元] に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。

c. [ルールの追加] を選択します。[種類] に [SSH] を選択し、[送信元] にネットワークの IP アドレス範囲を入力します。

7. 次のように、[アウトバウンドルール] でアウトバウンドトラフィックのルールを追加します。

a. [ルールの追加] を選択します。[種類] に [HTTP] を選択し、[送信先] に「0.0.0.0/0」と入力します。

b. [ルールの追加] を選択します。[種類] に [HTTPS] を選択し、[送信先] に「0.0.0.0/0」と入力します。

8. [セキュリティグループの作成] を選択します。

詳細については、「セキュリティグループ」を参照してください。

3. NAT AMI を作成する

NAT AMI は、EC2 インスタンスで NAT を実行するように設定されます。NAT AMI を作成した後、NAT AMI を使用して NAT インスタンスを起動する必要があります。

NAT AMI に Amazon Linux 以外のオペレーティングシステムを使用する予定の場合は、そのオペレーティングシステムのドキュメントを参照して NAT の設定方法を確認します。インスタンスの再起動後も維持されるように、これらの設定を保存するようにしてください。

Amazon Linux 用の NAT AMI を作成するには

1. AL2023 または Amazon Linux 2 を実行する EC2 インスタンスを起動します。必ず NAT インスタンス用に作成したセキュリティグループを指定してください。
2. インスタンスに接続し、次のコマンドをそのインスタンスで実行して、iptables を有効にします。

```
sudo yum install iptables-services -y
sudo systemctl enable iptables
sudo systemctl start iptables
```

3. 再起動後も維持されるように IP 転送を有効にするには、インスタンスで次の操作を行います。

a. テキストエディタ (nano や vim) を使用し、設定ファイル (`/etc/sysctl.d/custom-ip-forwarding.conf`) を作成します。

b. 設定ファイルに次の行を追加します。

```
net.ipv4.ip_forward=1
```

c. 設定ファイルを保存し、テキストエディタを終了します。

d. 次のコマンドを実行し、設定ファイルを適用します。

```
sudo sysctl -p /etc/sysctl.d/custom-ip-forwarding.conf
```

4. インスタンスで次のコマンドを実行して、プライマリネットワークインターフェイスの名前を書き留めます。この情報は、次のステップで必要になります。

```
netstat -i
```

次の出力例では、docker0 は Docker が作成したネットワークインターフェイス、eth0 はプライマリネットワークインターフェイス、lo はループバックインターフェイスです。

```
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
docker0   1500        0      0      0 0             0      0      0      0 BMU
eth0      9001  7276052      0      0 0       5364991      0      0      0 BMRU
lo       65536   538857      0      0 0        538857      0      0      0 LRU
```

次の出力例では、プライマリネットワークインターフェイスは enX0 です。

```
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
enX0      9001     1076      0      0 0          1247      0      0      0 BMRU
lo       65536       24      0      0 0            24      0      0      0 LRU
```

次の出力例では、プライマリネットワークインターフェイスは ens5 です。

```
Iface      MTU    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
ens5      9001    14036      0      0 0          2116      0      0      0 BMRU
lo       65536       12      0      0 0            12      0      0      0 LRU
```

5. 次のコマンドをインスタンスで実行して、NAT を設定します。プライマリネットワークインターフェイスが eth0 でない場合は、*eth0* を、前のステップでメモしたプライマリネットワークインターフェイスに置き換えます。

```
sudo /sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

```
sudo /sbin/iptables -F FORWARD
sudo service iptables save
```

6. EC2 インスタンスから NAT AMI を作成します。詳細については、「Amazon EC2 ユーザーガイド」の「インスタンスからの Linux AMI の作成」を参照してください。

4. NAT インスタンスの作成

以下の手順に従って、作成した VPC、セキュリティグループ、NAT AMI を使用し、NAT インスタンスを起動します。

NAT インスタンスを起動するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ダッシュボードで、[Launch Instance (インスタンスの起動)] を選択してください。
3. [名前] に、NAT インスタンスの名前を入力します。
4. [アプリケーションと OS イメージ] で、NAT AMI を選択します ([その他の AMI を参照]、[My AMI] を選択します)。
5. [インスタンスタイプ] で、NAT インスタンスに必要なコンピューティング、メモリ、ストレージリソースを持つインスタンスタイプを選択します。
6. [キーペア] で、既存のキーペアを選択するか、[新しいキーペアの作成] を選択します。
7. [Network settings] (ネットワーク設定) で、次の操作を行います。
   a. [編集] を選択します。
   b. [VPC] で、作成した VPC を選択します。
   c. [サブネット] で、作成済みのパブリックサブネットを選択します。
   d. [Auto-assign public IP] (パブリック IP の自動割り当て) で、[Enable] (有効化) を選択します。または、NAT インスタンスを起動した後、Elastic IP アドレスを割り振り、それを NAT インスタンスに割り当てます。
   e. [ファイアウォール] で [既存のセキュリティグループの選択] を選択してから、作成済みのセキュリティグループを選択します。
8. [インスタンスを起動] を選択してください。インスタンス ID を選択して、インスタンスの詳細ページを開きます。インスタンスの状態が [実行中] に変わり、ステータスチェックが成功するまで待ちます。
9. NAT インスタンスの送信元/送信先チェックを無効にします (5. 送信元/送信先チェックを無効にする を参照)。

10. 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 (6. ルートテーブルを更新する を参照)。

5. 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。

送信元/送信先チェックを無効にするには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[インスタンス] を選択してください。
3. NAT インスタンスを選択します。
4. [アクション]、[ネットワーキング]、[送信元/送信先チェックの変更] の順にクリックします。
5. [送信元/宛先を確認] には、[停止] を選択します。
6. [保存] を選択します。
7. 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。

6. ルートテーブルを更新する

プライベートサブネットのルートテーブルには、インター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が必要です。

ルートテーブルを更新するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[Route tables] (ルートテーブル) を選択します。
3. プライベートサブネットのルートテーブルを選択します。
4. [ルート] タブで、[ルートの編集] 、[ルートの追加] の順に選択します。

5. [送信先] には「0.0.0.0/0」、[ターゲット] には NAT インスタンスのインスタンス ID を入力します。
6. [Save changes] (変更の保存) をクリックします。

詳細については、「ルートテーブルを設定する」を参照してください。

7. NAT インスタンスをテストする

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、テストを行い、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します。

タスク

- ステップ 1: NAT インスタンスのセキュリティグループを更新する
- ステップ 2: プライベートサブネット内でインスタンスを起動する
- ステップ 3: ICMP が有効なウェブサイトに ping を送信する
- ステップ 4: クリーンアップする

ステップ 1: NAT インスタンスのセキュリティグループを更新する

プライベートサブネットのインスタンスが NAT インスタンスに ping トラフィックを送信できるようにするには、インバウンドとアウトバウンドの ICMP トラフィックを許可するルールを追加します。NAT インスタンスを踏み台サーバーとして機能させるには、プライベートサブネットへのアウトバウンド SSH トラフィックを許可するルールを追加します。

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 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] をクリックします。
3. NAT 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チェックボックスをオンにします。
4. [インバウンドルール] タブで、[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。[送信元] で [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。[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。[送信先] で [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。
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。[ルールの保存] を選択します。

ステップ 2: プライベートサブネット内でインスタンスを起動する

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。NAT インスタンスからの SSH アクセスを許可する必要があります。また、NAT インスタンスに使用したのと同じキーペアを使用する必要があります。

プライベートサブネット内でテストインスタンスを起動するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ダッシュボードで、[Launch Instance (インスタンスの起動)] を選択してください。
3. プライベートサブネットを選択します。
4. この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。
5. このインスタンスのセキュリティグループが、NAT インスタン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インバウンド SSH アクセス、およびアウトバウンド ICMP トラフィックを許可していることを確認します。
6. 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。

ステップ 3: ICMP が有効なウェブサイトに ping を送信する

プライベートサブネットのテストインスタンスが NAT インスタンスを使用してインターネットと通信できることを検証するには、ping コマンドを実行します。

プライベートインスタンスからインターネット接続をテストするには

1. ローカルコンピュータから SSH エージェント転送を設定して、NAT インスタンスを踏み台サーバーとして使用できるようにします。

   Linux and macOS

   ```
   ssh-add key.pem
   ```

Windows

まだインストールされていない場合は、Pageant をダウンロードしてインストールします。

PuTTYgen を使用してプライベートキーを変換する。

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) のコンテキストメニューを開き、[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。

2. ローカルコンピュータから、NAT インスタンスに接続します。

Linux and macOS

```
ssh -A ec2-user@nat-instance-public-ip-address
```

Windows

PuTTY を使用して NAT インスタンスに接続します。[Auth] については、[Allow agent forwarding] を選択します。[Private key file for authentication] は空白のままにしてください。

3. NAT インスタンスから、ICMP が有効なウェブサイトを指定して、ping コマンドを実行します。

```
[ec2-user@ip-10-0-4-184]$ ping ietf.org
```

NAT 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するため、次のような出力を受け取ったことを確認してから、Ctrl+C キーを押して ping コマンドをキャンセルします。または、NAT インスタンスがパブリックサブネットにある (その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) ことを確認します。

```
PING ietf.org (104.16.45.99) 56(84) bytes of data.
64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=1 ttl=33 time=7.88 ms
64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=2 ttl=33 time=8.09 ms
64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=3 ttl=33 time=7.97 ms
...
```

4. 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。

```
[ec2-user@ip-10-0-4-184]$ ssh ec2-user@private-server-private-ip-address
```

5. 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。

   ```
   [ec2-user@ip-10-0-135-25]$ ping ietf.org
   ```

   プライベートインスタンスが NAT インスタンスを介してインターネットにアクセスできることを確認するため、次のような出力を受け取ったことを確認してから、Ctrl+C キーを押して ping コマンドをキャンセルします。

   ```
   PING ietf.org (104.16.45.99) 56(84) bytes of data.
   64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=1 ttl=33 time=8.76 ms
   64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=2 ttl=33 time=8.26 ms
   64 bytes from 104.16.45.99 (104.16.45.99): icmp_seq=3 ttl=33 time=8.27 ms
   ...
   ```

トラブルシューティング

プライベートサブネットのサーバーから ping コマンドが失敗した場合は、次の手順を使用して問題のトラブルシューティングを行います。

- ICMP が有効な Web サイトに ping を実行できたことを確認します。そうでない場合、サーバーは応答パケットを受信できません。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。
- NAT インスタンスのセキュリティグループで、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。そうでない場合、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。
- 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、「5. 送信元/送信先チェックを無効にする」を参照してください。
- 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、「6. ルートテーブルを更新する」を参照してください。

ステップ 4: クリーンアップする

プライベートサブネットのテストサーバーが不要になった場合、インスタンスを終了し、今後料金を請求されないようにします。詳細については、「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 インスタンスが不要になった場合は、そのインスタンスを停止または終了して、料金を請求されないようにできます。NAT AMI を作成した場合は、必要なときにいつでも新しい NAT インスタンスを作成できます。

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
可用性	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	100 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	AWS によって管理されます。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。

属性	NAT ゲートウェイ	NAT インスタンス
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。

属性	NAT ゲートウェイ	NAT インスタンス
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。
トラフィックのメトリクス	NAT ゲートウェイの CloudWatch メトリクスを表示します。	インスタンスの CloudWatch メトリクスを表示します。
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。

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。

Elastic IP アドレスを VPC 内のリソースに関連付ける

Elastic IP アドレスは、動的なクラウドコンピューティング専用に設計された静的なパブリック IPv4 アドレスです。この機能を使用すると、AWS アカウントの任意の仮想プライベートクラウド (VPC) 内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 Elastic IP アドレスを関連付けることができます。Elastic IP アドレスを使用することで、クラウドベースのインフラストラクチャの管理およびレジリエンスを簡素化する多数のメリットが得られます。

Elastic IP アドレスの主な利点の一つは、インスタンスの障害をマスキングできることです。インスタンスで計画外の停止が発生したり、インスタンスの交換が必要になったりした場合は、関連付けられた Elastic IP アドレスを VPC 内の別のインスタンスに再マッピングすることができます。このフェイルオーバープロセスにより、アプリケーションとサービスで、一貫性のある信頼性の高いパブリックエンドポイントを維持し、ダウンタイムを最小限に抑えて、優れたユーザーエクスペリエンスを提供することができます。

さらに、Elastic IP アドレスは、ネットワークリソースの管理方法における柔軟性を高めます。これらのアドレスは必要に応じてプログラム的に関連付けたり関連付けを解除したりできるため、トラフィックを、ビジネス要件の変化に応じて別のインスタンスに転送することが可能となります。このようにパブリック IP アドレスを動的に割り当てることで、需要の変化に柔軟に対応し、インフラストラクチャを拡張し、静的 IP の割り当ての制約なしに革新的なアーキテクチャを実装することが可能になります。

インスタンスのフェイルオーバーに使用できるだけでなく、Elastic IP アドレスはクラウドベースのリソースの安定した識別子としても機能します。こちらは、DNS レコードやファイアウォールルールなど外部のサービスを設定して、AWS がホストするアプリケーションと通信するときに有用です。永続的なパブリック IP アドレスを関連付けることで、ネットワーク設定を将来にわたって保護することができ、基盤となるインスタンスを交換したり拡張したりする際には外部参照を更新する必要がなくなります。

内容

- Elastic IP アドレスの概念とルール
- Elastic IP アドレスの使用を開始する

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、「Amazon EC2 ユーザーガイド」の「Elastic Network Interface」を参照してください。

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 をインスタンスのプライマリ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プライマリ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、「NAT デバイスを使用してインターネットまたは他のネットワークに接続する」を参照してください
- 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、「セキュリティグループ」および「ネットワーク ACL」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、「Amazon EC2 ユーザーガイド」の「自分の IP アドレス (BYOIP) を持ち込む」を参照してください。
- パブリック IPv4 アドレスはコスト配分タグをサポートします。Elastic IP アドレスにタグを適用する場合、これらのタグを使用して、AWS Cost Explorer でパブリック IPv4 アドレスのコストを追跡できます。

  タグをコスト配分タグとして使用する前に、タグをアクティブ化する必要があります。詳細については、AWS Billing ユーザーガイドの「ユーザー定義のコスト配分タグのアクティブ化」を参照してください。ユーザー定義のタグを作成してリソースに適用した後、アクティブ化のためにタグキーがコスト配分タグページに表示されるまでに最大で 24 時間かかる場合があることに留意してください。

  コスト配分タグがアクティブ化されると、次のようになります。

  - Elastic Network Interface に関連付けられているすべてのパブリック IPv4 アドレス (EC2 インスタンスに割り当てられたパブリック IPv4 アドレスと Elastic IP アドレスを含む) については、Cost Explorer で [使用タイプ] > [PublicIPv4InUseAddress (時間)] を選択して、パブリック IPv4 アドレスに関連付けられているコストを表示できます。
  - タグ付けされた Elastic IP アドレスが ENI に関連付けられていない場合、または停止したリソース (停止した EC2 インスタンスなど) に関連付けられている場合、アイドル状態の IPv4 アドレスとみなされます。Cost Explorer で [使用タイプ] > [PublicIPv4IdleAddress (時間)] を選択すると、アイドル状態の IPv4 アドレスに関連するコストを表示できます。

  Cost Explorer の詳細については、「AWS Billing ユーザーガイド」の「AWS Cost Explorer を使用したコストの分析」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 アドレスの料金について、詳細は「Amazon VPC の料金」の [パブリック IPv4 アドレス] タブを参照してください。

Elastic IP アドレスの使用を開始する

以下のセクションでは、Elastic IP アドレスの使用開始方法について説明します。

タスク

- 1. Elastic IP アドレスを割り当てる
- 2. Elastic IP アドレスの関連付け
- 3. Elastic IP アドレスの関連付けを解除する
- 4. Elastic IP アドレスの移転
- 5. Elastic IP アドレスをリリース
- 6. Elastic IP アドレスの復元
- コマンドラインの概要

1. 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。

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. [Elastic IP アドレスの割り当て] を選択してください。
4. (オプション) Elastic IP アドレス (EIP) を割り当てるときはEIP を割り当てるネットワークボーダーグループを選択してください。ネットワークボーダーグループはAWS がパブリック IP アドレスをアドバタイズするアベイラビリティーゾーン (AZ)、Local Zones、または Wavelength Zones のコレクションです。Local Zones と Wavelength Zones はAWS ネットワークとこれらのゾーンのリソースにアクセスする顧客との間のレイテンシーや物理的距離を最小限に抑えるために、リージョン内の AZ とは異なるネットワークボーダーグループを持つ場合があります。

> **Important**
>
> EIP に関連付ける AWS リソースと同じネットワークボーダーグループに EIP を割り当てる必要があります。あるネットワークボーダーグループ内の EIP はそのネットワークボーダーグループ内のゾーンでのみアドバタイズでき、他のネットワークボーダーグループで表される他のゾーンではアドバタイズできません。

Local Zones または Wavelength Zones を有効にしている場合 (詳細については「Local Zone を有効にする」または「Wavelength Zones を有効にする」を参照)、AZ、Local Zones、または Wavelength Zones のネットワークボーダーグループを選択できます。EIP とそれが関連付けられている AWS リソースは同じネットワークボーダーグループに属している必要があるため、ネットワークボーダーグループは慎重に選択してください。EC2 コンソールを使用して、アベイラビリティーゾーン、Local Zones、または Wavelength Zones が属するネットワークボーダーグループを表示できます (「Local Zones」を参照)。通常、リージョン内のすべてのアベイラビリティーゾーンは同じネットワークボーダーグループに属しますが、Local Zones や Wavelength Zones はそれぞれ別のネットワークボーダーグループに属します。

Local Zones または Wavelength Zones が有効になっていない場合、EIP を割り当てると、リージョン (`us-west-2` など) のすべての AZ を表すネットワークボーダーグループが定義済みになり、変更することはできません。つまり、このネットワークボーダーグループに割り当てた EIP は現在のリージョンのすべての AZ でアドバタイズされます。

5. [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 を使用していない場合、このオプションは使用できません。

6. (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。

7. [割り当て] を選択してください。

2. 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、「VPC の DNS 属性」を参照してください

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。

3. 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。

4. Elastic IP アドレスの移転

このセクションでは、Elastic IP アドレスを ある AWS アカウント から別のアカウントに転送する方法について説明します。Elastic IP アドレスの移管は、次のような状況で役に立ちます。

- 組織の再構築 - Elastic IP アドレス転送を使用すると、ある AWS アカウント から別のアカウントにワークロードをすばやく移動できます。新しい Elastic IP アドレスがセキュリティグループと NACL の許可リストに追加されるのを待つ必要がありません。
- 一元的なセキュリティ管理 - 一元化された AWS セキュリティアカウントを使用して、セキュリティコンプライアンスのために精査された Elastic IP アドレスを追跡および移管できます。
- ディザスタリカバリ – 緊急時には、Elastic IP アドレス移管を使用することで、一般向けインターネットワークロードの IP アドレスをすばやく再マッピングできます。

Elastic IP アドレスの移管には料金はかかりません。

タスク

- Elastic IP アドレスの移管を有効にする
- Elastic IP アドレス転送を無効にする
- 転送された Elastic IP アドレスを承諾する

Elastic IP アドレスの移管を有効にする

このセクションでは移管された Elastic IP アドレスを承諾する方法について説明します。Elastic IP アドレスの移管を有効にする際には以下の制限に注意してください。

- 任意の AWS アカウント (ソースアカウント) から同じ AWS リージョン内の他の AWS アカウント (転送先アカウント) に Elastic IP アドレスを転送できます。
- Elastic IP アドレスを転送する場合、AWS アカウント の間で 2 段階のハンドシェイクが行われます。ソースアカウントが移管を開始してから 7 日間は転送先アカウントが Elastic IP アドレス移管を受け入れることができます。この 7 日間、ソースアカウントは保留中の移管を (AWS コンソールや AWS CLI コマンドの describe-address-transfers などを使用して) 確認できます。7 日後、移管の有効期限が切れ、Elastic IP アドレスの所有権がソースアカウントに戻ります。
- 移管が受け入れられてから 14 日間、ソースアカウントは受け入れられた移管を (AWS コンソールや AWS CLI コマンド describe-address-transfers などを使用して) 表示できます。
- AWS は保留中の Elastic IP アドレス転送リクエストについて、転送先アカウントに通知しません。ソースアカウントの所有者は承諾する必要がある Elastic IP アドレス転送リクエストがあることを転送先アカウントの所有者に通知する必要があります。
- 転送中の Elastic IP アドレスに関連付けられているタグは転送が完了するとリセットされます。
- AWS アカウント に持ち込んだパブリック IPv4 アドレスプール (一般的に Bring-Your-Own-IP (BYOIP) アドレスプールと呼ばれる) から割り当てられた Elastic IP アドレスは転送できません。

- 逆引き DNS レコードが関連付けられている Elastic IP アドレスを転送する場合、転送プロセスを開始できます。ただし、関連付けられた DNS レコードが削除されるまで、転送先アカウントは転送を受け入れることができません。
- AWS Outposts を有効にして設定している場合はカスタマー所有の IP アドレスプール (CoIP) から Elastic IP アドレスを割り当てている可能性があります。CoIP から割り当てられた Elastic IP アドレスを転送することはできません。ただし、AWS RAM を使用して CoIP を別のアカウントと共有することはできます。CoIP の詳細についてはAWS Outposts ユーザーガイドの「カスタマー所有 IP アドレス」を参照してください 。
- Amazon VPC IPAM を使用して、AWS Organizations から組織内のアカウントへの Elastic IP アドレスの転送を追跡することができます。詳細については「IP アドレスの履歴の表示」を参照してください。Elastic IP アドレスが組織外の AWS アカウント に転送されると、その Elastic IP アドレスの IPAM 監査履歴は失われます。

これらのステップはソースアカウントで実行する必要があります。

Elastic IP アドレスの移管を有効にするには

1. ソースの AWS アカウントを使用していることを確認してください。
2. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
3. ナビゲーションペインで [Elastic IP] を選択します。
4. 移管を有効にする Elastic IP アドレスを 1 つ以上選択し、[Action] (アクション)、[Enable transfer] (移管を有効にする) を選択してください。
5. 複数の Elastic IP アドレスを移管する場合は [転送タイプ] オプションが表示されます。以下のオプションのいずれかを選択してください。
   - Elastic IP アドレスを単一の AWS アカウントに移管する場合は[Single account] (単一アカウント) を選択してください。
   - Elastic IP アドレスを複数の AWS アカウントに移管する場合は[Multiple accounts] (複数アカウント) を選択してください。
6. [Transfer account ID] (アカウント ID の移管) に、Elastic IP アドレスの転送先の AWS アカウント ID を入力してください。
7. テキストボックスに「**enable**」と入力して移管を確定します。
8. [送信] を選択してください。
9. 転送を承諾するには、「転送された Elastic IP アドレスを承諾する」を参照してください。転送を無効にするには、「Elastic IP アドレス転送を無効にする」を参照してください。

Elastic IP アドレス転送を無効にする

このセクションではElastic IP 移管を有効にした後に Elastic IP 転送を無効にする方法について説明します。

これらのステップは移管を有効にしたソースアカウントが実行する必要があります。

Elastic IP アドレス移管を無効にするには

1. ソースの AWS アカウントを使用していることを確認してください。
2. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
3. ナビゲーションペインで [Elastic IP] を選択します。
4. Elastic IP のリソースリストで、[Transfer status] (移管ステータス) 列を表示するプロパティが有効になっていることを確認します。
5. [Transfer status] (移管ステータス) が [Pending] (保留中) の Elastic IP アドレスを 1 つ以上選択し、[Action] (アクション)、[Disable transfer] (移管を無効にする) を選択してください。
6. テキストボックスに「**`disable`**」と入力して確認します。
7. [Submit] を選択してください。

転送された Elastic IP アドレスを承諾する

このセクションでは移管された Elastic IP アドレスを承諾する方法について説明します。

Elastic IP アドレスを転送する場合、AWS アカウント の間で 2 段階のハンドシェイクが行われます。ソースアカウントが移管を開始してから 7 日間は転送先アカウントが Elastic IP アドレス移管を受け入れることができます。この 7 日間、ソースアカウントは保留中の移管を (AWS コンソールや AWS CLI コマンドの describe-address-transfers などを使用して) 確認できます。7 日後、移管の有効期限が切れ、Elastic IP アドレスの所有権がソースアカウントに戻ります。

転送を承諾する際に発生する可能性のある例外と、解決する方法は次のとおりです。

- AddressLimitExceeded: 転送先アカウントが Elastic IP アドレスのクォータを超えている場合、ソースアカウントは Elastic IP アドレス移管を有効にできますが、この例外は転送先アカウントが移管を承諾しようとした場合に発生します。デフォルトではすべての AWS アカウントはリージョンあたり 5 つの Elastic IP アドレスに制限されています。制限を引き上げる手順については、「Amazon EC2 ユーザーガイド」の「Elastic IP アドレスのクォータ」を参照してください。
- InvalidTransfer.AddressCustomPtrSet: 転送しようとしている Elastic IP アドレスは、逆引き DNS ルックアップを使用するように設定されます。送信元のアカウントは、Elastic IP アドレスの転送

を有効にできます。ただし、この例外は、転送先アカウントが転送を受け入れようとしたときに発生します。この問題を解決するには転送元アカウントで Elastic IP アドレスの DNS レコードを削除する必要があります。詳細については、「Amazon EC2 Linux」の「Remove a reverse DNS record」を参照してください。

- InvalidTransfer.AddressAssociated: Elastic IP アドレスが ENI や EC2 インスタンスと関連付けられている場合、転送元アカウントはその Elastic IP アドレスに対して移管を有効にできますが、転送元アカウントが移管を受け入れようとするとこの例外が発生します。この問題を解決するにはソースアカウントが Elastic IP アドレスの関連付けを解除する必要があります。詳細については、「Amazon EC2 ユーザーガイド」の「Disassociate an Elastic IP address」を参照してください。

その他の例外についてはサポート にお問い合わせください。

これらのステップは転送先アカウントで実行する必要があります。

Elastic IP アドレスの移管を承諾するには

1. 転送先アカウントを使用していることを確認してください。
2. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
3. ナビゲーションペインで [Elastic IP] を選択します。
4. [Action] (アクション)、[Accept transfer] (移管を許可する) を選択してください。
5. 転送を受け入れると、移管される Elastic IP アドレスに関連付けられたタグは転送されません。承諾する Elastic IP アドレスの [Name] (名前) タグを定義する場合は[Create a tag with a key of 'Name' and a value that you specify] ('Name'のキーと指定した値を使用してタグを作成) を選択してください。
6. 移管する Elastic IP アドレスを入力してください。
7. 複数の移管された Elastic IP アドレスを受け入れる場合は[Add address] (アドレスを追加) を選択して追加の Elastic IP アドレスを入力してください。
8. [Submit] を選択してください。

5. Elastic IP アドレスをリリース

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 での使用に割り当てられている Elastic IP アドレス に対しては、インスタンスに関連付けられていなくても、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。

6. Elastic IP アドレスの復元

Elastic IPアドレスを解放した後で元に戻したくなった場合、復元が可能なこともあります。既に他の AWS アカウントに割り当てられている場合、または復元によって Elastic IP アドレスのクォータを超過してしまう場合は、Elastic IP アドレスを復元することはできません。

Elastic IP アドレスを復元するには、Amazon EC2 API またはコマンドラインツールを使用します。

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
```

コマンドラインの概要

このセクション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 EC2 の使用」を参照してください。

Elastic IP アドレス転送を承諾する

- accept-address-transfer (AWS CLI)
- Approve-EC2AddressTransfer (AWS Tools for Windows PowerShell)

Elastic IP アドレスを割り当てる

- allocate-address (AWS CLI)

- New-EC2Address (AWS Tools for Windows PowerShell)

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

- associate-address (AWS CLI)
- Register-EC2Address (AWS Tools for Windows PowerShell)

Elastic IP アドレス転送の詳細を表示する

- describe-address-transfers (AWS CLI)
- Get-EC2AddressTransfer (AWS Tools for Windows PowerShell)

Elastic IP アドレス転送を無効にする

- disable-address-transfer (AWS CLI)
- Disable-EC2AddressTransfer (AWS Tools for Windows PowerShell)

Elastic IP アドレスの関連付けを解除する

- disassociate-address (AWS CLI)
- Unregister-EC2Address (AWS Tools for Windows PowerShell)

Elastic IP アドレス転送を有効にする

- enable-address-transfer (AWS CLI)
- Enable-EC2AddressTransfer (AWS Tools for Windows PowerShell)

Elastic IP アドレスをリリース

- release-address (AWS CLI)
- Remove-EC2Address (AWS Tools for Windows PowerShell)

Elastic IP アドレスにタグを適用する

- create-tags (AWS CLI)

- New-EC2Tag (AWS Tools for Windows PowerShell)

Elastic IP アドレスの表示

- describe-addresses (AWS CLI)
- Get-EC2Address (AWS Tools for Windows PowerShell)

トランジットゲートウェイを使用して VPC を他の VPC およびネットワークに接続する

中心的なハブとして機能し、VPC、VPN 接続、および Direct Connect 接続間でトラフィックをルーティングするトランジットゲートウェイを使用して、仮想プライベートクラウド (VPC) とオンプレミスネットワークを接続できます。

トランジットゲートウェイを使用する主な利点の一つは、VPC とオンプレミスネットワーク間の接続管理を集約して簡素化できることです。複数の VPN 接続または Direct Connect リンクを設定する代わりに、トランジットゲートウェイを単一の統合ポイントとして使用してネットワークアーキテクチャの全般的な複雑さと運用上のオーバーヘッドを軽減できます。

トランジットゲートウェイの使用料金は、ゲートウェイを介して転送したデータの量に基づきます。トランジットゲートウェイを介して転送されるデータは GB あたりで料金が設定され、トランジットゲートウェイのリソース自体は 1 時間あたりの料金が別に設定されています。具体的な金額は AWS リージョンごとに異なり、また変更される可能性があるため、現行の AWS Transit Gateway 料金ページで最新の情報を確認することが重要です。トランジットゲートウェイの料金モデルを理解しておくことで、AWS ネットワークサービスに関連した継続コストのプランニングと予算編成を改善することができます。その上さらに、効率的な運用性と接続上の利点を兼ね備えたトランジットゲートウェイは、スケーラブルで費用対効果の高いハイブリッドクラウドソリューションの構築を求める組織に実に魅力的な選択肢です。

以下の表ではトランジットゲートウェイの一般的なユースケースについて紹介しています。各ユースケースの詳細については、AWS Transit Gateway ユーザーガイドの「Example transit gateway scenarios」を参照してください。

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。
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。
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、AWSTransit Gatewayを参照してください。

AWS Virtual Private Network を使用して VPC をリモートネットワークに接続する

以下の VPN 接続オプションを使用すると、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。

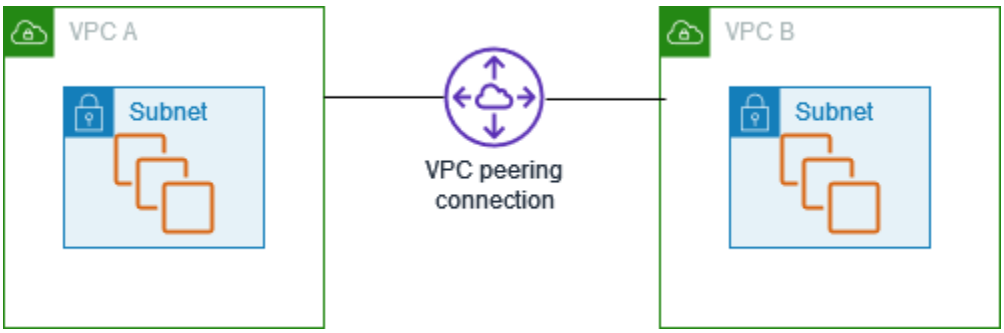
VPN 接続オプション	説明
AWS Site-to-Site VPN	VPC とリモートネットワーク間で、IPsec および VPN 接続を作成できます。Site-to-Site VPN 接続の AWS 側では、仮想プライベートゲートウェイまたはトランジットゲートウェイによって、自動フェイルオーバーのための 2 つの VPN エンドポイント (トンネル) が提供されます。Site-to-Site VPN 接続のリモート側でカスタマーゲートウェイデバイスを設定します。詳細については、「AWS Site-to-Site VPN ユーザーガイド」を参照してください。

VPN 接続オプション	説明
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。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。。AWS Marketplace

また、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、Direct Connectユーザーガイドの「Direct Connect とは」を参照してください。

VPC ピアリングを使用して VPC を接続する

VPC ピアリング接続は、AWS インフラストラクチャ内の 2 つの仮想プライベートクラウド (VPC) 間での安全な直接通信を可能にするネットワーク機能です。このプライベート接続により、ピア接続された VPC 内のリソースは、同じネットワークに含まれているかのように相互に通信できるようになり、パブリックインターネットを経由する必要がなくなります。



VPC ピアリング接続を作成するプロセスでは、既存の VPC インフラストラクチャを使用して接続が確立されるので、ゲートウェイ、AWS Site-to-Site VPN、その他の追加の物理ハードウェアは不要です。この設計により、単一障害点や帯域幅のボトルネックを取り除いています。

VPC ピアリング接続の大きな利点の一つは、異なる AWS アカウント間、または異なる AWS リージョン間で VPC を接続できることです。この柔軟性により、組織はクラウドリソースを、それが同じアカウント内にあるかまたは複数のアカウントや地理的な場所に分散しているかにかかわらず、シームレスに統合することができます。また、接続のプライベート性により、ピア接続された VPC 間のすべてのデータトラフィックは、パブリックインターネットを経由せずに AWS ネットワーク内に留まることができます。

VPC ピアリング接続のユースケースは幅広い範囲に及びます。この機能を活用することで、アプリケーションのさまざまな階層 (ウェブサーバーやデータベースサーバーなど) 間で安全な通信を実現し、複数のチームやビジネスユニット間でのリソース共有を促すことや、オンプレミスネットワークを AWS VPC に接続してハイブリッドクラウドのアーキテクチャを有効にすることもできます。

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。ピア接続された VPC のリソースは、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。ピア接続された VPC 間のトラフィックは、パブリックインターネットを経由しません。

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。

VPC のモニタリング

次のツールを使用して、仮想プライベートクラウド (VPC) 内のトラフィックまたはネットワークアクセスをモニタリングできます。

VPC フローログ

VPC フローログを使用して、ネットワークインターフェイス間で送受信されるトラフィックに関する詳細情報を取得できます。

Amazon CloudWatch Internet Monitor

Internet Monitor を利用すると、AWS でホストされているアプリケーションとエンドユーザーの間で、インターネットの問題がパフォーマンスや可用性にどのように影響しているかを可視化できます。また、アプリケーションの予測されるレイテンシーをどのように改善するかを、ほぼリアルタイムで探索できます。そのために、使用するサービスの切り替えや、さまざまな AWS リージョン を経由してのワークロードに対するトラフィックの再ルーティングをします。詳細については、「Amazon CloudWatch Internet Monitor の使用」を参照してください。

Amazon VPC IP Address Manager (IPAM)

IPAM を使用して、ワークロードの IP アドレスを計画、追跡、およびモニタリングできます。詳細については、「IP Address Manager」を参照してください。

トラフィックのミラーリング

この機能を使用して、Amazon EC2 インスタンスのネットワークインターフェイスからネットワークトラフィックをコピーし、それを帯域外のセキュリティおよびモニタリングアプライアンスに送信して、ディープパケットインスペクションを実行できます。ネットワークとセキュリティの異常を検出し、運用上のインサイトを得て、コンプライアンスとセキュリティのコントロールを実装し、問題をトラブルシューティングできます。詳細については、「トラフィックミラーリング」を参照してください。

Reachability Analyzer

このツールを使用して、VPC 内の 2 つのリソース間のネットワーク到達可能性を分析およびデバッグできます。ソースリソースと宛先リソースを指定した後、Reachability Analyzer は、到達可能なときにそれらの間の仮想パスのホップバイホップの詳細を生成し、到達できないときにブロッキングコンポーネントを識別します。詳細については、「Reachability Analyzer」(到達可能性アナライザー) を参照してください。

Network Access Analyzer

Network Access Analyzer を使用して、リソースへのネットワークアクセスを理解できます。これは、ネットワークセキュリティ体制に対する改善点を特定し、ネットワークが特定のコンプライアンス要件を満たしていることを実証するのに役立ちます。詳細については、「Network Access Analyzer」を参照してください。

CloudTrail ログ

AWS CloudTrail は、次のような Amazon VPC の API コールを記録します。

- 実行された API コール (VPC リソースの作成や変更などのアクション)
- コールの送信元 IP アドレス
- コールを実行したユーザー
- コールが実行された日時

`CreateVpc`、`DeleteVpc` および `CreateDefaultVpc` アクション用に個別のログが作成されます。これらのログには、VPC に対して作成および関連付けられるデフォルトのリソース (デフォルトのインターネットゲートウェイまたはデフォルトのセキュリティグループなど) も含まれます。

詳細については、Amazon EC2 ユーザーガイドの「AWS CloudTrail を使用した Amazon EC2 API コールのログ記録」を参照してください。

VPC フローログを使用した IP トラフィックのログ記録

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、Amazon S3、Amazon Data Firehose に発行できます。ネットワーク トラフィック ログを CloudWatch Logs や S3 などの送信先に送信できるようにする設定された配信パスとアクセス許可は、サブスクリプションと呼ばれます。フローログを作成したら、設定したロググループ、バケット、または配信ストリームのフローログレコード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監視する
- 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。

> **Note**
>
> このセクションでは、VPC のフローログについてのみ説明します。バージョン 6 で導入されたトランジットゲートウェイのフローログの詳細については、「Amazon VPC Transit Gateways ユーザーガイド」の「Transit Gateway フローログを使用したネットワークトラフィックのログ記録」を参照してください。

内容

- フローログの基礎
- フローログレコード
- フローログレコードの例
- フローログの制限事項
- 料金
- フローログの使用
- CloudWatch Logs へのフローログの発行
- フローログを Amazon S3 に発行する
- Amazon Data Firehose へのフローログの発行
- Amazon Athena を使用したフローログのクエリ
- VPC フローログの Identity and Access Management
- VPC フローログトラブルシューティング

フローログの基礎

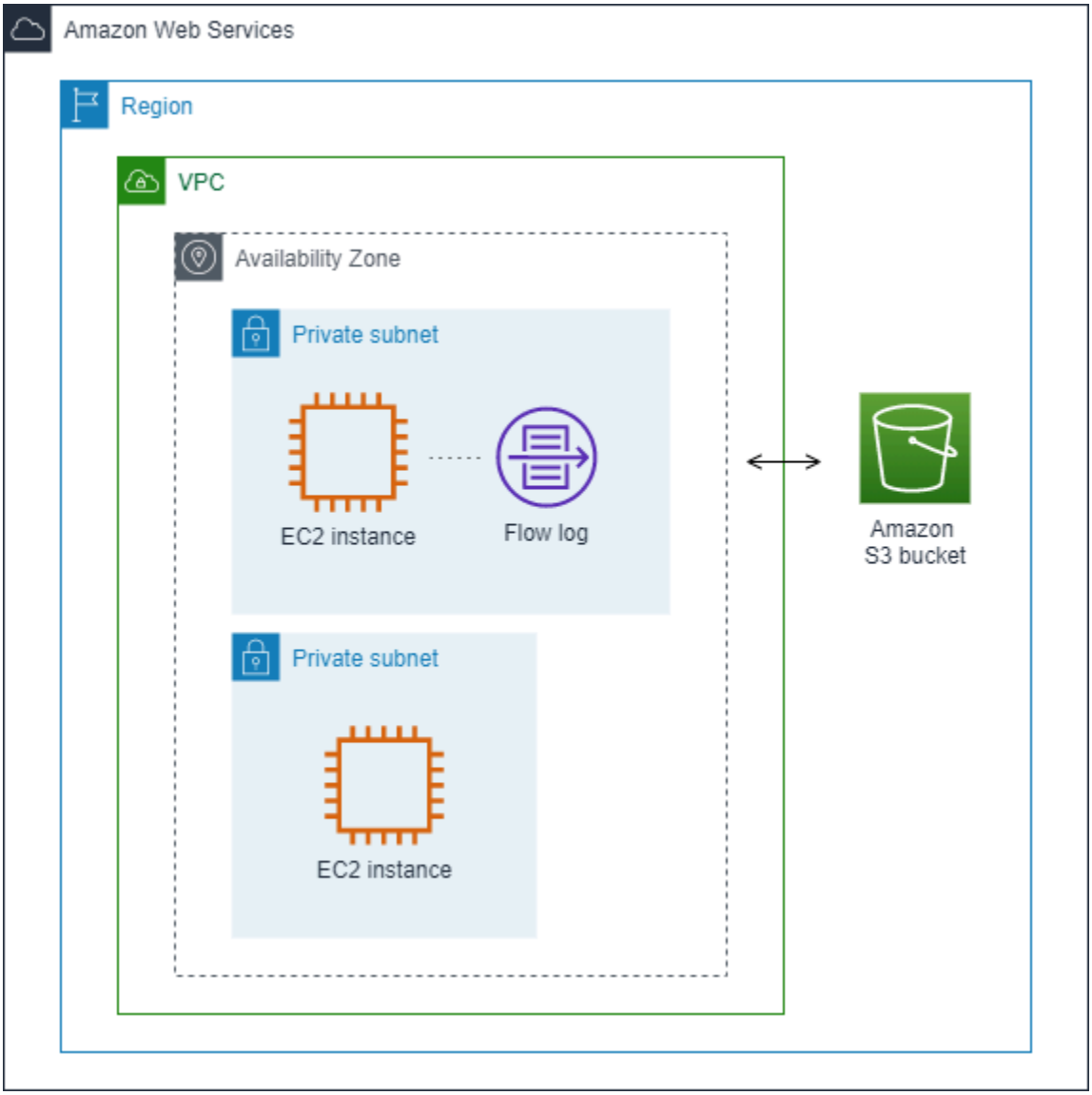
VPC、サブネット、またはネットワークインターフェイスのフローログを作成できます。サブネットまたは VPC のフローログを作成する場合、そのサブネットまたは VPC 内の各ネットワークインターフェイスが監視されます。

監視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、「フローログレコード」を参照してください

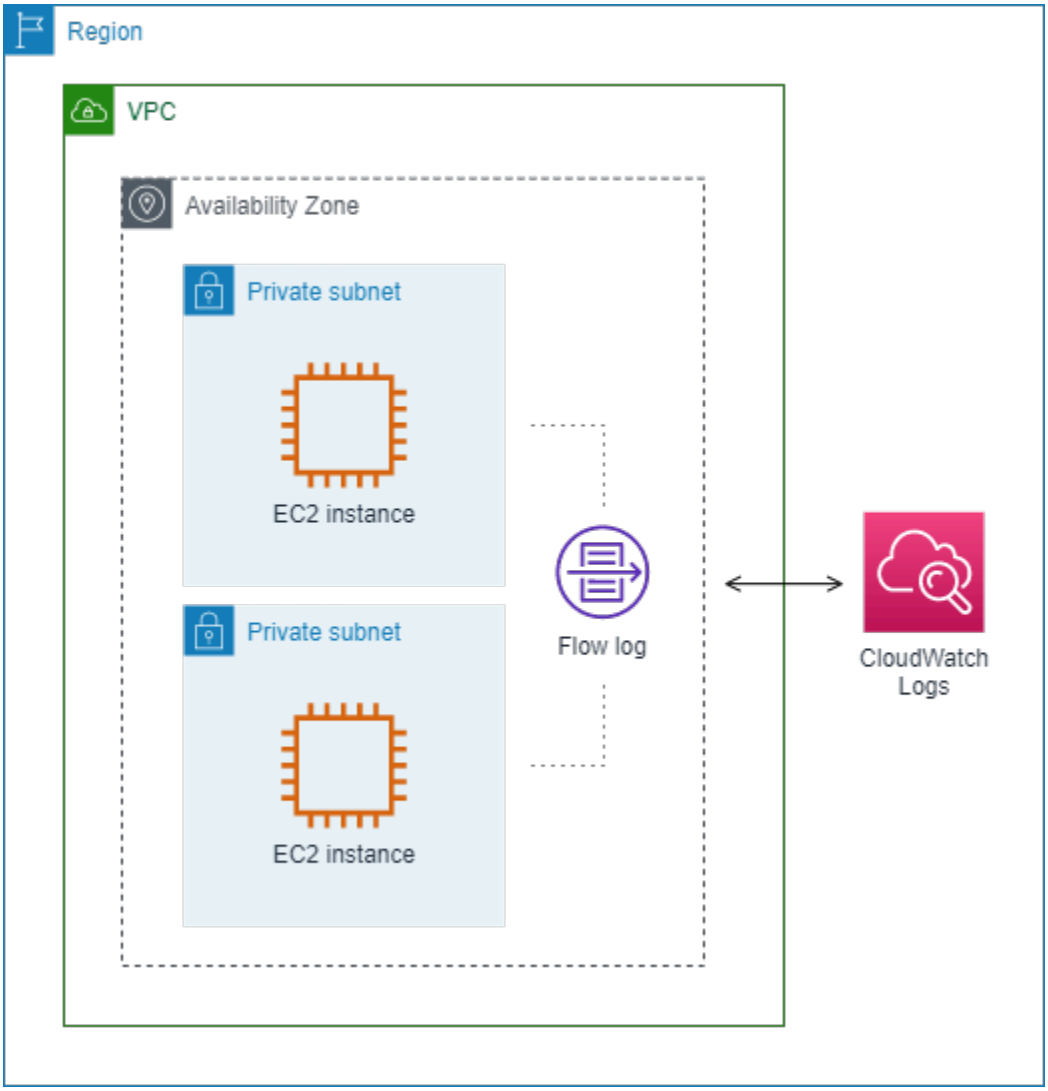
フローログを作成するには、以下の内容を指定します。

- 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、プライベートサブネットの EC2 インスタンス の 1 つ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を作成します。



次の例では、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 に発行します。フローログは、サブネット内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。



フローログを作成後、データ収集と選択された送信先へのデータ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、「2. フローログの作成」を参照してください。

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のログストリーム (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。ログストリームまたはファイルオブジェクトは、インターフェイスでネットワークトラフィックが発生するとすぐに作成されます。

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の作成や発行は行われなくなります。フローログを削除しても、既存のフローログデータは削除されません。フローログの削除が完了したら、送信先から直接フローログデータを削除できます。詳細については、「4. フローログの削除」を参照してください。

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
- デフォルトの形式

- カスタム形式
- 使用可能なフィールド

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。

Amazon S3 にフローログデータを公開する場合、フィールドのデータ型はフローログ形式によって異なります。形式がプレーンテキストの場合、すべてのフィールドは STRING 形式です。形式が Parquet の場合は、フィールドのデータ型の表を参照してください。

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。 Parquet データ型: INT_32	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。 Parquet データ型: STRING	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。リージョン NAT ゲートウェイに関連付けられたフローの「-」記号を返します。 Parquet データ型: STRING	2
srcaddr	受信トラフィックの場合、これはトラフィックの送信元の IP アドレスです。送信トラフィックの場合、これはトラフィックを送信するネットワークインターフェイスのプライベート IPv4 アドレスまたは IPv6 アドレスです。リージョン NAT ゲートウェイからの送信	2

フィールド	説明	バージョン
	トラフィックの場合、これは pkt-srcaddr と同じパケットレベルの送信元 IP アドレスです。pkt-srcaddr も参照してください。 Parquet データ型: STRING	
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。リージョン NAT ゲートウェイへの受信トラフィックの場合、これは pkt-dstaddr と同じパケットレベルの送信先 IP アドレスです。pkt-dstaddr も参照してください。 Parquet データ型: STRING	2
srcport	トラフィックの送信元ポート。 Parquet データ型: INT_32	2
dstport	トラフィックの送信先ポート。 Parquet データ型: INT_32	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。 Parquet データ型: INT_32	2
packets	フロー中に転送されたパケットの数。 Parquet データ型: INT_64	2
bytes	フロー中に転送されたバイト数。 Parquet データ型: INT_64	2

フィールド	説明	バージョン
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ース上で受信される最大 60 秒前、またはパケットが送信される最大 60 秒後である可能性があります。 Parquet データ型: INT_64	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。 Parquet データ型: INT_64	2
action	トラフィックに関連付けられたアクション: • ACCEPT – トラフィックが承認されました。 • REJECT – トラフィックが拒否されました。例えば、トラフィックがセキュリティグループまたはネットワーク ACL により許可されなかった、接続終了後にパケットが到着したなどです。 Parquet データ型: STRING	2

フィールド	説明	バージョン
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。 集計間隔中に一部のフローログレコードがスキップされることがあります (使用可能なフィールド の log-status を参照)。これは、内部の AWS キャパシティの制限または内部エラーが原因で発生する場合があります。VPC フローログの請求金額を表示するために AWS Cost Explorer を使用しており、一部のフローログがフローログの集計間隔中にスキップされた場合は、AWS Cost Explorer で報告されるフローログの数が Amazon VPC により発行されたフローログの数よりも多くなります。 Parquet データ型: STRING	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。 Parquet データ型: STRING	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。リージョン NAT ゲートウェイに関連付けられたフローの「-」記号を返します。 Parquet データ型: STRING	3

フィールド	説明	バージョン
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。 Parquet データ型: STRING	3

フィールド	説明	バージョン
tcp-flags	次の TCP フラグのビットマスク値: • FIN — 1 • SYN — 2 • RST — 4 • SYN-ACK — 18 サポートされているフラグが記録されていない場合、TCP フラグ値は 0 です。例えば、tcp-flags は ACK または PSH フラグのロギングをサポートしていないため、これらのサポートされていないフラグを持つトラフィックのレコードは TCP フラグの値が 0 になります。ただし、サポートされていないフラグにサポートされているフラグが付いている場合は、サポートされているフラグの値を報告します。たとえば、ACK が SYN-ACK の一部であれば、18 と報告します。また、SYN+ECE のようなレコードがある場合、SYN はサポートされているフラグで ECE はサポートされていないため、TCP フラグの値は 2 になります。何らかの理由でフラグの組み合わせが無効で値を計算できない場合、値は「-」になります。フラグが送信されていない場合、TCP フラグの値は 0 です。 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」を参照してください。 TCP フラグの一般的な情報 (FIN、SYN、ACK などのフラグの意味など) については、Wikipedia の「TCP セグメント構造」を参照してください。 Parquet データ型: INT_32	3

フィールド	説明	バージョン
type	トラフィックの種類。指定できる値は IPv4 | IPv6 | EFA です。詳細については、Elastic Fabric Adapterを参照してください。 Parquet データ型: STRING	3
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、トラフィックは NAT ゲートウェイのネットワークインターフェイス を通過する場合があります。別のケースとして、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があります (VPC 内の通信の場合)。 Parquet データ型: STRING	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、トラフィックは NAT ゲートウェイのネットワークインターフェイス を通過する場合があります。別のケースとして、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があります (VPC 内の通信の場合)。 Parquet データ型: STRING	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。 Parquet データ型: STRING	4

フィールド	説明	バージョン
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。 Parquet データ型: STRING	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。 Parquet データ型: STRING	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。 Parquet データ型: STRING	4

フィールド	説明	バージョン
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)。ソース IP アドレスが重複範囲内にある場合、pkt-src-aws-service には AWS サービスコードのいずれか 1 つだけが表示されます。指定できる値は、AMAZON | AMAZON_APPFLOW | AMAZON_CONNECT | API_GATEWAY | AURORA_DSQL | CHIME_MEETINGS | CHIME_VOICECONNECTOR | CLOUD9 | CLOUDFRONT | CLOUDFRONT_ORIGIN_FACING | CODEBUILD | DYNAMODB | EBS | EC2 | EC2_INSTANCE_CONNECT | GLOBALACCELERATOR | IVS_LOW_LATENCY | IVS_REALTIME | KINESIS_VIDEO_STREAMS | MEDIA_PACKAGE_V2 | ROUTE53 | ROUTE53_HEALTHCHECKS | ROUTE53_HEALTHCHECKS_PUBLISHING | ROUTE53_RESOLVER | S3 | WORKSPACES_GATEWAYS です。 Parquet データ型: STRING	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。 Parquet データ型: STRING	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。 Parquet データ型: STRING	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 — AWS マネージドネットワークインターフェイスや Outpost ローカルゲートウェイなど、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 ピアリング接続経由 • 5 — リージョン間 VPC ピアリング接続経由 • 6 — ローカルゾーンまたは Wavelength ゾーン経由 • 7 — ゲートウェイ VPC エンドポイント経由 • 8 — インターネットゲートウェイ経由 Parquet データ型: INT_32	5
ecs-cluster-arn	AWSトラフィックが実行中の ECS タスクからのものである場合の ECS クラスターの リソースネーム (ARN)。このフィールドをサブスクリプションに含めるには、ecs:ListClusters を呼び出すアクセス許可が必要です。 Parquet データ型: STRING	7
ecs-cluster-name	トラフィックが実行中の ECS タスクからのものである場合の ECS クラスターの名前。このフィールドをサブスクリプションに含めるには、ecs:ListClusters を呼び出すアクセス許可が必要です。 Parquet データ型: STRING	7

フィールド	説明	バージョン
ecs-container-instance-arn	トラフィックが EC2 インスタンスで実行中の ECS タスクからのものである場合の ECS コンテナインスタンスの ARN。キャパシティープロバイダーが AWS Fargate の場合、このフィールドは「-」になります。このフィールドをサブスクリプションに含めるには、ecs:ListClusters と ecs:ListContainerInstances を呼び出すアクセス許可が必要です。 Parquet データ型: STRING	7
ecs-container-instance-id	トラフィックが EC2 インスタンスで実行中の ECS タスクからのものである場合の ECS コンテナインスタンスの ID。キャパシティープロバイダーが AWS Fargate の場合、このフィールドは「-」になります。このフィールドをサブスクリプションに含めるには、ecs:ListClusters と ecs:ListContainerInstances を呼び出すアクセス許可が必要です。 Parquet データ型: STRING	7
ecs-container-id	トラフィックが実行中の ECS タスクからのものである場合のコンテナの Docker ランタイム ID。ECS タスクに 1 つまたは複数のコンテナがある場合、これは最初のコンテナの Docker ランタイム ID になります。このフィールドをサブスクリプションに含めるには、ecs:ListClusters を呼び出すアクセス許可が必要です。 Parquet データ型: STRING	7
ecs-second-container-id	トラフィックが実行中の ECS タスクからのものである場合のコンテナの Docker ランタイム ID。ECS タスクに 1 つ以上のコンテナがある場合、これは 2 番目のコンテナの Docker ランタイム ID になります。このフィールドをサブスクリプションに含めるには、ecs:ListClusters を呼び出すアクセス許可が必要です。 Parquet データ型: STRING	7

フィールド	説明	バージョン
ecs-service-name	トラフィックが実行中の ECS タスクからのもので、ECS タスクが ECS サービスによって開始される場合の ECS サービスの名前。ECS サービスによって ECS タスクが開始されない場合、このフィールドは「-」になります。このフィールドをサブスクリプションに含めるには、ecs:ListClusters と ecs:ListServices を呼び出すアクセス許可が必要です。 Parquet データ型: STRING	7
ecs-task-definition-arn	トラフィックが実行中の ECS タスクからのものである場合の ECS タスク定義の ARN。このフィールドをサブスクリプションに含めるには、ecs:ListClusters と ecs:ListTaskDefinitions を呼び出すアクセス許可が必要です。 Parquet データ型: STRING	7
ecs-task-arn	トラフィックが実行中の ECS タスクからのものである場合の ECS タスクの ARN。このフィールドをサブスクリプションに含めるには、ecs:ListClusters と ecs:ListTasks を呼び出すアクセス許可が必要です。 Parquet データ型: STRING	7
ecs-task-id	トラフィックが実行中の ECS タスクからのものである場合の ECS タスクの ID。このフィールドをサブスクリプションに含めるには、ecs:ListClusters と ecs:ListTasks を呼び出すアクセス許可が必要です。 Parquet データ型: STRING	7

フィールド	説明	バージョン
reject-reason	トラフィックが拒否された理由。可能な値: BPA、EC。他の拒否理由の場合は「-」を返します。 • BPA — VPC ブロックパブリックアクセス (BPA) についての詳細は、「VPC とサブネットへのパブリックアクセスをブロックする」を参照してください。 • EC — フローは暗号化コントロールのため VPC エンドポイントによって拒否されます。VPC 暗号化コントロールの詳細については、「転送時の VPC 暗号化を強制する」を参照してください。 Parquet データ型: STRING	8
resource-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を含むリージョン NAT ゲートウェイの ID。リージョン NAT ゲートウェイに関連付けられていないトラフィックフローの「-」記号を返します。リージョン NAT ゲートウェイの詳細については、「自動マルチ AZ 拡張用のリージョン NAT ゲートウェイ」を参照してください。 Parquet データ型: STRING	9

フィールド	説明	バージョン

encryption-status	フローの暗号化ステータス。VPC 暗号化コントロールの詳細については、「転送時の VPC 暗号化を強制する」を参照してください。指定できる値は以下のとおりです。
	• 0 — 暗号化されていません。
	• 1 — Nitro 暗号化。Nitro System ハードウェアによって暗号化されています。
	• 2 — アプリケーション暗号化。アプリケーション暗号化と見なされるのは、以下のみです。
	  • TCP ポート 443 でのインターフェイスエンドポイントから AWS サービスへのフロー*
	  • ゲートウェイエンドポイントに対する TCP ポート 443 のフロー*
	  • VPC エンドポイントを介して暗号化された Redshift クラスターへのフロー**
	• 3 — Nitro 暗号化およびアプリケーション暗号化。
	
	VPC 暗号化コントロールが有効になっていない場合、または FlowLog がステータスを取得できない場合、値は「-」です。
	
	Note
	* インターフェイスエンドポイントとゲートウェイエンドポイントの場合、AWS はパケットデータを調べて暗号化ステータスを判断するのではなく、暗号化ステータスを引き受けるために使用されるポートに依存します。
	** 指定された AWS マネージドエンドポイントの場合、AWS はサービス設定の TLS の要件に基づいて暗号化ステータスを決定します。
	
	Parquet データ型: INT_32
		10

フィールド	説明	バージョン
instance-tag	TagFieldSpecifications に含まれる 1 つ目のインスタンスタグのタグ値。タグ値は、UTF-8 パーセントエンコードですべての特殊文字を翻訳して表示されます。このフィールドをサブスクリプションに含めるには、ec2:DescribeTags および iam:CreateServiceLinkedRole を呼び出すアクセス許可が必要です。フローログ Amazon EC2 タグのサービスリンクロールの詳細については、「VPC フローログのサービスリンクロールの使用」を参照してください。 Parquet データ型: STRING	11
instance-tag-2	TagFieldSpecifications に含まれる 2 つ目のインスタンスタグのタグ値。タグ値は、UTF-8 パーセントエンコードですべての特殊文字を翻訳して表示されます。このフィールドをサブスクリプションに含めるには、ec2:DescribeTags および iam:CreateServiceLinkedRole を呼び出すアクセス許可が必要です。フローログ Amazon EC2 タグのサービスリンクロールの詳細については、「VPC フローログのサービスリンクロールの使用」を参照してください。 Parquet データ型: STRING	11
interface-tag	TagFieldSpecifications に含まれる 1 つ目のネットワークインターフェイスタグのタグ値。タグ値は、UTF-8 パーセントエンコードですべての特殊文字を翻訳して表示されます。このフィールドをサブスクリプションに含めるには、ec2:DescribeTags および iam:CreateServiceLinkedRole を呼び出すアクセス許可が必要です。フローログ Amazon EC2 タグのサービスリンクロールの詳細については、「VPC フローログのサービスリンクロールの使用」を参照してください。 Parquet データ型: STRING	11

フィールド	説明	バージョン
interface-tag-2	TagFieldSpecifications に含まれる 2 つ目のネットワークインターフェイスタグのタグ値。タグ値は、UTF-8 パーセントエンコードですべての特殊文字を翻訳して表示されます。このフィールドをサブスクリプションに含めるには、ec2:DescribeTags および iam:CreateServiceLinkedRole を呼び出すアクセス許可が必要です。フローログ Amazon EC2 タグのサービスリンクロールの詳細については、「VPC フローログのサービスリンクロールの使用」を参照してください。 Parquet データ型: STRING	11
asg-tag	TagFieldSpecifications に含まれる 1 つ目の auto-scaling-group タグのタグ値。タグ値は、UTF-8 パーセントエンコードですべての特殊文字を翻訳して表示されます。このフィールドをサブスクリプションに含めるには、autoscaling:DescribeTags および iam:CreateServiceLinkedRole を呼び出すアクセス許可が必要です。Auto Scaling グループのタグ値をリアルタイムで適切に追跡および更新するには、アカウントに 1 つ以上の有効な CloudTrail 証跡が必要です。フローログ Amazon EC2 タグのサービスリンクロールの詳細については、「VPC フローログのサービスリンクロールの使用」を参照してください。 Parquet データ型: STRING	11

フィールド	説明	バージョン
asg-tag-2	TagFieldSpecifications に含まれる 2 つ目の auto-scaling-group タグのタグ値。タグ値は、UTF-8 パーセントエンコードですべての特殊文字を翻訳して表示されます。このフィールドをサブスクリプションに含めるには、autoscaling:DescribeTags および iam:CreateServiceLinkedRole を呼び出すアクセス許可が必要です。Auto Scaling グループのタグ値をリアルタイムで適切に追跡および更新するには、アカウントに 1 つ以上の有効な CloudTrail 証跡が必要です。フローログ Amazon EC2 タグのサービスリンクロールの詳細については、「VPC フローログのサービスリンクロールの使用」を参照してください。 Parquet データ型: STRING	11
interface-type	フローをキャプチャするローカルネットワークインターフェイスのタイプ。サポートされている値は次のとおりです: `nat_gateway` | `network_load_balancer` | `regional_nat_gateway` | `transit_gateway` | `vpc_endpoint` 。ネットワークインターフェイスに関連サービスがない場合、またはサービスがサポートされていない場合、値は「-」です。 Parquet データ型: STRING	11
next-hop-interface-id	ネクストホップネットワークインターフェイスの ID。出力トラフィックの場合、これはトラフィックを受信するインターフェイスです。入力トラフィックの場合、これはトラフィックを送信するインターフェイスです。 Parquet データ型: STRING	11
next-hop-subnet-id	ネクストホップネットワークインターフェイスを含むサブネットの ID。 Parquet データ型: STRING	11

フィールド	説明	バージョン
next-hop-az-id	ネクストホップネットワークインターフェイスを含むアベイラビリティーゾーンの ID。 Parquet データ型: STRING	11
next-hop-vpc-id	ネクストホップネットワークインターフェイスを含む VPC の ID。 Parquet データ型: STRING	11
next-hop-interface-type	ネクストホップネットワークインターフェイスのタイプ。可能な値の一覧については、interface-type フィールドを参照してください。 Parquet データ型: STRING	11

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」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」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
- データなしおよびスキップされたレコード
- セキュリティグループとネットワーク ACL ルール
- IPv6 トラフィック
- TCP フラグシーケンス
- ゾーン NAT ゲートウェイ経由のトラフィック
- リージョン NAT ゲートウェイ経由のトラフィック
- トランジットゲートウェイ経由のトラフィック
- サービス名、トラフィックパス、およびフロー方向

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、IP アドレス 172.31.16.139 からプライベート IP アドレスを使用したネットワークインターフェイスへの SSH トラフィック (宛先ポート 22、TCP プロトコル) が 172.31.16.21 で、アカウント 123456789010 の ID eni-1235b8ca123456789 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
```

VPC Flog Logs が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、内部容量を超えるため、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
```

> **Note**
>
> 集計間隔中に一部のフローログレコードがスキップされることがあります (使用可能なフィールド の log-status を参照)。これは、内部の AWS キャパシティの制限または内部エラーが原因で発生する場合があります。VPC フローログの請求金額を表示するために AWS Cost Explorer を使用しており、一部のフローログがフローログの集計間隔中にスキップされた場合は、AWS Cost Explorer で報告されるフローログの数が Amazon VPC により発行されたフローログの数よりも多くなります。

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

このセクションでは、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が含まれています。

```
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
```

このセクションの例の tcp-flags フィールドは、フローログの最後から 2 番目の値で表されます。TCP フラッグは、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。

> **Note**
>
> tcp-flags オプションおよび各 TCP フラグの説明についての詳細は、「使用可能なフィールド」を参照してください。

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。サーバーは、ポート 43418 での接続に対してクライアントに FIN フラグ (1) を送信しています。クライアントは、サーバーのポート 43418 に FIN で応答しています。

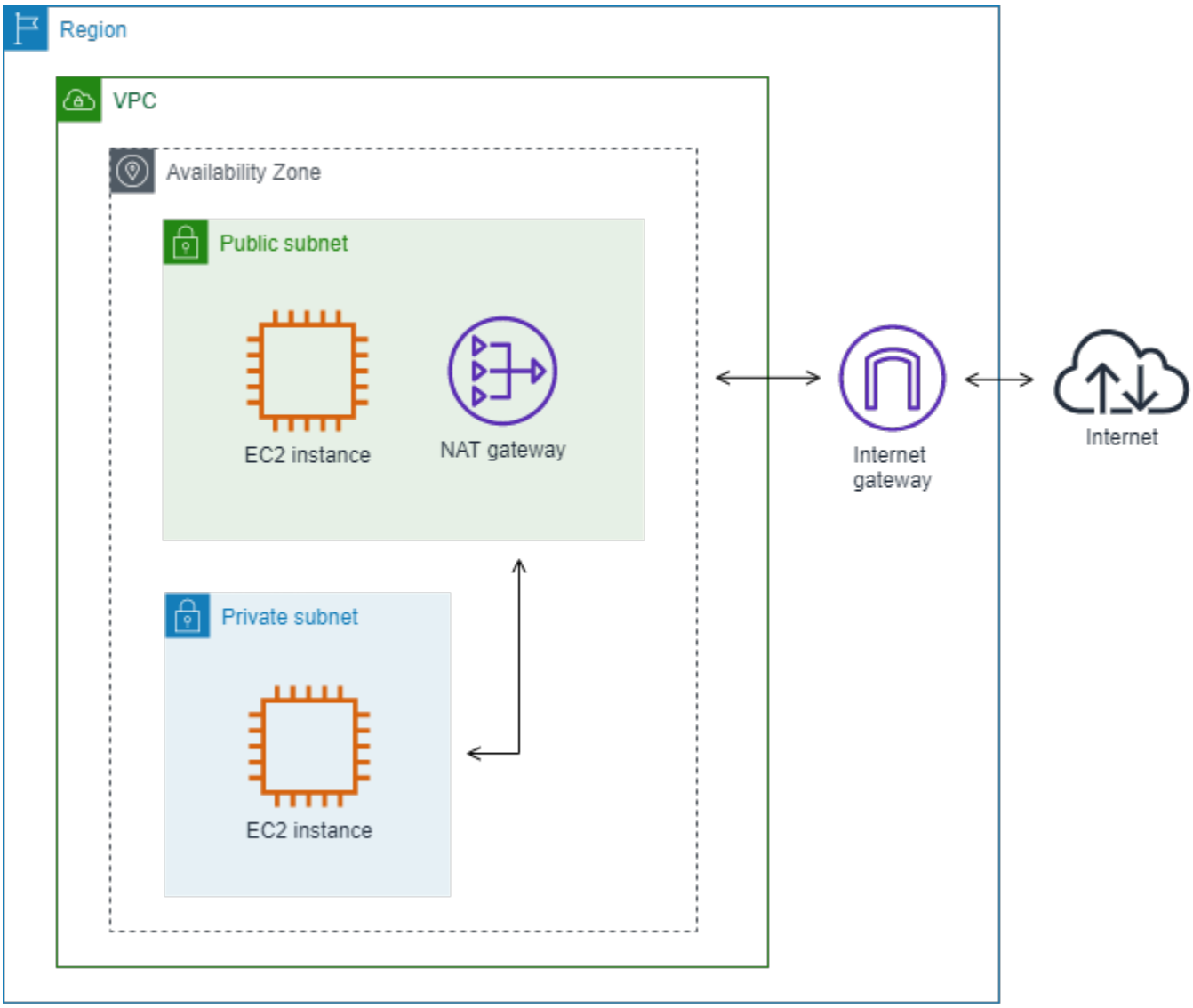
```
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456
 eni-1235b8ca123456789 123456789010 IPv4 10.0.0.62 52.213.180.42 5001 43418 10.0.0.62
 52.213.180.42 6 63388 1219 1566848933 1566849113 ACCEPT 1 OK
3 vpc-abcdefab012345678 subnet-aaaaaaaa012345678 i-01234567890123456
 eni-1235b8ca123456789 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、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
```

ゾーン 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ゾーン 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。



ゾーン 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) からゾーン 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が表示されます。ゾーン NAT ゲートウェイネットワークインターフェイスは、リクエスタマネージド型の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ゾーン 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、ゾーン 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、ゾーン 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、ゾーン 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、ゾーン 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
```

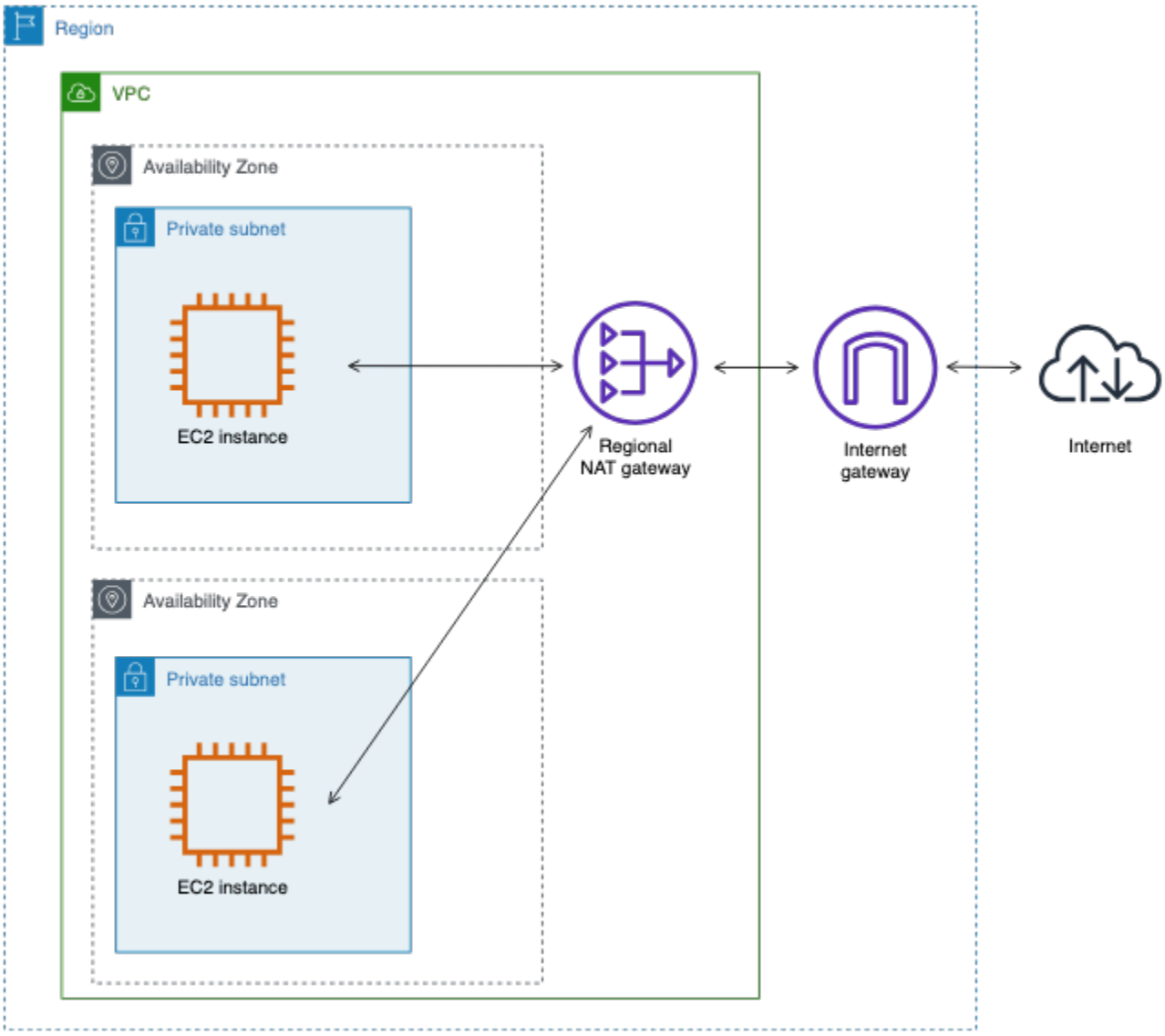
上記と同じフィールドセットを使用して別のカスタムフローログを作成できます。プライベートサブネット内のインスタンスのネットワークインターフェイスのフローログを作成します。この場合、instance-id フィールドはネットワークインターフェイスに関連付けられているインスタンスの ID を返します。dstaddr および pkt-dstaddr フィールドと srcaddr および pkt-srcaddr フィールドの間に違いはありません。ゾーン NAT ゲートウェイのネットワークインターフェイスとは異なり、このネットワークインターフェイスはトラフィックの中間ネットワークではありません。

```
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
```

リージョン NAT ゲートウェイ経由のトラフィック

リージョン NAT ゲートウェイは、異なるアベイラビリティーゾーンにまたがる複数のサブネットに接続できます。この例では、2 つの異なるアベイラビリティーゾーンのプライベートサブネット内の 2 つのインスタンスが、同じリージョン NAT ゲートウェイを介してインターネットにアクセスし

ます。次のフローログは、いずれかのインスタンスからリージョン NAT ゲートウェイを介したインターネットへのトラフィックを示しています。



リージョン NAT ゲートウェイ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。

```
resource-id instance-id interface-id subnet-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) からリージョン NAT ゲートウェイを経由してインターネット上のホスト (203.0.113.5) へのトラフィックのフローが表示されます。instance-id、interface-id、subnet-id は、リージョン NAT ゲートウェイには適用されません。したがって、フローログレコードでは、これらのフィールドに「-」記号が表示されます。代わりに、resource-

id フィールドにはリージョン NAT ゲートウェイの ID が表示されます。dstaddr フィールドと 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宛先 IP アドレスが表示されます。

```
nat-1234567890abcdef - - - 10.0.1.5 203.0.113.5 10.0.1.5 203.0.113.5
```

次の 2 行は、リージョン NAT ゲートウェイ (パブリック IP アドレス 107.22.182.139) からインターネット上のターゲットホストへのトラフィックを示しています。また、ホストからリージョン NAT ゲートウェイへのレスポンストラフィックも表示されます。

```
nat-1234567890abcdef - - - 107.22.182.139 203.0.113.5 107.22.182.139 203.0.113.5
nat-1234567890abcdef - - - 203.0.113.5 107.22.182.139 203.0.113.5 107.22.182.139
```

次の行は、リージョン NAT ゲートウェイから送信元インスタンスへのレスポンス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ます。

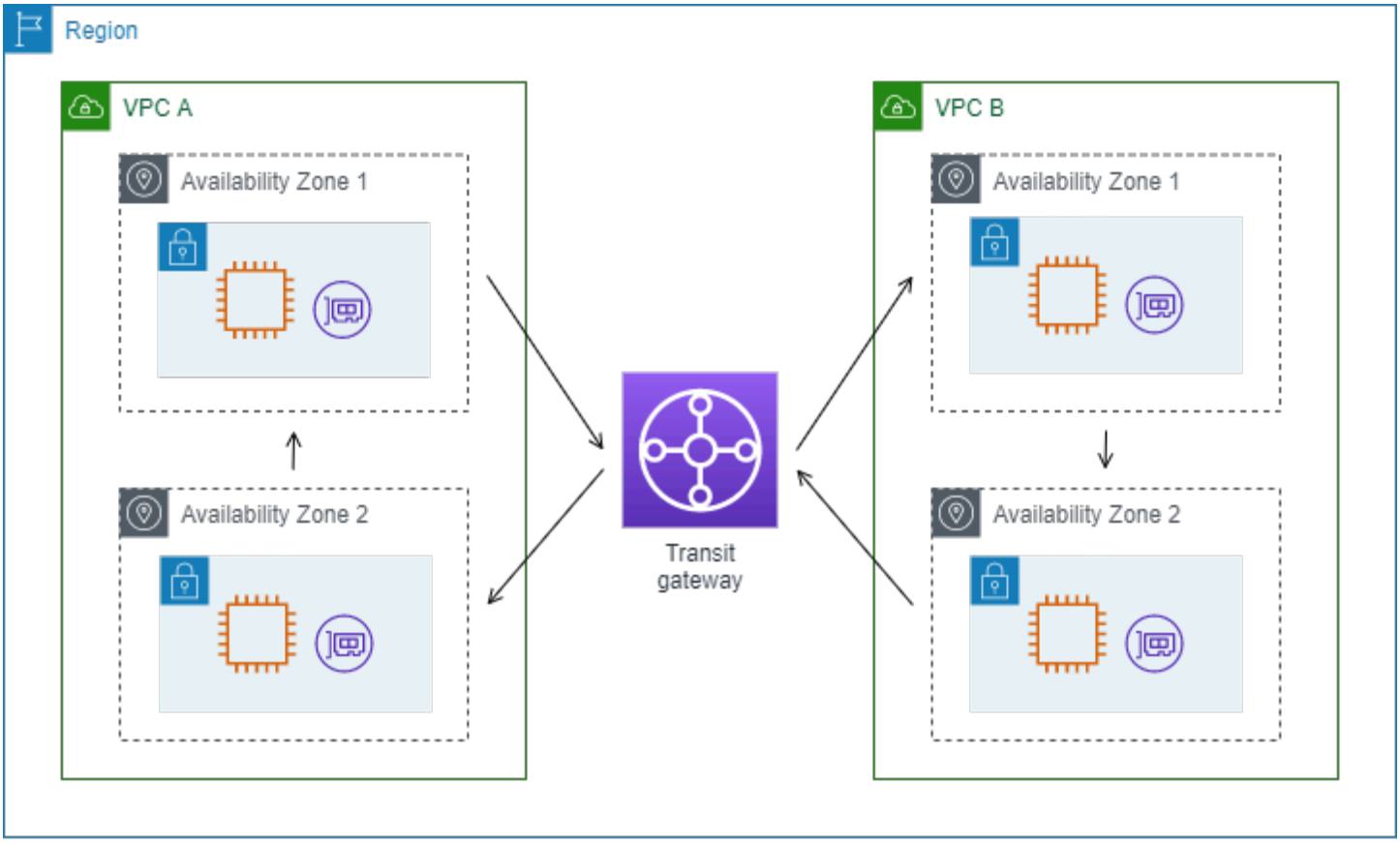
```
nat-1234567890abcdef - - - 203.0.113.5 10.0.1.5 203.0.113.5 10.0.1.5
```

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付けられているインスタンスの ID を返し、resource-id は「-」になります。dstaddr および pkt-dstaddr フィールドと srcaddr および pkt-srcaddr フィールドの間に違いはありません。

```
- i-01234567890123456 eni-1111aaaa2222bbbb3 subnet-aaaaaaaa012345678 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 subnet-aaaaaaaa012345678 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
```

トランジット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 Transit Gateway 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。トラフィックは、1 つの Elastic Network Interface ID を使用して VPC B のサーバーに到着し (この例では、ID が eni-11111111111111111 であるとします)、別のものを使用して VPC B を離れます (例: eni-22222222222222222)。



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 Transit Gateway 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、Transit Gateway 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-1111111111111111 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 Transit Gateway 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、Transit Gateway 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。

- フローログの作成後は、選択したネットワークインターフェイス、サブネット、VPC のアクティブなトラフィックがあるまで、フローログデータは表示されません。
- 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。
- フローログを作成後に、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、動作はフローの方向に応じて異なります。エグレスフローの場合、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。

  ログレコードがネットワークインターフェイスへの入力フロー用である場合、ネットワークインターフェイスのプライマリプライベート IP は srcaddr フィールドに表示されません。
- 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。

- `pkt-srcaddr` と `pkt-dstaddr` のフィールドについては、中間レイヤーでクライアント IP アドレスの保存が有効になっている場合、このフィールドに表示されるのが中間レイヤーの IP アドレスではなく、保存されたクライアント IP になることがあります。
- `traffic-path` フィールドについては、値は同じ VPC 内のリソースを経由するフローと、Outpost ローカルゲートウェイを経由するフローで同じです。
- 集計間隔中に一部のフローログレコードがスキップされることがあります (使用可能なフィールド の log-status を参照)。これは、内部の AWS キャパシティの制限または内部エラーが原因で発生する場合があります。VPC フローログの請求金額を表示するために AWS Cost Explorer を使用しており、一部のフローログがフローログの集計間隔中にスキップされた場合は、AWS Cost Explorer で報告されるフローログの数が Amazon VPC により発行されたフローログの数よりも多くなります。
- VPC ブロックパブリックアクセス (BPA) を使用している場合:
  - VPC BPA のフローログには、スキップされたレコードは含まれません。
  - VPC BPA のフローログには、フローログに `bytes` フィールドを含めた場合であっても、`bytes` は含まれません。
- VPC フローログは、アカウントごとにリソースごとに最大 250 件のサブスクリプションをサポートします。この制限に達したリソースで追加のサブスクリプションを作成するには、まず既存のサブスクリプションを削除する必要があります。

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。
- トラフィックミラーリングされたソーストラフィック。トラフィックミラーリングされたターゲットトラフィックのみが表示されます。
- 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。
- アドレス解決プロトコル (ARP) トラフィック。
- 有効期間が短いリージョン NAT ゲートウェイ上のトラフィック。作成後数分で削除されます。

バージョン 7 で利用可能な ECS フィールドに固有の制約事項:

- 基盤となる ECS タスクがフローログサブスクリプションの所有者によって所有されていない場合、ECS フィールドは計算されません。

  例えば、サブネット (SubnetA) を別のアカウント (AccountB) と共有し、SubnetA のフローログサブスクリプションを作成するとします。AccountB が共有サブネットで ECS タスクを起動すると、サブスクリプションはそれらのタスクからトラフィックログを受け取ります。ただし、セキュリティ上の懸念により、これらのログの ECS フィールドは計算されません。
- VPC / サブネットリソースレベルで ECS フィールドを使用してフローログサブスクリプションを作成すると、ECS 以外のネットワークインターフェイス用に生成されたトラフィックもサブスクリプションに対して配信されます。ECS フィールドの値は、ECS 以外の IP トラフィックでは「-」になります。例えば、サブネット (subnet-000000) があり、ECS フィールド (fl-00000000) を使用してこのサブネットのフローログサブスクリプションを作成する場合を考えてみます。subnet-000000 で、インターネットに接続され、IP トラフィックをアクティブに生成している EC2 インスタンス (i-0000000) を起動します。また、同じサブネットで実行中の ECS タスク (ECS-Task-1) を起動します。i-0000000 と ECS-Task-1 の両方が IP トラフィックを生成しているため、フローログサブスクリプション fl-00000000 は両方のエンティティのトラフィックログを配信します。ただし、logFormat に含めた ECS フィールドの実際の ECS メタデータを取得するのは ECS-Task-1 だけです。i-0000000 に関連するトラフィックの場合、これらのフィールドの値は「-」になります。
- ecs-container-id と ecs-second-container-id の順序は、VPC フローログサービスが ECS イベントストリームからそれらを受信したときに基づきます。ECS コンソールまたは DescribeTask API コールで表示されるのと同じ順序であるとは限りません。タスクがまだ実行しているときにコンテナが STOPPED ステータスになると、このコンテナが引き続きログに表示されることがあります。
- ECS メタデータと IP トラフィックログは、2 つの異なるソースからのものです。アップストリームの依存関係から必要な情報をすべて取得すると、ECS トラフィックの計算がすぐに開始されます。

新しいタスクを開始した後、ECS フィールドの計算は、基盤となるネットワークインターフェイスの IP トラフィックの受信、およびタスクが現在実行中であることを示す ECS タスクのメタデータを含む ECS イベントの受信という 2 つの条件が両方とも満たされたときに開始されます。

タスクを停止した後、ECS フィールドの計算は、基盤となるネットワークインターフェイスの IP トラフィックが受信されなくなる (または 1 日以上遅延した IP トラフィックを受信する)、およびタスクがもう実行中でなくなったことを示す ECS タスクのメタデータを含む ECS イベントが受信されるという 2 つの条件が両方とも満たされたときに停止します。

- サポートされるのは、awsvpc ネットワークモードで起動した ECS タスクのみです。

encryption-status フィールドに固有の制限:

- 一部のネットワークアプライアンスが暗号化ステータスを報告する制限があるため、一部のフローでは暗号化ステータスが「-」(使用不可) になる場合があります。ユーザーは分析でこれらのフローを無視できます。
- モニターモードで暗号化として表示されても、フローが強制モードで許可されるわけではありません。逆も同様です。
  - フローがモニターモードで暗号化されている場合、強制モードでは準拠していない可能性があります。
    - フローに AWS サービスによって作成された ENI が含まれる場合、サービスは暗号化コントロールをサポートする必要があります。
    - フローが VPC ピアリングを通過する場合、ピアリングされた VPC は暗号化コントロールを強制しない可能性があります。
  - フローがモニターモードで暗号化されていない場合、フローに関連するサービスが除外として追加されている限り、強制モードで準拠している可能性があります。

バージョン 11 で利用できるフローログ Amazon EC2 タグフィールドに固有の制限:

- リソースのタグがフローログサブスクリプションの所有者によって所有されていない場合、タグフィールドは計算されません。

  例えば、サブネット (SubnetA) を別のアカウント (AccountB) と共有し、ネットワークインターフェイス上のタグフィールドで SubnetA のフローログサブスクリプションを作成するとします。設定したキーを使用してタグ付けされたネットワークインターフェイスが AccountB によって起動すると、サブスクリプションは AccountB によって起動されたネットワークインターフェイス

のトラフィックログを受信しますが、サブスクリプションによって設定されたタグフィールドはセキュリティ上の懸念により計算されません。必要に応じて、AccountB によって起動されたネットワークインターフェイスにタグを付けることでタグを表示できます。

- VPC/サブネットリソースレベルでタグフィールドを使用してフローログサブスクリプションを作成すると、タグ付けされていないネットワークインターフェイス用に生成されたトラフィックも、サブスクリプションに対して配信されます。タグフィールドの値は、タグ付けされていないリソースでは「-」になります。
- Auto Scaling グループは、「-」という名前のタグを表示できません。これは、値が欠落/欠落していることを示す予約文字であるためです。
- タグフィールドを使用してフローログサブスクリプションを作成すると、お客様に代わっていくつかのリソースが作成されます。タグ値の更新を消費するために、EventBridge マネージドルールが作成され、フローログサービスにタグの変更が送信されます。これらの EventBridge マネージドルールは、タグフィールドを使用するすべてのサブスクリプションが削除されると、自動的にクリーンアップされます。これらの EventBridge マネージドルールを手動で削除しないでください。手動で削除した場合、タグ値の更新に大幅な遅延が発生します。この作成/クリーンアップを制御し、タグ値へのアクセスを提供するために、サービスリンクロールがアカウントに作成されます。詳細については、「VPC フローログのサービスリンクロールの使用」を参照してください。
- Amazon CloudWatch Logs の解析と Athena の解析の制約により、タグ値のすべての特殊文字は UTF-8 パーセントエンコードを使用してエンコードされます。これは、url_decode を使用して Athena でネイティブにデコードすることも、任意の URL または URI デコーダーでデコードすることもできます。
- 新しいサブスクリプションが作成されてから最初の 1 時間は、タグ値が欠落しているか不正確である可能性があります。この最初の 1 時間の後、タグ値はタグ付けされたリソースの値を 1 分単位で正確に反映できるようになります。
- 同じリソースで同じ 1 秒以内に複数のタグが変更された場合、更新が失われ、最大 1 時間の古いタグ値が発生する可能性があります。
- Auto Scaling グループタグフィールドの場合、アカウントで 1 つ以上の有効な CloudTrail 証跡が必要です。有効な証跡がないと、Auto Scaling グループのタグ値が古くなったり不正確になったりする可能性があります。

next-hop- フィールドに固有の制限:

- ネクストホップネットワークインターフェイスが next-hop-az-id 以外のフローログサブスクリプションの所有者によって所有されていない場合、ネクストホップフィールドは計算されません。

- ネクストホップフィールドにネットワークインターフェイス (インターネットゲートウェイへのトラフィックなど) がない場合、ネクストホップフィールドは使用できません。
- ネクストホップフィールドは、クロスリージョントラフィックでは使用できません。
- ネクストホップフィールドは、一部のネットワークサービス (トランジットゲートウェイや Network Load Balancer など) からの入力トラフィックには使用できません。
- トラフィックがミドルボックス (トランジットゲートウェイや Network Load Balancer など) を通過する場合、ネクストホップネットワークインターフェイスは、トラフィックの最終送信先ではなく、ミドルボックス (Transit Gateway アタッチメントなど) に関連付けられたネットワークインターフェイスです。

料金

フローログを発行すると、提供されたログに対するデータインジェスト料金とアーカイブ料金が適用されます。提供されたログの発行に伴う料金の詳細については、「Amazon CloudWatch 料金表」を開き、[Logs] (ログ) を選択して [Vended Logs] (提供されたログ) を参照してください。

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リソースに適用できます。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状況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。詳細については次を参照してください:

- 「AWS Billing ユーザーガイド」の「コスト配分タグの使用」
- 「Amazon CloudWatch Logs ユーザーガイド」の「Amazon CloudWatch Logs のロググループにタグを付ける」
- 「Amazon Simple Storage Service ユーザーガイド」の「S3 バケットタグでのコスト配分タグの使用」
- 「Amazon Data Firehose デベロッパーガイド」の「Tagging Your Delivery Streams」

フローログの使用

Amazon EC2 および Amazon VPC の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

- 1. IAM を使用してフローログの使用を管理する
- 2. フローログの作成

- 3. フローログへのタグ付け
- 4. フローログの削除
- コマンドラインの概要

1. IAM を使用してフローログの使用を管理する

デフォルトでは、 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ポリシーがアタッチされた IAM ロールを作成できます。

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。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DeleteFlowLogs",
        "ec2:CreateFlowLogs",
        "ec2:DescribeFlowLogs"
      ],
      "Resource": "*"
    }
  ]
}
```

詳細については、「the section called “Amazon VPC で IAM を使用する方法”」を参照してください。

2. 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。フローログを作成するときは、フローログの送信先を指定する必要があります。詳細については次を参照してください:

- the section called “CloudWatch Logs に発行するフローログの作成”
- the section called “Amazon S3 に発行するフローログの作成”
- the section called “Amazon Data Firehose に発行するフローログの作成”

3. フローログへのタグ付け

フローログに対するタグの追加または削除は随時行うことができます。

フローログのタグを管理するには

1. 次のいずれかを行います。
   - Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。ナビゲーションペインで、[Network Interfaces] を選択してください。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。
   - Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。ナビゲーションペインで、[Your VPCs (お使いの VPC)] を選択します。VPC のチェックボックスをオンにします。
   - Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。ナビゲーションペインで、[サブネット] を選択してください。サブネットのチェックボックスをオンにします。
2. [Flow Logs] (フローログ) を選択します。
3. [Actions] (アクション)、[Manage tags] (タグの管理) を選択します。
4. タグを追加するには、[Add new tag] (新しいタグを追加) をクリックし、キーと値を入力します。タグを削除するには[削除] を選択してください。
5. タグの追加や削除が完了したら、[Save] (保存) を選択します。

4. フローログの削除

フローログはいつでも削除できます。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。

フローログを削除しても、送信先からログデータが削除されたり、送信先リソースが変更されたりすることはありません。送信先サービスのコンソールを使用して、既存のフローログデータを送信先から直接削除し、送信先リソースをクリーンアップする必要があります。

フローログを削除するには

1. 次のいずれかを行います。
   - Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。ナビゲーションペインで、[Network Interfaces] を選択してください。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。
   - Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。ナビゲーションペインで、[Your VPCs (お使いの VPC)] を選択します。VPC のチェックボックスをオンにします。
   - Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。ナビゲーションペインで、[サブネット] を選択してください。サブネットのチェックボックスをオンにします。
2. [Flow Logs] (フローログ) を選択します。
3. [Actions] (アクション)、[Delete flow logs] (フローログの削除) を選択します。
4. 確認を求められたら、**`delete`** と入力し、[Delete] (削除) を選択します。

コマンドラインの概要

このページで説明しているタスクは、コマンドラインを使用して実行できます。

フローログの作成

- create-flow-logs (AWS CLI)
- New-EC2FlowLog (AWS Tools for Windows PowerShell)

フローログの説明

- describe-flow-logs (AWS CLI)
- Get-EC2FlowLog (AWS Tools for Windows PowerShell)

フローログへのタグ付け

- create-tags および delete-tags (AWS CLI)
- New-EC2Tag および Remove-EC2Tag (AWS Tools for Windows PowerShell)

フローログの削除

- delete-flow-logs (AWS CLI)
- Remove-EC2FlowLog (AWS Tools for Windows PowerShell)

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。Amazon CloudWatch は、モニタリングとオブザーバビリティの総合サービスです。さまざまな AWS リソースや独自のアプリケーションおよびサービスからメトリクス、ログ、イベントデータを収集して追跡します。CloudWatch を使用することで、リソースの使用状況、アプリケーションのパフォーマンス、稼働状況を可視化でき、システム全体のパフォーマンス変化や潜在的問題を特定して対処することができます。CloudWatch を使用すると、アラームの設定やログおよびメトリクスの可視化が行え、クラウドリソースの収集および最適化に自動的に対応できます。これは、クラウドベースのインフラストラクチャとアプリケーションの信頼性、可用性、パフォーマンスを確実なものとするために欠かせないツールです。

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。

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。

CloudWatch Logs の詳細については、「Amazon CloudWatch Logs ユーザーガイド」の「CloudWatch Logs に送信されたログ」を参照してください。

料金

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インジェスト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 料金表」を開き、[Logs] (ログ) を選択して [Vended Logs] (提供されたログ) を参照してください。

内容

- CloudWatch Logs へのフローログ発行のための IAM ロール
- CloudWatch Logs に発行するフローログの作成
- CloudWatch Logs を使用してフローログレコードを表示する
- フローログレコードの検索
- CloudWatch Logs でのフローログレコードの処理

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。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Resource": "*"
    }
  ]
}
```

ロールに以下の信頼ポリシーがあることを確認します。これにより、フローログサービスがロールを引き受けることができます。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "vpc-flow-logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Confused Deputy Problem (混乱した使節の問題) から自分を守るために、aws:SourceAccount および aws:SourceArn の条件キーを使用することをお勧めします。例えば、前述の信頼ポリシーに次の条件ブロックを追加できます。ソースアカウントはフローログの所有者であり、ソース ARN はフローログ ARN です。フローログ ID が不明な場合は、ARN の不明部分をワイルドカード (*) に置き換え、フローログ作成後にポリシーを更新できます。

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account_id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:ec2:region:account_id:vpc-flow-log/flow-log-id"
    }
}
```

フローログの IAM ロールの作成

前述のように既存のロールを更新できます。また、以下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することもできます。このロールは、フローログの作成時に指定します。

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、ポリシー を選択してください。

3. [ポリシーの作成] を選択します。
4. [ポリシーの作成] ページで、次の操作を行います。
   a. [JSON] を選択します。
   b. このウィンドウのコンテンツを、このセクションの冒頭にあるアクセス許可ポリシーに置き換えてください。
   c. [次へ] を選択します。
   d. ポリシーの名前、説明 (省略可能)、タグを入力し、[ポリシーの作成] をクリックします。
5. ナビゲーションペインで Roles (ロール) を選択してください。
6. [ロールの作成] を選択してください。
7. [Trusted entity type] (信頼されたエンティティのタイプ) で、[Custom trust policy] (カスタム信頼ポリシー) を選択します。[Custom trust policy] (カスタム信頼ポリシー) で、`"Principal": {},` を次のように置き換え、[Next] (次へ) を選択します。

```
"Principal": {
   "Service": "vpc-flow-logs.amazonaws.com"
},
```

8. [Add permissions] (アクセス許可の追加) ページで、この手順で先ほど作成したポリシーの横にあるチェックボックスを選択し、[Next] (次へ) を選択します。
9. ロールの名前を入力し、オプションで説明を入力します。
10. [ロールの作成] を選択します。

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 ロールを使用するユーザーとして実行する場合は、そのロールが `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。

前提条件

リクエストを作成するために使用している IAM プリンシパルに `iam:PassRole` を呼び出すアクセス許可があることを確認してください。

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::111122223333:role/flow-log-role-name"
        }
    ]
}
```

コンソールを使用してフローログを作成するには

1. 次のいずれかを行います。
   - Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。ナビゲーションペインで、[Network Interfaces] を選択してください。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。
   - Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。ナビゲーションペインで、[Your VPCs (お使いの VPC)] を選択します。VPC のチェックボックスをオンにします。
   - Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。ナビゲーションペインで、[サブネット] を選択してください。サブネットのチェックボックスをオンにします。
2. [アクション]、[フローログの作成] を選択します。
3. [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。
4. [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。
5. [送信先] で、[Send to CloudWatch Logs (CloudWatch ログへの送信)] を選択します。
6. [送信先ロググループ] には、既存のロググループの名前を選択するか、新しいロググループの名前を入力します。名前を入力すると、記録するトラフィックがある場合にロググループが作成されます。

7. [サービスアクセス]では、CloudWatch Logs にログを発行する権限を持つ既存の IAM サービスロールを選択するか、新しいサービスロールを作成することを選択します。
8. [L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 のデフォルト形式] を選択します。
   - カスタム形式を使用するには、[カスタム形式] を選択し、[ログ形式] からフィールドを選択します。
9. [追加のメタデータ]で、Amazon ECS からのメタデータをログ形式に含めるかどうかを選択します。
10. (オプション) フローログにタグを適用するには、[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-EC2FlowLog (AWS Tools for Windows PowerShell)

次の AWS CLI の例では、指定したサブネット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、指定されたロググループに配信されます。`--deliver-logs-permission-arn` パラメータは、CloudWatch Logs への発行に必要な IAM ロールを指定します。

```
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 を使用してフローログレコードを表示する

CloudWatch Logs 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。

コンソールを使用して 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 コンソールの https://console.aws.amazon.com/cloudwatch/ を開いてください。
2. ナビゲーションペインで、[ログ]、[ロググループ] の順に選択します。

3. フローログを含むロググループの名前を選択して、その詳細ページを開きます。
4. フローログレコードを含むログストリームの名前を選択します。詳細については、「フローログレコード」を参照してください。

コマンドラインを使用して 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

- get-log-events (AWS CLI)
- Get-CWLLogEvent (AWS Tools for Windows PowerShell)

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ーを使用すると、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 コンソールの https://console.aws.amazon.com/cloudwatch/ を開いてください。
2. ナビゲーションペインで、[ログ]、[ロググループ] の順に選択します。
3. フローログを含むロググループを選択し、検索するネットワークインターフェイスが分かっている場合は、ログストリームを選択します。または、[Search log group] (ロググループの検索) を選択します。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. [イベントのフィルター] に以下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、フィルターを変更します。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]
```

CloudWatch Logs でのフローログレコードの処理

フローログレコードは、CloudWatch Logs で収集した他のログイベントのように処理することができます。モニタリングログデータとメトリックフィルターの詳細については、「Amazon CloudWatch Logs ユーザーガイド」の「フィルターを使用したログイベントからのメトリクスの作成」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ーとアラームの作成

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ーを作成する必要があります。次に、メトリクスフィルター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ーを作成し、フィルタのアラームを作成するには

1. CloudWatch コンソールの https://console.aws.amazon.com/cloudwatch/ を開いてください。
2. ナビゲーションペインで、[ログ]、[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、[アクション]、[メトリクスフィルターの作成] を選択します。
4. [Filter pattern] (フィルターパターン) で、次の文字列を入力します。

   ```
   [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。

6. 準備ができたら、[次へ] を選択します。
7. 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。
8. 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。
9. [アラームの作成] を選択します。
10. 作成したメトリクス名を選択し、その後 [メトリクスの選択] を選択します。
11. アラームを以下のように設定して、[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。
    - [TimeSinceLastActive が次の場合...] で、[以上] 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. [次へ] を選択します。
13. [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。[次へ] を選択します。
14. 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。
15. アラームの設定が終わったら、[アラームの作成] 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 (Simple Storage Service) は、スケーラブルで耐久性のあるオブジェクトストレージサービスです。ウェブ上のどの場所からも、任意の量のデータを保存して取得することができるように設計されています。S3 は、データバージョニング、暗号化、アクセスコントロールの機能が組み込まれており、業界をリードする耐久性と可用性を備えています。

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。監視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。

フローログに使用する Amazon S3 バケットの作成方法については、「Amazon S3 ユーザーガイド」の「バケットの作成」を参照してください。

VPC フローログの取り込み、フローログの処理、フローログの可視化に関する詳細は、AWS ソリューションライブラリ内「OpenSearch を利用した統合ログ記録」を参照してください。

CloudWatch Logs の詳細については、「Amazon CloudWatch Logs ユーザーガイド」の「Amazon S3 に送信されたログ」を参照してください。

料金

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インジェスト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 料金表」を開き、[Logs] (ログ) を選択して [Vended Logs] (提供されたログ) を参照してください。

内容

- フローログファイル
- フローログのための Amazon S3 バケットのアクセス許可
- SSE-KMS に使用する必須のキーポリシー
- Amazon S3 ログファイルのアクセス許可
- Amazon S3 に発行するフローログの作成
- Amazon S3 を使ってフローログレコードを表示する

フローログファイル

VPC フローログは、VPC で送受信される IP トラフィックに関するデータをログレコードに収集し、それらのレコードをログファイルに集約してから、5 分間隔でログファイルを Amazon S3 バケットに発行します。複数のファイルが公開され、各ログファイルに過去 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の一部またはすべてが含まれている場合があります。

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 は列指向データ形式です。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。

> **Note**
>
> gzip 圧縮を使用した Parquet 形式のデータが集計期間あたり 100 KB 未満の場合、Parquet ファイルのメモリ要件により、Parquet 形式のデータの保存には gzip 圧縮によるプレーンテキストよりも多くの容量が必要になる可能性があります。

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/aws-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/aws-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。

フローログを作成するユーザーがバケットを所有し、そのバケットに PutBucketPolicy および GetBucketPolicy 許可を持っている場合、次のポリシーが自動的にその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。

それ以外の場合は、バケット所有者が、フローログ作成者の AWS アカウント ID を指定して、このポリシーをバケットに追加しなければ、フローログの作成は失敗します。詳細については、「Amazon Simple Storage Service ユーザーガイド」の「バケットポリシーの使用」を参照してください。

JSON

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

```
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012",
                    "s3:x-amz-acl": "bucket-owner-full-control"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:*"
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:logs:us-east-1:123456789012:*"
                }
            }
        }
    ]
}
```

my-s3-arn に指定する ARN は、Hive と互換性のある S3 のプレフィックスを使用するかどうかによって異なります。

- デフォルトのプレフィックス

```
arn:aws:s3:::bucket_name/optional_folder/AWSLogs/account_id/*
```

- Hive 互換の S3 プレフィックス

```
arn:aws:s3:::bucket_name/optional_folder/AWSLogs/aws-account-id=account_id/*
```

ベストプラクティスは、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、これらのアクセス許可を付与することです。また、aws:SourceAccount および aws:SourceArn 条件キーを使用して、混乱した使節の問題から保護することもベストプラクティスです。ソースアカウントはフローログの所有者であり、ソース ARN は、ログサービスのワイルドカード (*) ARN です。

ログ配信サービスは、HeadBucket Amazon S3 API アクションを呼び出して、S3 バケットの存在と場所を確認することに注意してください。このアクションを呼び出すアクセス許可をログ配信サービスに付与する必要はありません。S3 バケットが存在することを確認できない場合でも、サービスは VPC フローログを配信します。ただし、CloudTrail ログの HeadBucket への呼び出しには AccessDenied エラーが発生します。

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 S3 バケットに格納された 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-S3 を選択した場合、追加の設定は必要ありません。Amazon S3 が暗号化キーを処理します。

SSE-KMS を選択した場合は、カスタマーマネージドキー ARN を使用する必要があります。キー ID を使用すると、フローログの作成時に LogDestination が配信できない エラーが発生する可能性があります。また、ログ配信アカウントが S3 バケットに書き込めるように、カスタマーマネージドキーのキーポリシーを更新する必要があります。SSE-KMS での使用に必要なキーポリシーについては、「Amazon CloudWatch Logs ユーザーガイド」の「Amazon S3 バケットのサーバー側の暗号化」を参照してください。

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、許可を持ちません。ログ配信アカウントには、READ および WRITE 許可があります。詳細については、「Amazon S3 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。

前提条件

フローログを作成する IAM プリンシパルには、フローログを宛先の Amazon S3 バケットに公開するために、次のアクセス許可が付与されている IAM ロールを使用している必要があります。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogDelivery",
        "logs:DeleteLogDelivery"
      ],
      "Resource": "*"
    }
  ]
}
```

コンソールを使用してフローログを作成するには

1. 次のいずれかを行います。

- Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。ナビゲーションペインで、[Network Interfaces] を選択してください。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。
- Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。ナビゲーションペインで、[Your VPCs (お使いの VPC)] を選択します。VPC のチェックボックスをオンにします。
- Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。ナビゲーションペインで、[サブネット] を選択してください。サブネットのチェックボックスをオンにします。

2. [アクション]、[フローログの作成] を選択します。
3. [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。
   - [承諾] - 承諾されたトラフィックのみをログに記録します。
   - [却下] - 却下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。
4. [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。
5. [送信先] で、[Amazon S3 バケットへの送信] を選択します。
6. [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」を参照してください。
7. [ログレコード形式] 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 のデフォルト形式] を選択します。
   - カスタム形式を作成するには、[カスタム形式] を選択します。[ログの形式] 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。

8. [追加のメタデータ]で、Amazon ECS からのメタデータをログ形式に含めるかどうかを選択します。
9. [ログファイル形式] 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 は列指向データ形式です。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。
10. (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]、[有効化] を選択します。
11. (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。
12. (オプション) フローログにタグを追加するには、[新しいタグを追加] を選択し、タグのキーと値を指定します。
13. [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 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-EC2FlowLog (AWS Tools for Windows PowerShell)

次の AWS CLI の例では、指定した VPC 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、そのフローログを指定した 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/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 を使ってフローログレコードを表示する

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。

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 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. Amazon S3 コンソール (https://console.aws.amazon.com/s3/) を開きます。
2. バケットの名前を選択して、その詳細ページを開きます。
3. ログファイルのあるフォルダに移動します。例: *#######*/AWSLogs/*account_id*/vpcflowlogs/*#####*/*#*/*#*/*#*/。
4. ファイル名の横にあるチェックボックスをオンにしてから、[Download] (ダウンロード) を選択します。

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。

Amazon Data Firehose 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 Data Firehose に発行できます。Amazon Data Firehose は、リアルタイムのデータストリームを収集、変換し、さまざまな AWS データストアや分析サービスに配信するフルマネージドサービスです。データインジェストをユーザーに代わって処理します。

VPC フローログの作業には Firehose が役立ちます。VPC フローログは、VPC のネットワークインターフェイスに出入りする IP トラフィックに関する情報を取得します。このデータは、セキュリティモニタリング、パフォーマンス分析、規制コンプライアンスに欠かせないものです。ただし、ログデータの継続的なフローの保存と処理を管理することは、複雑で、リソースを大量に消費します。

Firehose を VPC フローログと統合すれば、こうしたデータを Amazon S3 や Amazon Redshift などの希望する宛先に配信することができます。Firehose は VPC フローログの取り込み、変換、配信を

処理できるようにスケールするため、作業負担の軽減につながります。それによりユーザーは、基盤となるインフラストラクチャを気にすることなくログの分析とインサイトの取得に専念できます。

また、Firehose はデータ変換、圧縮、暗号化などの機能を備えており、VPC フローログの処理パイプラインの効率と安全性を強化することができます。VPC フローログに Firehose を使用することで、データ管理を簡素化し、ネットワークトラフィックのデータからさまざまなインサイトを入手できます。

Amazon Data Firehose に発行すると、フローログデータは Amazon Data Firehose 配信ストリームにプレーンテキスト形式で発行されます。

料金

標準の取り込み料金と配信料金が適用されます。詳細については、「Amazon CloudWatch 料金表」を開き、[Logs] (ログ) を選択して [Vended Logs] (提供されたログ) を参照してください。

内容

- クロスアカウント配信のための IAM ロール
- Amazon Data Firehose に発行するフローログの作成

クロスアカウント配信のための IAM ロール

Amazon Data Firehose に発行する場合、監視するリソースと同じアカウント (ソースアカウント) または別のアカウント (送信先アカウント) にある配信ストリームを選択できます。Amazon Data Firehose へのフローログのクロスアカウント配信を有効にするには、ソースアカウントと送信先アカウントに IAM ロールをそれぞれ作成する必要があります。

ロール

- ソースアカウントロール
- 送信先アカウントロール

ソースアカウントロール

ソースアカウントで、次のアクセス許可を付与するロールを作成します。この例のロールの名前は `mySourceRole` ですが、このロールには別の名前を選択できます。最後のステートメントにより、送信先アカウントのロールがこのロールを引き受けることができるようになります。条件ステートメントにより、このロールは指定されたリソースを監視する場合に限り、ログ配信サービスだけに渡さ

れます。ポリシーを作成するときに、監視する VPC、ネットワークインターフェイス、またはサブネットを条件キー `iam:AssociatedResourceARN` で指定します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::123456789012:role/mySourceRole",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "delivery.logs.amazonaws.com"
                },
                "StringLike": {
                    "iam:AssociatedResourceARN": [
                        "arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-00112233344556677"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:GetLogDelivery"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "sts:AssumeRole",
            "Resource": "arn:aws:iam::111122223333:role/
AWSLogDeliveryFirehoseCrossAccountRole"
        }
    ]
}
```

このロールに以下の信頼ポリシーがあることを確認します。これにより、ログ配信サービスがロールを引き受けることができます。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

ソースアカウントから、以下に説明する手順に従ってロールを作成します。

ソースアカウント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、ポリシー を選択してください。
3. [ポリシーの作成] を選択します。
4. [ポリシーの作成] ページで、次の操作を行います。
   a. [JSON] を選択します。
   b. このウィンドウのコンテンツを、このセクションの冒頭にあるアクセス許可ポリシーに置き換えてください。
   c. [次へ] を選択します。
   d. ポリシーの名前、説明 (省略可能)、タグを入力し、[ポリシーの作成] をクリックします。
5. ナビゲーションペインで Roles (ロール) を選択してください。
6. [ロールの作成] を選択してください。
7. [Trusted entity type] (信頼されたエンティティのタイプ) で、[Custom trust policy] (カスタム信頼ポリシー) を選択します。[Custom trust policy] (カスタム信頼ポリシー) で、`"Principal": {},` を次のように置き換え、ログ配信サービスを指定します。[次へ] を選択します。

```
"Principal": {
   "Service": "delivery.logs.amazonaws.com"
},
```

8. [Add permissions] (アクセス許可の追加) ページで、この手順で先ほど作成したポリシーの横にあるチェックボックスを選択し、[Next] (次へ) を選択します。
9. ロールの名前を入力し、オプションで説明を入力します。
10. [ロールの作成] を選択します。

送信先アカウントロール

送信先アカウントで、AWSLogDeliveryFirehoseCrossAccountRole で始まる名前のロールを作成します。このロールには、以下のアクセス許可が必要です。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iam:CreateServiceLinkedRole",
          "firehose:TagDeliveryStream"
      ],
      "Resource": "*"
    }
  ]
}
```

このロールに次の信頼ポリシーがあることを確認します。これにより、ソースアカウントで作成したロールがこのロールを引き受けることができます。

JSON

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/mySourceRole"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

送信先アカウントから、以下に説明する手順に従ってロールを作成します。

送信先アカウント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、ポリシー を選択してください。
3. [ポリシーの作成] を選択します。
4. [ポリシーの作成] ページで、次の操作を行います。
   a. [JSON] を選択します。
   b. このウィンドウのコンテンツを、このセクションの冒頭にあるアクセス許可ポリシーに置き換えてください。
   c. [次へ] を選択します。
   d. AWSLogDeliveryFirehoseCrossAccountRole で始まるポリシーの名前を入力し、[ポリシーの作成] を選択します。
5. ナビゲーションペインで Roles (ロール) を選択してください。
6. [ロールの作成] を選択してください。
7. [Trusted entity type] (信頼されたエンティティのタイプ) で、[Custom trust policy] (カスタム信頼ポリシー) を選択します。[Custom trust policy] (カスタム信頼ポリシー) で、`"Principal": {},` を次のように置き換え、ソースアカウントロールを指定します。[次へ] を選択します。

```
"Principal": {
   "AWS": "arn:aws:iam::source-account:role/mySourceRole"
},
```

8. [Add permissions] (アクセス許可の追加) ページで、この手順で先ほど作成したポリシーの横にあるチェックボックスを選択し、[Next] (次へ) を選択します。
9. ロールの名前を入力し、オプションで説明を入力します。
10. [ロールの作成] を選択します。

Amazon Data Firehose 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。

前提条件

- 送信先の Amazon Data Firehose 配信ストリームを作成します。ソースとして [Direct Put] を使用します。詳細については、「Creating an Amazon Data Firehose Delivery Stream」を参照してください。
- フローログを作成するアカウントは、フローログを Amazon Data Firehose に発行するための以下のアクセス許可を付与する IAM ロールを使用している必要があります。

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery",
                "iam:CreateServiceLinkedRole",
                "firehose:TagDeliveryStream"
            ],
            "Resource": "*"
        }
    ]
}
```

- フローログを別のアカウントに発行する場合は、「the section called “クロスアカウント配信のための IAM ロール”」の説明に従って必要な IAM ロールを作成します。

Amazon Data Firehose に発行するフローログを作成するには

1. 次のいずれかを行います。
   - Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。ナビゲーションペインで、[Network Interfaces] を選択してください。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。
   - Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。ナビゲーションペインで、[Your VPCs (お使いの VPC)] を選択します。VPC のチェックボックスをオンにします。
   - Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。ナビゲーションペインで、[サブネット] を選択してください。サブネットのチェックボックスをオンにします。
2. [アクション]、[フローログの作成] を選択します。
3. [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。
   - [Accepted] (承認) - 承認されたトラフィックのみをログに記録します
   - [Rejected] (拒否) - 拒否されたトラフィックのみをログに記録します
   - [All] (すべて) - 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。
4. [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。
5. [Destination] (送信先) で、次のいずれかのオプションを選択します。
   - [同じアカウント内の Amazon Data Firehose に送信] — 配信ストリームと監視するリソースは同じアカウントにあります。
   - [異なるアカウント内の Amazon Data Firehose に送信] — 配信ストリームと監視するリソースは異なるアカウントにあります。
6. Amazon Data Firehose 配信ストリーム名には、作成した配信ストリームを選択します。
7. [クロスアカウント配信のみ] [サービスアクセス]では、ログを発行するアクセス許可を持つクロスアカウント配信用の既存の IAM サービスロールを選択します。または、[許可を設定] を選択して IAM コンソールを開き、サービスロールを作成します。
8. [ログレコード形式] 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 のデフォルト形式] を選択します。

- カスタム形式を作成するには、[カスタム形式] を選択します。[ログの形式] 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。

9. [追加のメタデータ]で、Amazon ECS からのメタデータをログ形式に含めるかどうかを選択します。
10. (オプション) フローログにタグを適用するには、[タグの追加] をクリックします。
11. [フローログの作成] をクリックします。

コマンドラインを使用して Amazon Data Firehose に発行す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-EC2FlowLog (AWS Tools for Windows PowerShell)

次の AWS CLI の例では、指定した VPC 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、そのフローログを同じアカウント内の指定された Amazon Data Firehose 配信ストリームに配信します。

```
aws ec2 create-flow-logs --traffic-type ALL \
  --resource-type VPC \
  --resource-ids vpc-00112233344556677 \
  --log-destination-type kinesis-data-firehose \
  --log-destination arn:aws:firehose:us-
east-1:123456789012:deliverystream/flowlogs_stream
```

次の AWS CLI の例では、指定した VPC 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、そのフローログを別のアカウント内の指定された Amazon Data Firehose 配信ストリームに配信します。

```
aws ec2 create-flow-logs --traffic-type ALL \
  --resource-type VPC \
  --resource-ids vpc-00112233344556677 \
  --log-destination-type kinesis-data-firehose \
  --log-destination arn:aws:firehose:us-
east-1:123456789012:deliverystream/flowlogs_stream \
  --deliver-logs-permission-arn arn:aws:iam::source-account:role/mySourceRole \
  --deliver-cross-account-role arn:aws:iam::destination-account:role/
AWSLogDeliveryFirehoseCrossAccountRole
```

フローログを作成することで、配信ストリーム用に設定した送信先からフローログデータを取得することができます。

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。

オプション

- VPC フローログと Athena の統合を効率化および自動化できます。VPC を通過するトラフィックに関するインサイトを取得するには、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します。
- Athena を使用して独自のクエリを作成できます。詳細については、「Amazon Athena ユーザーガイド」の「Amazon Athena を使用したフローログのクエリ」を参照してください。

料金

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。

定義済みクエリを使用するには

- コンソールを使用した CloudFormation テンプレートの生成
- AWS CLI を使用した CloudFormation テンプレートの生成
- 事前定義されたクエリを実行する

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。

要件

- 選択したリージョンで、AWS Lambda と Amazon Athena がサポートされている必要があります。
- 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。
- フローログのログレコード形式には、実行する特定の事前定義されたクエリで使用されるフィールドを含める必要があります。

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イス] 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、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。

CloudFormation テンプレートによって作成されたリソース

- 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、「事前に定義されたクエリ」を参照してください
- 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。

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。
2. 左側のナビゲーションの [Query editor] (クエリエディタ) を選択します。[Workgroup] (ワークグループ) 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。
3. [Saved queries] (保存したクエリ) を選択してクエリを選択し、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。使用可能な事前定義されたクエリの一覧については、「事前に定義されたクエリ」を参照してください。
4. [Query results] (クエリ結果) で、クエリの結果を表示します。

事前に定義されたクエリ

Athena の名前付きクエリの完全なリストを次に示します。テンプレートを生成する際に提供される事前定義されたクエリは、フローログのログレコード形式の一部であるフィールドによって異なります。そのため、これらの事前定義されたクエリのすべてがテンプレートに含まれない場合があります。

- 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 — 管理ポートでリクエストを処理するアプリケーションによって記録された、トラフィックが最も多い上位 10 個の IP アドレス。

- 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 Identity and Access Management

AWS ではセキュリティ認証情報を使用して、ユーザーを識別し、AWS リソースへのアクセスを付与します。AWS Identity and Access Management (IAM) の機能を使用して、他のユーザー、サービス、およびアプリケーションが完全にまたは制限付きで AWS リソースを使用できるように許可します。その際、お客様のセキュリティ認証情報は共有されません。

このセクションでは、VPC フローログのために作成された AWS サービスリンクロールと、VPC フローログのサービスリンクロールにアタッチされたマネージドポリシーについて説明します。AWS IAM ロールおよびポリシーについての詳細については、IAM ユーザーガイドの[ロールに関する用語と概念]を参照してください。

VPC の Identity and Access Management に関する詳細については、Amazon VPC ユーザーガイドの「Amazon VPC の Identity and Access Management」を参照してください。

内容

- VPC フローログのサービスリンクロールの使用
- VPC フローログの AWS マネージドポリシー

VPC フローログのサービスリンクロールの使用

VPC フローログは、AWS Identity and Access Management (IAM) のサービスリンクロールを使用します。サービスリンクロールは、VPC フローログに直接リンクされた一意のタイプの IAM ロールです。サービスリンクロールは、VPC フローログによって事前定義されており、お客様の代わりにサービスから他の AWS のサービスを呼び出す必要のある許可がすべて含まれています。

サービスリンクロールを使用することで、必要な許可を手動で追加する必要がなくなるため、VPC フローログの設定が簡単になります。VPC フローログは、サービスリンクロールのアクセス許可を定義します。また、別途定義されている場合を除き、VPC フローログのみがそのロールを引き受けることができます。定義される許可は信頼ポリシーと許可ポリシーに含まれており、その許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスリンクロールを削除するには、最初に関連リソースを削除する必要があります。こうすることで、不注意によりリソースにアクセスするための許可を削除することがなくなるため、VPC フローログのリソースが保護されます。

サービスにリンクされたロールをサポートする他のサービスについては、「IAM と連携する AWS サービス」を参照して、[サービスにリンクされたロール] 列が [はい] になっているサービスを探してください。サービスリンクロールに関するドキュメントをサービスで表示するには、リンクで [はい] を選択します。

VPC フローログのサービスリンクロールのアクセス許可

VPC フローログは、AWSServiceRoleForVPCFlowLogs という名前のサービスリンクロールを使用します。このサービスリンクロールにより、VPC フローログはユーザーに代わって EventBridge マネージドルールを作成および管理し、DescribeTag API を呼び出して、タグフィールドを含むフローログサブスクリプションのリソースに関連付けられた EC2 タグ値の更新を自動的に追跡できるようになります。

AWSServiceRoleForVPCFlowLogs サービスリンクロールは、次のサービスを信頼してロールを引き受けます。

- `vpc-flow-logs.amazonaws.com`

AWSVPCFlowLogsServiceRolePolicy という名前のロールアクセス許可ポリシーを使用すると、VPC フローログは指定されたリソースに対して次のアクションを完了できます。

- アクション: タグ値を検証するために EC2 Autoscaling グループで `autoscaling:DescribeTags` を行う。

  アクション: タグ値を検証するために EC2 インスタンスと ElasticNetworkInterfaces で `tag:GetResources` を行う。

  アクション: タグ変更イベントに関連する詳細タイプのソース `aws.tag` および `aws.autoscaling` の新しいマネージドルールで `events:PutRule` を行う。

  アクション: `VPCFlowLogsEC2TagsManagedRule` および/または `VPCFlowLogsASGTagsManagedRule` という名前の VPC フローログによって作成されたマネージドルールで `events:DeleteRule` を行う。

  アクション: `VPCFlowLogsEC2TagsManagedRule` および/または `VPCFlowLogsASGTagsManagedRule` という名前の VPC フローログによって作成されたマネージドルールで `events:DescribeRule` を行う。

  アクション: `VPCFlowLogsEC2TagsManagedRule` および/または `VPCFlowLogsASGTagsManagedRule` という名前の VPC フローログによって作成されたマネージドルールで `events:PutTargets` を行う。

  アクション: `VPCFlowLogsEC2TagsManagedRule` および/または `VPCFlowLogsASGTagsManagedRule` という名前の VPC フローログによって作成されたマネージドルールで `events:RemoveTargets` を行う。

ユーザー、グループ、またはロールにサービスリンクロールの作成、編集、または削除を許可するには、アクセス許可を設定する必要があります。詳細についてはIAM ユーザーガイド の「サービスにリンクされた役割のアクセス許可」を参照してください。

VPC フローログのサービスリンクロールの作成

サービスリンクロールを手動で作成する必要はありません。ログ形式のタグフィールドと、AWS マネジメントコンソール、AWS CLI、または AWS API の関連付けられた TagFieldSpecifications パラ

メータを使用して CreateFlowLogs を作成すると、VPC フローログによってサービスリンクロールが作成されます。

> **Important**
>
> このサービスリンクロールはこのロールでサポートされている機能を使用する別のサービスでアクションが完了した場合にアカウントに表示されます。詳細については、「AWS アカウント に新しいロールが表示される」を参照してください。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は同じ方法でアカウントにロールを再作成できます。ログ形式のタグフィールドと関連付けられた TagFieldSpecifications パラメータを使用して CreateFlowLogs を作成すると、VPC フローログによって再度サービスリンクロールが作成されます。

AWSServiceRoleForVPCFlowLog ユースケースでサービスリンクロールを作成する場合にも、IAM コンソールを使用できます。AWS CLI または AWS API では、`vpc-flow-logs.amazonaws.com` サービス名を使用してサービスリンクロールを作成します。詳細については、IAM ユーザーガイドの「サービスリンクロールの作成」を参照してください。このサービスリンクロールを削除しても、同じ方法でロールを再作成できます。

VPC フローログのサービスリンクロールの編集

VPC フローログでは、AWSServiceRoleForVPCFlowLogs のサービスリンクロールを編集することはできません。サービスリンクロールの作成後は、さまざまなエンティティがロールを参照する可能性があるため、ロール名を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「サービスリンクロールの編集」を参照してください。

VPC フローログのサービスリンクロールの削除

サービスリンクロールを必要とする機能やサービスが不要になった場合は、ロールを削除することをお勧めします。そうすることで、使用していないエンティティがアクティブにモニタリングされたり、メンテナンスされたりすることがなくなります。ただし、手動で削除する前に、サービスリンクロールのリソースをクリーンアップする必要があります。

> **Note**
>
> リソースを削除する際に、VPC フローログサービスでそのロールが使用されている場合、削除できないことがあります。失敗した場合は数分待ってから操作を再試行してください。

AWSServiceRoleForVPCFlowLogs が使用する VPC フローログリソースを削除するには

1. ログ形式のタグフィールドを使用するすべての VPC フローログサブスクリプションを削除してください。
2. アカウントのすべてのタグサブスクリプションが削除されたことをVPC フローログが処理し、ログエンリッチメントをサポートするために作成された EventBridge マネージドルールを自動的にクリーンアップするまで、少なくとも 1 時間待ちます。

サービスリンクロールを IAM で手動削除するには

IAM コンソール、AWS CLI、または AWS API を使用して、AWSServiceRoleForVPCFlowLogs サービスリンクロールを削除します。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの削除」を参照してください。

VPC フローログのサービスリンクロールをサポートするリージョン

VPC フローログは、サービスを利用できるすべてのリージョンで、サービスリンクロールの使用をサポートしているわけではありません。AWSServiceRoleForVPCFlowLogs ロールは、次のリージョンで使用できます。

リージョン名	リージョン識別子	VPC フローログでのサポート
米国東部 (バージニア北部)	us-east-1	はい
米国東部 (オハイオ)	us-east-2	はい
米国西部 (北カリフォルニア)	us-west-1	はい
米国西部 (オレゴン)	us-west-2	はい
アフリカ (ケープタウン)	af-south-1	はい

リージョン名	リージョン識別子	VPC フローログでのサポート
アジアパシフィック (香港)	ap-east-1	はい
アジアパシフィック (ジャカルタ)	ap-southeast-3	はい
アジアパシフィック (ムンバイ)	ap-south-1	はい
アジアパシフィック (大阪)	ap-northeast-3	はい
アジアパシフィック (ソウル)	ap-northeast-2	はい
アジアパシフィック (シンガポール)	ap-southeast-1	はい
アジアパシフィック (シドニー)	ap-southeast-2	はい
アジアパシフィック (東京)	ap-northeast-1	はい
カナダ (中部)	ca-central-1	はい
欧州 (フランクフルト)	eu-central-1	はい
欧州 (アイルランド)	eu-west-1	はい
欧州 (ロンドン)	eu-west-2	はい
欧州 (ミラノ)	eu-south-1	はい
欧州 (パリ)	eu-west-3	はい
欧州 (ストックホルム)	eu-north-1	はい
中東 (バーレーン)	me-south-1	はい
中東 (アラブ首長国連邦)	me-central-1	はい
南米 (サンパウロ)	sa-east-1	はい
AWS GovCloud (US-East)	us-gov-east-1	はい
AWS GovCloud (米国西部)	us-gov-west-1	はい

VPC フローログの AWS マネージドポリシー

VPC フローログを使用しており、タグフィールドおよび関連付けられた TagFieldSpecifications パラメータを使用してサブスクリプションを作成した場合、AWSVPCFlowLogsServiceRolePolicy マネージドポリシーが IAM アカウント内に自動的に作成され、AWSServiceRoleForVPCFlowLogs のサービスリンクロールにアタッチされます。

このマネージドポリシーによって、VPCフローログでは次のことが実行できるようになります。

- EventBridge マネージドルールを作成および管理し、VPC フローログサービスにタグ更新イベントを送信する。
- 顧客に代わって API を呼び出し、ログエンリッチメントのタグ値の鮮度を検証する。

次の例は、作成されるマネージドポリシーの詳細を表示したものです。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleOnSpecificSourcesAndDetailTypes",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": [
                "arn:aws:events:*:*:rule/VPCFlowLogsEC2TagsManagedRule",
                "arn:aws:events:*:*:rule/VPCFlowLogsASGTagsManagedRule"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "events:source": [
                        "aws.tag",
                        "aws.autoscaling"
                    ],
                    "events:detail-type": [
                        "AWS API Call via CloudTrail",
                        "Tag Change on Resource"
                    ]
                },
                "Null": {
                    "events:source": "false",
```

```
                    "events:detail-type": "false"
                },
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "AllowOtherOperationsOnRulesManagedByVPCFlowLogs",
            "Effect": "Allow",
            "Action": [
                "events:DeleteRule",
                "events:DescribeRule",
                "events:PutTargets",
                "events:RemoveTargets"
            ],
            "Resource": [
                "arn:aws:events:*:*:rule/VPCFlowLogsEC2TagsManagedRule",
                "arn:aws:events:*:*:rule/VPCFlowLogsASGTagsManagedRule"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "AllowDescribeTagsOnAllEC2Resources",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources",
                "autoscaling:DescribeTags"
            ],
            "Resource": "*"
        }
    ]
}
```

前の例の 1 つ目のステートメントにより、VPC フローログはタグ変更イベントに関連する詳細タイプのソース `aws.tag` と `aws.autoscaling` について、AWS アカウント内に EventBridge マネージドルールを作成できるようになります。

前の例の 2 つ目のステートメントにより、VPC フローログは、VPCFlowLogsEC2TagsManagedRule および/または VPCFlowLogsASGTagsManagedRule という名前のリソースの AWS アカウント内で作成されたマネージドルールのライフサイクルを制御できるようになります。

前の例の 3 つ目のステートメントにより、VPC フローログは顧客に代わってタグ API を呼び出し、ログエンリッチメントのタグ値の鮮度を検証できるようになります。

AWS マネージドポリシー: AWSVPCFlowLogsServiceRolePolicy

AWSVPCFlowLogsServiceRolePolicy 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、VPC フローログが EventBridge マネージドルールを作成および管理し、ユーザーに代わって DescribeTag API を呼び出して、タグフィールドを含むフローログサブスクリプションのリソースに関連付けられた EC2 タグ値の更新を自動的に追跡できるようにするアクセス許可を付与します。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の「AWSVPCFlowLogsServiceRolePolicy」を参照してください。

AWS マネージドポリシーに対する更新

VPC フローログの AWS マネージドポリシーの更新に関する詳細を、このサービスがこれらの変更の追跡を開始した以降の分について示します。

変更	説明	日付
the section called “AWSVPCFlowLogsServiceRolePolicy ” - 新しいポリシー	新しい AWSVPCFlowLogsServiceRolePolicy ポリシーにより、VPC フローログは EventBridge マネージドルールを作成および管理し、ユーザーに代わって DescribeTag API を呼び出して、タグフィールドを含むフローログサブスクリプションのリソースに関連付けられた EC2 タグ値の更新を自動的に追跡できるようになります。	March 31, 2026

変更	説明	日付
VPC フローログが変更の追跡を開始	VPC フローログは AWS マネージドポリシーの変更の追跡を開始しました。	March 31, 2026

VPC フローログ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
- 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
- 「LogDestinationNotFoundException」または「Access Denied for LogDestination」エラー
- Amazon S3 バケットポリシーの制限の超過
- LogDestination が配信できない
- フローログのデータサイズが請求データと一致しない

不完全なフローログレコード

問題

フローログレコードが不完全であるか、公開が終了しています。

原因

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るか、SkipData エントリが存在する可能性があります。

ソリューション

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] タブを確認します。自分と共有されている VPC またはサブネットのフローログを記述することはできません。ただし、自分と共有されている VPC またはサブネットで作成したネットワークインターフェイスのフローログを記述することは可能です。エラーがある場合は、[ステータス] 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。

ステータスに表示される可能性のあるエラー値は、以下のとおりです。

- `Rate limited`: このエラーは、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を超えた場合に発生することがあります。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 サービスクォータ」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、「CloudWatch Logs へのフローログ発行のための IAM ロール」を参照してください。
- `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

問題

フローログを作成すると、Amazon VPC または Amazon EC2 コンソールにフローログが `Active` として表示されます。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。

考えられる原因

- フローログはまだ作成中である。場合によっては、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。

ソリューション

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

問題

フローログを作成すると、`Access Denied for LogDestination` または `LogDestinationNotFoundException` エラーが発生します。

考えられる原因

- Amazon S3 バケットにデータを発行するフローログを作成している場合、このエラーは、指定された S3 バケットが見つからなかったことを示します。また、バケットポリシーがバケットへのログの配信を許可していないことを意味することもあります。
- Amazon CloudWatch Logs にデータを発行するフローログを作成している場合、このエラーは、IAM ロールでロググループへのログの配信が許可されていないことを示します。

ソリューション

- Amazon S3 に配信する場合、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、バケットポリシーに必要な許可があり、ARN に正しいアカウント ID とバケット名が使用されていることを確認します。
- CloudWatch Logs に配信する場合は、IAM ロールに必要な許可があることを確認します。

Amazon S3 バケットポリシーの制限の超過

問題

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。

考えられる原因

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。

ソリューション

- 不要になったフローログエントリを削除して、バケット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。

LogDestination が配信できない

問題

フローログを作成しようとすると、`LogDestination <bucket name> is undeliverable` エラーが発生します。

考えられる原因

ターゲットの Amazon S3 バケットは、AWS KMS (SSE-KMS) によるサーバー側の暗号化を使って暗号化されます。バケットのデフォルトの暗号化は KMS キー ID です。

ソリューション

値は KMS キー ARN である必要があります。デフォルトの S3 暗号化タイプを KMS キー ID から KMS キー ARN に変更してください。詳細については、「Amazon Simple Storage Service ユーザーガイド」の「デフォルト暗号化の設定」を参照してください。

フローログのデータサイズが請求データと一致しない

問題

フローログの合計データサイズが、請求データが報告するサイズと一致しません。

考えられる原因

フローログに SKIPDATA エントリがある場合があります。SKIPDATA エントリの説明については、「データなしおよびスキップされたレコード」を参照してください。

ソリューション

log-status フィールドの異なるエントリがないかログのクエリを行うことで、ログエントリに SKIPDATA エントリが存在することを確認します。

SKIPDATA をチェックするサンプルクエリ:

CW Insights:

```
fields @timestamp, @message, @logStream, @log
| filter interfaceId = 'eni-123'
| stats count(*) by interfaceId, logStatus
| sort by interfaceId, logStatus
```

Athena:

```
SELECT log_status, interface_id, count(1)
FROM vpc_flow_logs
WHERE interface_id IN ('eni-1', 'eni-2', 'eni-3')
GROUP BY log_status, interface_id
```

VPC の CloudWatch メトリクス

Amazon VPC は VPC に関するデータを Amazon CloudWatch に公開します。VPC に関する統計情報を、メトリクスと呼ばれる時系列データの時間順のセットとして取得できます。メトリクスは監視対象の変数、データは時間の経過と共に変わる変数の値と考えることができます。詳細については、「Amazon CloudWatch ユーザーガイド」を参照してください。

内容

- NAU メトリクスとディメンション
- NAU の監視を有効または無効にする
- NAU CloudWatch アラームの例

NAU メトリクスとディメンション

ネットワークアドレスの使用状況 (NAU) は、仮想ネットワーク内のリソースに適用されるメトリクスで、VPC のサイズを計画およびモニタリングするのに役立ちます。NAU の監視には費用はかかりません。VPC の NAU またはピアリングされた NAU クォータを使い果たすと、新しい EC2 イ

ンスタンスを起動したり、Network Load Balancer、VPC エンドポイント、Lambda 関数、Transit Gateway アタッチメント、NAT ゲートウェイなどの新しいリソースをプロビジョニングしたりできなくなるため、NAU のモニタリングは役に立ちます。

VPC のネットワークアドレス使用状況の監視を有効にしている場合、Amazon VPC は NAU に関連するメトリクスを Amazon CloudWatch に送信します。VPC のサイズは、VPC に含まれるネットワークアドレス使用状況 (NAU) のユニットの数によって測定されます。

これらのメトリクスを使用して、VPC の増加率を把握したり、VPC がサイズ制限に達する時期を予測したり、サイズのしきい値を超えたときにアラームを発生させたりできます。

AWS/EC2 名前空間には、NAU の監視のための以下のメトリクスが含まれます。

メトリクス	説明
NetworkAddressUsage	VPC あたりの NAU 数。 レポート条件 • 24 時間毎。 ディメンション • 名前: Per-VPC Metrics、値: VPC ID。
NetworkAddressUsagePeered	VPC と、ピアリングされているすべての VPC の NAU 数。 レポート条件 • 24 時間毎。 ディメンション • 名前: Per-VPC Metrics、値: VPC ID。

AWS/Usage 名前空間には、NAU の監視のための以下のメトリクスが含まれます。

メトリクス	説明
ResourceCount	VPC あたりの NAU 数。 レポート条件 • 24 時間毎。 ディメンション • 名前: Service、値: EC2 • 名前: Type、値: Resource • 名前: Resource、値: VPC ID。 • 名前: Class、値: NetworkAddressUsage
ResourceCount	VPC と、ピアリングされているすべての VPC の NAU 数。 レポート条件 • 24 時間毎。 ディメンション • 名前: Service、値: EC2 • 名前: Type、値: Resource • 名前: Resource、値: VPC ID。 • 名前: Class、値: NetworkAddressUsagePeered
ResourceCount	VPC 全体の NAU 使用状況をまとめたビュー。 レポート条件 • 24 時間毎。

メトリクス	説明
	ディメンション • 名前: Service、値: EC2 • 名前: Type、値: Resource • 名前: Resource、値: VPC • 名前: Class、値: NetworkAddressUsage
ResourceCount	ピアリングされた VPC 全体の NAU 使用状況をまとめたビュー。 レポート条件 • 24 時間毎。 ディメンション • 名前: Service、値: EC2 • 名前: Type、値: Resource • 名前: Resource、値: VPC • 名前: Class、値: NetworkAddressUsagePeered

NAU の監視を有効または無効にする

CloudWatch で NAU メトリクスを表示するには、まず、監視する各 VPC で監視を有効にする必要があります。

NAU の監視を有効または無効にするには

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、[Your VPCs (お使いの VPC)] 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. [Actions] (アクション)、[Edit VPC settings] (VPC 設定の編集) を選択します。

5. 次のいずれかを行います：

   - 監視を有効にするには、[Network mapping units metrics settings] (ネットワークマッピングユニットのメトリクス設定)、[Enable network address usage metrics] (ネットワークアドレス使用状況メトリクスを有効にする) を選択します。
   - 監視を無効にするには、[Network mapping units metrics settings] (ネットワークマッピングユニットのメトリクス設定)、[Enable network address usage metrics] (ネットワークアドレス使用状況メトリクスを有効にする) をクリアします。

コマンドラインを使用して監視を有効または無効にするには

- modify-vpc-attribute (AWS CLI)
- Edit-EC2VpcAttribute (AWS Tools for Windows PowerShell)

NAU CloudWatch アラームの例

次の AWS CLI コマンドと .json の例を使用して、50,000 NAU をしきい値に設定した VPC の NAU 使用率を追跡する Amazon CloudWatch アラームおよび SNS 通知を作成することができます。この例で必要なのは、Amazon SNS トピックを作成することだけです。詳細については、[Amazon Simple Notification Service デベロッパーガイド] の [Amazon SNS の使用開始] を参照してください。

```
aws cloudwatch put-metric-alarm --cli-input-json file://nau-alarm.json
```

nau-alarm.json の例を次に示します。

```
{
    "Namespace": "AWS/EC2",
    "MetricName": "NetworkAddressUsage",
    "Dimensions": [{
        "Name": "Per-VPC Metrics",
        "Value": "vpc-0123456798"
    }],
    "AlarmActions": ["arn:aws:sns:us-west-1:123456789012:my_sns_topic"],
    "ComparisonOperator": "GreaterThanThreshold",
    "Period": 86400,
    "EvaluationPeriods": 1,
    "Threshold": 50000,
```

```
    "AlarmDescription": "Tracks NAU utilization of the VPC with 50k NAUs as the
 threshold",
    "AlarmName": "VPC NAU Utilization",
    "Statistic": "Maximum"
}
```

請求レポートと使用状況レポートに表示される Amazon VPC のコードを理解する

Amazon VPC をご利用中の場合、AWS の請求レポートと使用状況レポートに関連コードが表示されます。これらのコードを確認すると、Amazon VPC のコストと使用パターンを把握できます。コストの最適化には、経費の追跡と管理が不可欠です。

次の表は、請求および使用状況レポートに表示される Amazon VPC のコードを示しています。請求および使用状況レポートに使用されているリージョンコードのリストについては、「AWS Region billing codes」を参照してください。

請求コードの対象

- IP アドレス管理
- VPC エンドポイント
- Transit Gateway
- ネットワーク分析
- トラフィックのミラーリング
- VPC Lattice
- クロスアカウント/リージョンのリソース

関連リソース

- Amazon VPC の料金
- AWS PrivateLink 料金表
- AWS Transit Gateway 料金表
- Amazon VPC Lattice の料金

IP アドレス管理

Code	説明	単位	詳細度
region-PublicIPv4:InUseAddress	リソースでパブリック IPv4 アドレスが使用されていた時間。	時間	1 秒あたり
region-PublicIPv4:IdleAddress	リソースでパブリック IPv4 アドレスが使用されていなかった時間。	時間	1 秒あたり
region-PublicIPv4:ContiguousBlock	パブリック IPv4 アドレスの使用状況 (Amazon が提供する連続した IPv4 ブロックに含まれるアドレス)	時間	毎時
region-IPAddressManager-IP-Hours	IP アドレスが IPAM Advanced Tier によって管理されていた時間。	時間	毎時

VPC エンドポイント

Code	説明	単位	詳細度
region-VpcEndpoint-Hours	インターフェイス VPC エンドポイントがプロビジョニングされていた時間。	時間	毎時
region-VpcEndpoint-Bytes	インターフェイス VPC エンドポイントによって処理されたデータ。	GB	毎時

Code	説明	単位	詳細度
region-VpcEndpoint-GWLBE-Hours	Gateway Load Balancer エンドポイントがプロビジョニングされていた時間。	時間	毎時
region-VpcEndpoint-GWLBE-Bytes	Gateway Load Balancer エンドポイントによって処理されたデータ。	GB	毎時

Transit Gateway

Code	説明	単位	詳細度
region-TransitGateway-Hours	Transit Gateway アタッチメントの使用状況。	時間	毎時
region-TransitGateway-Bytes	Transit Gateway によって処理されたデータ。	GB	毎時
region-TGW-Multicast-Consumer-Bytes	マルチキャストレシーバーインスタンスによって処理されたデータ。	GB	毎時

ネットワーク分析

Code	説明	単位	詳細度
region-Analysis-Runs	Reachability Analyzer で分析されたネットワークパスの数。	カウント	分析ごと
region-NetworkInterface-Assessment	Network Access Analyzer で分析されたネットワークインターフェイスの数。	カウント	評価ごと

トラフィックのミラーリング

Code	説明	単位	詳細度
region-ENI-Mirror	ネットワークインターフェイスがトラフィックミラーリング用に設定されていた時間。	時間	毎時

VPC Lattice

Code	説明	単位	詳細度
region-VPCLattice-Service-Hourly	VPC Lattice サービスの実行時間。	時間	毎時
region-VPCLattice-DataPr	VPC Lattice サービスによって処理されたデータ。	GB	毎時

Code	説明	単位	詳細度
ocessing-Bytes			
region-VPCLattice-RequestCount-Free	無料の HTTP リクエストおよび TCP 接続。	カウント	毎時
region-VpcLattice-Service-Network-Resource-Hours	VPC Lattice サービスネットワークの実行時間。	時間	毎時

クロスアカウント/リージョンのリソース

Code	説明	単位	詳細度
region-VpcResource-Provider-Bytes	アカウント間またはリージョン間でプロバイダーリソースから転送されたデータ。	GB	毎時
region-VpcResource-Consumer-Bytes	アカウント間またはリージョン間でコンシューマーリソースによって転送されたデータ。	GB	毎時

VPC ネットワークアーキテクチャを説明する

Amazon VPC を使用すると、AWS クラウド内に、仮想プライベートクラウド (VPC) と言う、論理的に分離された仮想ネットワークを定義でき、個別の VPC を作成することで、ワークロードまたは組織エンティティごとにインフラストラクチャの分離が可能となります。VPC を設定するには、IP アドレス範囲の選択、ルーティングの設定、ネットワークゲートウェイの追加を行って、VPC を相互接続し、VPC とインターネットまたは独自の企業ネットワークとの接続を行います。VPC では、EC2 インスタンスや RDS インスタンスなどの AWS リソースを起動します。

次の表には、VPC ネットワークの主な特性の説明が記載されています。ネットワーク管理者は、このガイダンスを利用することで VPC ネットワークのアーキテクチャと設定を説明できます。こうした情報があれば、オンプレミスや別のクラウドプロバイダーを使用して同等の機能を持つネットワークを設定することが可能になります。

特性	説明
地理的場所	Amazon VPC は、世界中のすべての AWS リージョンでホストされています。顧客に最も近い場所に AWS リソースを配置するために、VPC ネットワークのリージョンを選択できます。
サブネット –	VPC に対して定義するサブネットによってネットワーク境界が定義され、AWS リソースの IP アドレスが決定されます。複数のアベイラビリティーゾーンにサブネットを追加して、リソースの可用性を高めることができます。
ネットワーク接続	VPC またはサブネットにアタッチして VPC ネットワークとその他のネットワーク (他の VPC または サブネット、インターネット、またはオンプレミスネットワーク) の間の接続を提供するゲートウェイ。
セキュリティコントロール	VPC にセキュリティグループを作成すると、コンピューティングリソース、データベースリソース、ロードバランサーといった関連リソース間のトラフィックを制御できます。また、各サブネットが持つネットワーク ACL によっ

特性	説明
	て、サブネットに出入りするトラフィックの制御が可能です。
トラフィック管理	ルーティングルールは、サブネット、VPC、外部拠点の間を流れるトラフィックフローを制御します。Elastic Load Balancing のロードバランサーを使用すると、受信トラフィックを複数のターゲット (EC2 インスタンス、コンテナ、Lambda 関数など) に分散できます。

地理的場所

Amazon VPC は、世界中のどの AWS リージョンでも利用できます。各リージョンは、個別の地理的領域です。ユーザーの大部分が存在する場所に近いリージョンにリソースの VPC を作成することで、ネットワークレイテンシーを低減できます。

Amazon EC2 Global View を使用して、すべてのリージョンの VPC をグラフィカルユーザーインターフェイスで一覧表示できます (同等のプログラムインターフェイスはありません)。Amazon VPC コンソール、AWS API、AWS コマンドラインインターフェイスでは、各リージョンの VPC と VPC リソースを個別に一覧表示する必要があります。

地理的場所が重要である理由

VPC の場所を特定すると、必要に応じて、機能的に同等のネットワークを、同じ場所に設定するか、異なる場所に設定するかを決定できます。

すべてのリージョンの VPC 概要を表示するには

1. Amazon EC2 Global View コンソール (https://console.aws.amazon.com/ec2globalview/home) を開きます。
2. [リージョンエクスプローラー] タブの [概要] で、[VPC] のリソース数を確認します。これには、VPC 数とリージョン数が含まれており、これは、AWS によって自動作成されるデフォルト VPC の数と、ユーザーが作成するデフォルト以外の VPC の数をともに示しています。下線付きのテキストを選択して、VPC の数がリージョン全体にどのように分散しているかを確認します。リージョンに VPC が 1 つしかない場合、それは、対象リージョンのデフォルト VPC である可能性が高くなります。

3. [グローバル検索] タブで、クライアントフィルター [リソースタイプ = Vpc] を選択します。リージョンまたはタグを指定すると、結果をさらにフィルタリングできます。

AWS CLI を使用してリージョン内の VPC 情報を取得する

次の describe-vpcs コマンドを使用します。このコマンドは、VPC が存在する各リージョンで実行する必要があります。--query パラメータは、VPC ID のみを出力に含めます。必要に応じて追加のフィールドを含めることができます。

```
aws ec2 describe-vpcs \
    --region us-east-2 \
    --query "Vpcs[*].VpcId"
```

各リージョンにはデフォルトの VPC が表示されます。デフォルトの VPC を使用していない場合は、次のフィルターを追加して、結果からデフォルトの VPC を除外きます。

```
--filters Name=is-default,Values=false
```

サブネット

サブネットは、VPC 内にある論理的なネットワーク境界です。サブネットを作成する際、IP アドレスのブロックを割り当てます。サブネットで起動するリソースには、サブネットの IP アドレスのブロックから IP アドレスが割り当てられます。IP アドレスがあると、リソースがローカルネットワークまたはインターネット経由で相互に通信できるようになります。

Amazon VPC コンソールのリソースマップにより、VPC のサブネットが可視化されます。

サブネットが重要である理由

サブネットを使用することでセキュリティ境界を実装し、アプリケーション層の間のトラフィックを制御できます。サブネットの IP アドレスを把握しておくことで、同等の機能を持つネットワーク内のリソースが、VPC ネットワーク内で通信できるものと同じクライアントやアプリケーションと通信可能であることを確実にできます。

リソースマップを使用して VPC のサブネットを表示するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [VPC] を選択します。
3. VPC のチェックボックスにチェックを入れます。

4. [リソースマップ] タブを選択します。
5. VPC ペインで、[詳細を表示] を選択します。[サブネット] ペインには、VPC 内のすべてのサブネット一覧とともに、IP アドレス範囲が表示されます。サブネットにカーソルを合わせると、関連するルートテーブルとネットワーク接続が強調表示されます。詳細を確認するには、リンクを選択してサブネットの詳細ページを開きます。

AWS CLI を使用して VPC のサブネットを記述する

次の describe-subnets コマンドを使用します。--filters パラメータは、指定された VPC のサブネットを示す検索の範囲を指定します。--query パラメータに含まれるのは、出力内の指定されたフィールドのみです。必要に応じて追加のフィールドを含めることができます。

```
aws ec2 describe-subnets \
    --filters Name=vpc-id,Values=vpc-1234567890abcdef0 \
    --query Subnets[*].
[SubnetId,AvailabilityZoneId,CidrBlock,Ipv6CidrBlockAssociationSet[0].Ipv6CidrBlock] \
    --output table
```

以下は出力の例です。各列は、サブネット ID、AZ ID、IPv4 アドレス範囲、最初の IPv6 アドレス範囲 (該当する場合) を示しています。

```
-------------------------------------------------------------------------------------
|                                  DescribeSubnets                                  |
+---------------------------+-----------+----------------+----------------------------+
|  subnet-0d2d1b81e0bc9c6d4 |  usw2-az1 |  10.0.144.0/20 |  2600:1f14:1e6e:a003::/64  |
|  subnet-0e01d500780bb7468 |  usw2-az1 |  10.0.16.0/20  |  2600:1f14:1e6e:a001::/64  |
|  subnet-0eb17d85f5dfd33b1 |  usw2-az2 |  10.0.128.0/20 |  2600:1f14:1e6e:a002::/64  |
|  subnet-0e990c67809773b19 |  usw2-az2 |  10.0.0.0/20   |  2600:1f14:1e6e:a000::/64  |
+---------------------------+-----------+----------------+----------------------------+
```

ネットワーク接続

Amazon VPC の接続オプションを使用すると、複数のアカウント、リージョン、リモートネットワーク内の VPC にまたがるネットワークを作成できます。

Amazon VPC コンソールのリソースマップでは、VPC で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、NAT ゲートウェイ、ゲートウェイ VPC エンドポイントが使用されているかどうかを確認できます。Transit Gateway、ピアリング接続、仮想プライベートゲートウェイ、その他のタイプの VPC エンドポイントは、使用されていてもリソースマップには表示され

ません。VPC のゲートウェイとピアリング接続の完全なリストは、コンソール、API、またはコマンドラインインターフェイスを使用して一度に 1 つを記述することで取得できます。

ネットワーク接続が重要である理由

VPC ネットワークで可能な接続を把握できていれば、同等の機能を持つネットワーク内のリソースと、同様のローカルおよびリモートリソースを通信させることができます。

リソースマップを使用して VPC のネットワーク接続を表示するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [VPC] を選択します。
3. VPC のチェックボックスにチェックを入れます。
4. [リソースマップ] タブを選択します。
5. VPC ペインで、[詳細を表示] を選択します。[ネットワーク接続] ペインには、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、NAT ゲートウェイ、ゲートウェイ VPC エンドポイントが一覧表示されます。リソースタイプが不明な場合は、ネットワーク接続のリンクアイコンにカーソルを合わせ、その結果の URL を確認します。この URL は、コンソールのリソースへのリンクであり、コンソールには、リソースタイプとリソース ID (例: internetGatewayId=igw-0123456780abcdef) が表示されています。

AWS CLI を使用して VPC のネットワーク接続情報を取得するには

1. 次の describe-internet-gateways コマンドを使用して、指定されたリージョンのインターネットゲートウェイ情報を取得します。--query パラメータに含まれるのは、出力内の指定されたフィールドのみです。必要に応じて追加のフィールドを含めることができます。

```
aws ec2 describe-internet-gateways \
    --region us-east-2 \
    --query InternetGateways[*].[Attachments[0].VpcId,InternetGatewayId] \
    --output table
```

以下は出力の例です。列はそれぞれ、VPC ID とインターネットゲートウェイ ID を示しています。

```
-----------------------------------------------------
|             DescribeInternetGateways             |
+------------------------+-------------------------+
```

```
|  None                   |  igw-04c61dba10EXAMPLE  |
|  vpc-0bf4c2739bEXAMPLE |  igw-09737a4029EXAMPLE  |
|  vpc-060415a18fEXAMPLE |  igw-0c562bd22aEXAMPLE  |
|  vpc-0ea9d41094EXAMPLE |  igw-0e06f7033dEXAMPLE  |
|  vpc-03b86de356EXAMPLE |  igw-0a9ff72d05EXAMPLE  |
+------------------------+-------------------------+
```

2. 次の describe-egress-only-internet-gateways コマンドを使用して、指定されたリージョンの egress-only インターネットゲートウェイ情報を取得します。--query パラメータに含まれるのは、出力内の指定されたフィールドのみです。必要に応じて追加のフィールドを含めることができます。

```
aws ec2 describe-egress-only-internet-gateways \
    --region us-east-2 \
    --query EgressOnlyInternetGateways[*].
[Attachments[0].VpcId,EgressOnlyInternetGatewayId] \
    --output table
```

以下は出力の例です。列はそれぞれ、VPC ID と Egress-Only インターネットゲートウェイ ID を示しています。

```
-------------------------------------------------------
|        DescribeEgressOnlyInternetGateways           |
+------------------------+----------------------------+
|  vpc-060415a18fEXAMPLE |  eigw-0b8ca558acEXAMPLE  |
+------------------------+----------------------------+
```

3. 次の describe-nat-gateways コマンドを使用して、指定されたリージョンの NAT ゲートウェイ情報を取得します。--query パラメータに含まれるのは、出力内の指定されたフィールドのみです。必要に応じて追加のフィールドを含めることができます。

```
aws ec2 describe-nat-gateways \
    --region us-east-2 \
    --query NatGateways[*].[VpcId,NatGatewayId,SubnetId] \
    --output table
```

以下は出力の例です。各列は、VPC ID、NAT ゲートウェイ ID、サブネット ID を示しています。

```
-------------------------------------------------------------------------------
```

```
|                              DescribeNatGateways                              |
+-------------------------+-------------------------+----------------------------+
|  vpc-060415a18fEXAMPLE |  nat-026316334aEXAMPLE  |  subnet-0eb17d85f5EXAMPLE  |
|  vpc-060415a18fEXAMPLE |  nat-0f08bc5f52EXAMPLE  |  subnet-0d2d1b81e0EXAMPLE  |
+-------------------------+-------------------------+----------------------------+
```

4. 次の describe-transit-gateway-vpc-attachments コマンドを使用して、指定されたリージョンの Transit Gateway VPC アタッチメント情報を取得します。--query パラメータに含まれるのは、出力内の指定されたフィールドのみです。必要に応じて追加のフィールドを含めることができます。

```
aws ec2 describe-transit-gateway-vpc-attachments \
    --region us-east-2 \
    --query TransitGatewayVpcAttachments[*].
[VpcId,TransitGatewayId,length(SubnetIds[])] \
    --output table
```

以下は出力の例です。各列は、VPC ID、Transit Gateway ID、サブネット数を示しています。

```
-------------------------------------------------------------
|         DescribeTransitGatewayVpcAttachments              |
+-------------------------+-------------------------+-----+
|  vpc-0bf4c2739bEXAMPLE |  tgw-055dc4e47bEXAMPLE  |  4 |
|  vpc-0ea9d41094EXAMPLE |  tgw-055dc4e47bEXAMPLE  |  2 |
+-------------------------+-------------------------+-----+
```

5. 次の describe-vpc-peering-connections コマンドを使用して、指定されたリージョンにある VPC のピアリング接続情報を取得します。--query パラメータに含まれるのは、出力内の指定されたフィールドのみです。必要に応じて追加のフィールドを含めることができます。

```
aws ec2 describe-vpc-peering-connections \
    --region us-east-2 \
    --query VpcPeeringConnections[*].[AccepterVpcInfo.VpcId,RequesterVpcInfo.VpcId]
 \
    --output table
```

以下は出力の例です。各列は、アクセプタ VPC ID、アクセプタ VPC 所有者、リクエスタ VPC ID、リクエスタ VPC 所有者を示しています。

```
-------------------------------------------------------------------------------
```

```
|                          DescribeVpcPeeringConnections
 |
+------------------------+--------------+------------------------+---------------
+
|  vpc-0ea9d41094EXAMPLE |  123456789012 |  vpc-03b86de356EXAMPLE |  123456789012
 |
+------------------------+--------------+------------------------+---------------
+
```

6. 次の describe-vpn-gateways コマンドを使用して、指定されたリージョンの仮想プライベートゲートウェイ情報を取得します。--query パラメータに含まれるのは、出力内の指定されたフィールドのみです。必要に応じて追加のフィールドを含めることができます。

```
aws ec2 describe-vpn-gateways \
    --region us-east-2 \
    --query VpnGateways[*].[VpcAttachments[0].VpcId,VpnGatewayId] \
    --output table
```

以下は出力の例です。列はそれぞれ、VPC ID と仮想プライベートゲートウェイ ID を示しています。

```
-----------------------------------------------------
|                DescribeVpnGateways                |
+-------------------------+-------------------------+
|  vpc-0bf4c2739bEXAMPLE |  vgw-0cb3226c4aEXAMPLE  |
+-------------------------+-------------------------+
```

7. 次の describe-vpc-endpoints コマンドを使用して、指定されたリージョンの VPC エンドポイント情報を取得します。--query パラメータに含まれるのは、出力内の指定されたフィールドのみです。必要に応じて追加のフィールドを含めることができます。

```
aws ec2 describe-vpc-endpoints \
    --region us-east-2 \
    --query 'VpcEndpoints[*].[VpcId,VpcEndpointType,ServiceName||
ServiceNetworkArn||ResourceConfigurationArn]' \
    --output table
```

以下は出力の例です。最初の列は VPC ID を示し、2 番目の列は VPC エンドポイントタイプを示しています。3 番目の列はエンドポイントタイプに応じて異なり、サービス名、リソース設定 ARN、サービスネットワーク ARN のいずれかが表示されます。

```
-------------------------------------------------------------------------------------------
|                                                    DescribeVpcEndpoints
                                                 |
+------------------------+-----------------
+-------------------------------------------------------------------------------------------
+
|  vpc-060415a18fcc9afde |  Interface      |  com.amazonaws.vpce.us-west-2.vpce-
svc-007832a03d60fc387                                |
|  vpc-060415a18fcc9afde |  Interface      |  com.amazonaws.vpce.us-west-2.vpce-
svc-007832a03d60fc387                                |
|  vpc-0bf4c2739bc05a694 |  Gateway        |  com.amazonaws.us-west-2.s3
                                                |
|  vpc-0ea9d410947d27b7d |  Interface      |  com.amazonaws.us-west-2.logs
                                                |
|  vpc-0bf4c2739bc05a694 |  Resource       |  arn:aws:vpc-lattice:us-
east-2:123456789012:resourceconfiguration/rcfg-07129f3acded87625  |
|  vpc-0bf4c2739bc05a694 |  ServiceNetwork |  arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetwork/sn-0808d1748faee0c1e           |
|  vpc-0bf4c2739bc05a694 |  ServiceNetwork |  arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetwork/sn-0808d1748faee0c1e           |
+------------------------+-----------------
+-------------------------------------------------------------------------------------------
+
```

セキュリティコントロール

Amazon VPC のセキュリティコントロールを使用することで、VPC と、VPC にデプロイされたリソースへのネットワークアクセスを決定できます。

これが重要な理由

サブネットとリソースに到達可能なインバウンドトラフィックと、サブネットおよびリソースからの送信を許可するトラフィックが決定していれば、機能的に同等のネットワークに必要なファイアウォールルールを計画できます。

セキュリティコントロール

- セキュリティグループ
- ネットワーク ACL

セキュリティグループ

セキュリティグループを使用すると、特定のインバウンドおよびアウトバウンドトラフィックをリソースレベルで許可できます。セキュリティグループは、VPC 内のリソースへのアクセスを制御する主要な仕組みとして使用します。

VPC のセキュリティグループ情報を取得するには

次の describe-security-groups コマンドを使用すると、指定した VPC のセキュリティグループが表示されます。

```
aws ec2 describe-security-groups \
    --filters Name=vpc-id,Values=vpc-1234567890abcdef0 \
    --query SecurityGroups[*].GroupId
```

セキュリティグループのインバウンドルールを取得するには

次の describe-security-group-rules コマンドを使用すると、指定したセキュリティグループのルールが表示されます。IsEgress には false を指定しています。

```
aws ec2 describe-security-group-rules \
    --filters Name=group-id,Values=sg-0abcdef1234567890 \
    --query 'SecurityGroupRules[?IsEgress==`false`]'
```

セキュリティグループのアウトバウンドルールを取得するには

次の describe-security-group-rules コマンドを使用すると、指定したセキュリティグループのルールが表示されます。IsEgress には true を指定しています。

```
aws ec2 describe-security-group-rules \
    --filters Name=group-id,Values=sg-0abcdef1234567890 \
    --query 'SecurityGroupRules[?IsEgress==`true`]'
```

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) を使用すると、特定のインバウンドおよびアウトバウンドトラフィックをサブネットレベルで許可または拒否できます。ネットワーク ACL は、適切なセキュリティグループを設定せずにリソースをデプロイしてしまった場合の多層防御として使用できます。

サブネットのネットワーク ACL を取得するには

次の describe-network-acls コマンドを使用すると、VPC のネットワーク ACL とそのサブネットの関連付けが表示されます。

```
aws ec2 describe-network-acls \
    --filters Name=vpc-id,Values=vpc-1234567890abcdef0 \
    --query "NetworkAcls[*].{ID:NetworkAclId,Subnets:Associations[].SubnetId}"
```

ネットワーク ACL のインバウンドルールを取得するには

次の describe-network-acls コマンドを使用すると、指定したネットワーク ACL のルールが表示されます。Egress には false を指定しています。

```
aws ec2 describe-network-acls \
    --network-acl-ids acl-0abcdef1234567890 \
    --query 'NetworkAcls[*].Entries[?Egress==`false`]'
```

ネットワーク ACL のアウトバウンドルールを取得するには

次の describe-network-acls コマンドを使用すると、指定したネットワーク ACL のルールが表示されます。Egress には true を指定しています。

```
aws ec2 describe-network-acls \
    --network-acl-ids acl-0abcdef1234567890 \
    --query 'NetworkAcls[*].Entries[?Egress==`true`]'
```

トラフィック管理

トラフィック管理の効果を高めるには、ルートテーブルによってネットワークレベルでルーティングを決定すると同時に、ロードバランシングによるアプリケーションレベルの分散戦略を実行します。

トラフィック管理が重要である理由

ネットワーク管理者は、セキュリティ境界を維持し、パフォーマンス要件を満たしながらトラフィックフローを最適化するように、サブネット、ルーティング、DNS 解決、負荷分散を設計する必要があります。VPC ネットワーク内のこれらのコンポーネントの設定を把握しておくことで、同等の機能を持つネットワーク内のリソースが、VPC ネットワーク内で通信できるものと同じクライアントやデバイスと通信可能であることを確実にできます。

トラフィック管理

- ルートテーブル
- DHCP オプションセット
- ロードバランサー

ルートテーブル

ルートテーブルを使用すると、サブネット、VPC、オンプレミスネットワーク、インターネットといったネットワークの境界を越えるトラフィックフローを制御できます。

Amazon VPC コンソールのリソースマップにより、VPC のルートテーブルが可視化されます。

リソースマップを使用して VPC のルートテーブルを表示するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで [VPC] を選択します。
3. VPC のチェックボックスにチェックを入れます。
4. [リソースマップ] タブを選択します。
5. [ルートテーブル] ペインには、VPC のすべてのルートテーブルが一覧表示されます。ルートテーブルにカーソルを合わせると、関連するサブネットとネットワーク接続が強調表示されます。詳細を確認するには、リンクを選択してルートテーブルの詳細ページを開きます。

ルートテーブルを記述する

describe-route-tables コマンドを使用すると、指定した VPC のルートテーブルとそのサブネットの関連付けを記述できます。

```
aws ec2 describe-route-tables \
    --filters Name=vpc-id,Values=vpc-1234567890abcdef0 \
    --query "RouteTables[*].{ID:RouteTableId,Subnets:Associations[].SubnetId}"
```

ルートテーブルのルートを取得するには

describe-route-tables コマンドを使用して、指定されたルートテーブルのルートを記述します。

```
aws ec2 describe-route-tables \
    --route-table-ids rtb-02ec01715bEXAMPLE \
```

```
    --query RouteTables[*].Routes
```

DHCP オプションセット

VPC には、さまざまなネットワーク設定を行うために使用できる DHCP オプションセットがあります。例えば、EC2 インスタンスが既存の DNS インフラストラクチャを使用して内部ホスト名を解決できるように、カスタム DNS サーバーを設定することができます。詳細については、「the section called “DHCP オプションセットの概念”」を参照してください。

VPC の DHCP オプションを記述する

describe-dhcp-options コマンドを使用して、指定された DHCP オプションを記述します。この例では、describe-vpcs コマンドを使用することで、指定された VPC の DHCP オプションの ID も取得します。

```
aws ec2 describe-dhcp-options \
    --dhcp-options-id "$(aws ec2 describe-vpcs \
        --vpc-id vpc-1234567890abcdef0 \
        --query Vpcs[].DhcpOptionsId --output text)"
```

以下は、デフォルトの DHCP オプションを使用する VPC の出力例です。

```
{
    "DhcpOptions": [
        {
            "OwnerId": "415546850671",
            "Tags": [],
            "DhcpOptionsId": "dopt-1234567890abcdef0",
            "DhcpConfigurations": [
                {
                    "Key": "domain-name",
                    "Values": [
                        {
                            "Value": "us-west-2.compute.internal"
                        }
                    ]
                },
                {
                    "Key": "domain-name-servers",
                    "Values": [
                        {
                            "Value": "AmazonProvidedDNS"
```

```
                        }
                    ]
                }
            ]
        }
    ]
}
```

ロードバランサー

ロードバランシングを使用すると、クライアントからの受信トラフィックを複数のターゲットに分散できます。ロードバランサーでターゲットの状態をモニタリングして、トラフィックを分散処理しているサーバー群から異常なターゲットを自動的に削除し、正常なターゲットのみが使用されるようにします。このため、アプリケーションの可用性とパフォーマンスが向上し、リソース使用率も最適化されます。詳細については、「Elastic Load Balancing ユーザーガイド」を参照してください。

ロードバランサーを記述する

describe-load-balancers コマンドを使用すると、指定された VPC のロードバランサーが表示されます。

```
aws elbv2 describe-load-balancers \
    --query 'LoadBalancers[?VpcId==`vpc-1234567890abcdef0`].LoadBalancerArn'
```

関連リソース

VPC ネットワークで使用されている可能性のあるオプションのサービスまたは機能を以下に示します。

- Direct Connect
- AWS Network Firewall
- IPAM
- トラフィックのミラーリング
- VPC フローログ

Amazon Virtual Private Cloud のセキュリティの責任の管理

AWS ではクラウドセキュリティが最優先事項です。セキュリティを最も重視する組織の要件を満たすために構築された 「AWS」 のデータセンターとネットワークアーキテクチャは、お客様に大きく貢献します。

セキュリティは、AWS とお客様とが共有する責務です。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 Cloud 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、サードパーティ監査者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 のサービスによって決まります。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 リソースの監視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。

内容

- Amazon Virtual Private Cloud のデータ保護の確保
- 転送時の VPC 暗号化を強制する
- Amazon VPC の Identity and Access Management
- Amazon VPC のインフラストラクチャセキュリティ
- セキュリティグループを使用して AWS リソースへのトラフィックを制御する
- ネットワークアクセスコントロールリストを使用して、サブネットのトラフィックを制御する
- Amazon Virtual Private Cloud での耐障害性
- Amazon Virtual Private Cloud のコンプライアンス検証
- VPC とサブネットへのパブリックアクセスをブロックする

- VPC のセキュリティのベストプラクティス

Amazon Virtual Private Cloud のデータ保護の確保

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があります。ユーザーは、このインフラストラクチャでホストされるコンテンツに対する管理を維持する責任があります。また、使用する「AWS のサービス」のセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーに関する詳細については、Data Privacy FAQ をご覧ください。欧州におけるデータ保護に関する情報については、General Data Protection Regulation (GDPR) Center 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」認証情報を保護し、AWS IAM アイデンティティセンター または AWS Identity and Access Management (IAM) を使用して個々のユーザー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、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 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 は必須ですが、TLS 1.3 を推奨します。
- AWS CloudTrail で API とユーザーアクティビティロギングを設定します。CloudTrail 証跡を使用して AWS アクティビティをキャプチャする方法については、「AWS CloudTrail ユーザーガイド」の「CloudTrail 証跡の使用」を参照してください。
- AWS のサービス内のすべてのデフォルトセキュリティコントロールに加え、AWS 暗号化ソリューションを使用します。
- Amazon Macie などの高度な管理されたセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- 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 にアクセスする際に FIPS 140-3 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。利用可能な 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) 140-3」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報を、タグ、または [名前] フィールドなどの自由形式のテキストフィールドに含めないことを強くお勧めします。これには、コンソール、API、AWS CLI、または AWS SDK を使用して、Amazon VPC または他の AWS のサービス で作業する場合も含まれます。タグ、または名前に使用される自由記述のテキストフィールドに入力し

たデータは、請求または診断ログに使用される場合があります。外部サーバーに URL を提供する場合、そのサーバーへのリクエストを検証できるように、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の確保

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、監視できます。

- セキュリティグループ: セキュリティグループは、リソースレベル (EC2 インスタンスなど) で特定のインバウンドおよびアウトバウンドトラフィックを許可します。インスタンスを起動する際、そのインスタンスに 1 つまたは複数のセキュリティグループを割り当てることができます。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際にセキュリティグループを指定しなかった場合、インスタンスはその 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、「セキュリティグループ」を参照してください。
- 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、サブネットレベルで特定のインバウンドおよびアウトバウンドトラフィックを許可または拒否します。詳細については、「ネットワークアクセスコントロールリストを使用して、サブネットのトラフィックを制御する」を参照してください。
- 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 フローログを使用した IP トラフィックの口グ記録
- 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。

転送時の VPC 暗号化を強制する

VPC 暗号化コントロールは、トラフィックフローの暗号化ステータスをモニタリングするための一元的な権限制御を提供するセキュリティおよびコンプライアンス機能です。これにより、クリアテ

キスト通信が可能なリソースを特定し、最終的にはリージョンでの VPC 内での転送、および複数の VPC 間での転送時に暗号化を強制するメカニズムを提供できるようになります。

VPC 暗号化コントロールでは、アプリケーションレイヤー暗号化と組み込み暗号化の両方を AWS Nitro System ハードウェアの転送機能で使用して、暗号化を確実に強制できるようにします。また、この機能により、ネイティブハードウェアレイヤー暗号化が最新の Nitro インスタンスから Fargate、Application Load Balancer、Transit Gateway などの他の AWS サービスにまで拡張されます。

この機能は、すべてのトラフィックの暗号化ステータスで可視性と制御を確保する必要がある、すべてのユーザー向けに設計されています。HIPAA、FedRamp、PCI DSS などのコンプライアンス基準を満たすためにデータ暗号化が最優先される業界では、特に便利な機能です。セキュリティ管理者とクラウドアーキテクトは、この機能を使用して AWS 環境全体で転送時の暗号化ポリシーを一元的に実践できます。

この機能は、モニタリングモードと強制モードという 2 つのモードで使用できます。

暗号化コントロールのモード

モニタリングモード

モニタリングモードでは、暗号化コントロールによって、VPC 内および VPC 間の AWS リソース間のトラフィックフローの暗号化ステータスが可視化されます。また、転送時に暗号化が強制されない VPC リソースを特定することもできます。VPC フローログを設定すると、トラフィックが暗号化されているかどうかを示すエンリッチされたフィールド (`encryption-status`) を出力できます。また、コンソールまたは `GetVpcResourcesBlockingEncryptionEnforcement` コマンドを使用して、転送時に暗号化が強制されていないリソースを特定することもできます。

> **Note**
>
> 既存の VPC は、最初はモニタリングモードでのみ有効にできます。これにより、クリアテキストトラフィックであるリソース、またはクリアテキストトラフィックが可能なリソースを可視化できます。VPC で強制モードを有効にすることができるのは、これらのリソースが暗号化の強制を開始したとき (または暗号化に対する除外を作成したとき) のみです。

強制モード

強制モードでは、VPC 暗号化コントロールにより、VPC 境界内で暗号化されていないトラフィックを許可する機能やサービスを使用できなくなります。既存の VPC では、強制モードで暗号化コント

ロールを直接有効にすることはできません。まず、モニタリングモードで暗号化コントロールを有効にし、非準拠のリソースを特定および変更のうえ転送時に暗号化を強制してから、強制モードを有効にする必要があります。ただし、新規 VPC の場合は、作成時に強制モードで暗号化コントロールを有効にすることができます。

有効にしても、強制モードでは、ネイティブの組み込み暗号化をサポートしていない古い EC2 インスタンスやインターネットゲートウェイなど、暗号化されていない VPC リソースの作成やアタッチはできません。暗号化が強制された VPC で非準拠のリソースを実行する場合は、そのリソースに対して除外を作成する必要があります。

トラフィックフローの暗号化ステータスのモニタリング

VPC フローログの `encryption-status` フィールドを使用して、VPC 内のトラフィックフローの暗号化ステータスを監査できます。以下のイベント値があります。

- `0` = 暗号化なし
- 1 = Nitro 暗号化 (VPC 暗号化コントロールによって管理)
- 2 = アプリケーション暗号化
  - AWS サービスへのインターフェイスエンドポイントに対する TCP ポート 443 のフロー*
  - ゲートウェイエンドポイントに対する TCP ポート 443 のフロー*
  - VPC エンドポイントを介して暗号化された Redshift クラスターへのフロー**
- 3 = Nitro 暗号化とアプリケーション暗号化の両方
- (-) = 不明な暗号化ステータスまたは VPC 暗号化コントロールがオフ

注記:

* インターフェイスエンドポイントとゲートウェイエンドポイントの場合、AWS は暗号化ステータスを引き受けるために使用されるポートを確認します (暗号化ステータスを判断するためにパケットデータは確認されません)。

** 指定された AWS マネージドエンドポイントの場合、AWS はサービス設定の TLS の要件に基づいて暗号化ステータスを決定します。

VPC フローログの制限事項

- VPC 暗号化コントロールのフローログを有効にするには、encryption-status フィールドを使用して新しいフローログを手動で作成する必要があります。encryption-status フィールドは、既存のフローログに自動的に追加されません。

- フローログの詳細を確認するには、${traffic-path} フィールドと ${flow-direction} フィールドをフローログに追加することをお勧めします。

  例:

```
aws ec2 create-flow-logs \
--resource-type VPC \
--resource-ids vpc-12345678901234567 \
--traffic-type ALL \
--log-group-name my-flow-logs \
--deliver-logs-permission-arn arn:aws:iam::123456789101:role/publishFlowLogs
--log-format '${encryption-status} ${srcaddr} ${dstaddr} ${srcport} ${dstport}
 ${protocol} ${traffic-path} ${flow-direction} ${reject-reason}'
```

VPC 暗号化コントロールの除外

VPC 暗号化コントロールの強制モードでは、VPC 内のすべてのリソースに暗号化を強制する必要があります。これにより、リージョンの AWS 内で確実に暗号化が実行されます。ただし、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など、お客様がエンドツーエンドの暗号化の設定と維持を担当する AWS のネットワーク以外への接続を許可するリソースが含まれている場合があります。これらのリソースを暗号化が強制された VPC で実行するには、リソースの除外を作成できます。除外では、顧客が暗号化の維持を担うリソース (通常はアプリケーションレイヤー) に対して監査可能な例外が作成されます。

VPC 暗号化コントロールでサポートされている除外は 8 つのみです。VPC にこれらのリソースがあり、強制モードに移行する場合は、モニタリングモードから強制モードに切り替えるときにこれらの除外を追加する必要があります。他のリソースは除外できません。これらのリソースの除外を作成することで、VPC を強制モードに移行できます。これらのリソースとの間で送受信されるトラフィックフローの暗号化については、お客様の責任となります。

- インターネットゲートウェイ
- NAT Gateway
- Egress-only インターネットゲートウェイ
- 暗号化が強制されていない VPC への VPC ピアリング接続 (詳細なシナリオについては、「VPC ピアリングサポート」セクションを参照してください)
- 仮想プライベートゲートウェイ
- VPC 内の Lambda 関数

- VPC Lattice
- Elastic File System

実装ワークフロー

1. モニタリングを有効化 - モニタリングモードで VPC 暗号化コントロールを作成します
2. トラフィックの分析 - フローログを確認して、トラフィックフローの暗号化ステータスをモニタリングします
3. リソースの分析 - コンソールまたは `GetVpcResourcesBlockingEncryptionEnforcement` コマンドを使用して、転送時に暗号化が強制されていないリソースを特定します。
4. 準備 (オプション) - 強制モードを有効にする場合に、リソースの移行と必要な除外について計画します
5. 強制 (オプション) - 必要な除外が設定された状態で強制モードに切り替えます
6. 監査 - フローログによる継続的なコンプライアンスのモニタリング

詳細なセットアップ手順については、ブログ記事「VPC 暗号化コントロールのご紹介: リージョンの VPC 内および VPC 間での転送中の暗号化の強制」を参照してください。

VPC 暗号化コントロールの状態

VPC 暗号化コントロールは、次のいずれかの状態になります。

作成中

VPC 暗号化コントロールが VPC に作成されています。

変更中

VPC 暗号化コントロールが VPC で変更されています

削除中

VPC 暗号化コントロールが VPC で削除されています

利用可能

VPC 暗号化コントロールが VPC でのモニタリングモードまたは強制モードの実装に成功しました

AWS サービスのサポートと互換性

暗号化に準拠するには、ハードウェアレイヤーまたはアプリケーションレイヤーでリソースが常に転送時の暗号化を強制する必要があります。ほとんどのリソースでは、お客様によるアクションは必要ありません。

自動コンプライアンスによるサービス

クロスリージョン接続の PrivateLinks など、PrivateLinks でサポートされているほとんどの AWS サービスが、アプリケーションレイヤーで暗号化されたトラフィックを受け入れます。これらのリソースを変更する必要はありません。AWS によって、アプリケーションレイヤーで暗号化されていないトラフィックは自動的にドロップされます。いくつかの例外として、Redshift クラスターなどがあります (プロビジョニング済みとサーバーレスの両方。基盤となるリソースを手動で移行する必要があります)。

自動的に移行するリソース

Network Load Balancer、Application Load Balancer、Fargate クラスター、EKS Control Plane は、モニタリングモードを有効にすると、暗号化をネイティブにサポートするハードウェアに自動的に移行します。これらのリソースを変更する必要はありません。AWS で自動的に移行が処理されます。

手動による移行が必要なリソース

特定の VPC リソースおよびサービスでは、基盤となるインスタンスタイプを選択する必要があります。最新の EC2 インスタンスでは、すべて転送中の暗号化がサポートされています。お使いのサービスで既に最新の EC2 インスタンスを使用している場合は、変更を加える必要はありません。コンソールまたは GetVpcResourcesBlockingEncryptionEnforcement コマンドを使用して、これらのサービスのいずれかが古いインスタンスを使用しているかどうかを特定できます。このようなリソースを特定する場合は、Nitro System ハードウェアのネイティブ暗号化をサポートする最新の EC2 インスタンスにアップグレードする必要があります。これらのサービスには、EC2 インスタンス、Auto Scaling グループ、RDS (すべてのデータベースおよびドキュメント DB)、プロビジョニングされた Elasticache、Amazon Redshift でプロビジョニングされたクラスター、EKS、ECS-EC2、プロビジョニングされた OpenSearch、EMR が含まれます。

互換性のあるリソース:

次のリソースは VPC 暗号化コントロールと互換性があります。

- Nitro ベースの EC2 インスタンス
- Network Load Balancer (制限付き)

- Application Load Balancer
- AWS Fargate クラスター
- Amazon Elastic Kubernetes Service (EKS)
- Amazon EC2 Auto Scaling グループ
- Amazon Relational Database Service (RDS: すべてのデータベース)
- Amazon ElastiCache のノードベースのクラスター
- Amazon Redshift でプロビジョニングされたクラスターとサーバーレスクラスター
- Amazon Elastic Container Service (ECS) - EC2 コンテナインスタンス
- Amazon OpenSearch Service
- Amazon Elastic MapReduce (EMR)
- Amazon Managed Streaming for Apache Kafka (Amazon MSK)
- VPC 暗号化コントロールにより、PrivateLink 経由でアクセスされるすべての AWS サービスに対して、アプリケーションレイヤーで暗号化が強制されます。アプリケーションレイヤーで暗号化されていないトラフィックは、VPC 内でホストされている PrivateLink エンドポイントと強制モードの暗号化コントロールによってドロップされます。

サービス固有の制限

Network Load Balancer の制限

TLS 設定: 含まれる VPC で暗号化コントロールを強制する場合に、TLS リスナーを使用して、暗号化と復号の作業をロードバランサーにオフロードすることはできません。ただし、TLS 暗号化と復号を実行するようにターゲットを設定できます。

プロビジョニングされた Redshift とサーバーレスの Redshift

既存のクラスター/エンドポイントが含まれる VPC で、顧客が強制モードに移行することはできません。Redshift で VPC 暗号化コントロールを使用するには、スナップショットからクラスターまたは名前空間を復元する必要があります。プロビジョニングされたクラスターの場合は、既存の Redshift クラスターのスナップショットを作成し、クラスタースナップショットからの復元オペレーションを使用してスナップショットから復元します。サーバーレスクラスターの場合は、既存の名前空間のスナップショットを作成し、サーバーレスワークグループのスナップショットからの復元オペレーションを使用してスナップショットから復元します。VPC 暗号化コントロールは、スナップショットと復元プロセスを実行しないと、既存のクラスターまたは名前空間で有効にできないことに注意してください。スナップショットの作成については、Amazon Redshift のドキュメントを参照してください。

Amazon MSK (Managed Streaming for Apache Kafka)

この機能は、独自の VPC の 4.1 に対応する、新しいクラスターでサポートされています。次の手順は、MSK で VPC 暗号化を使用する際に役立ちます。

- 顧客は、他に MSK クラスターのない VPC で VPC 暗号化を有効にします
- 顧客は、Kafka バージョン 4.1 を使用し、インスタンスタイプを M7g としてクラスターを作成します

Gateway Load Balancer と AWS Network Firewall

Gateway Load Balancer と AWS Network Firewall は、強制モードの VPC ではサポートされていません。これらのリソースが VPC に存在する場合は、VPC をモニターモードで実行する必要があります。

リージョンとゾーンの制限

- ローカルゾーンサブネット: 強制モードではサポートされていません - VPC から削除する必要があります

VPC ピアリングサポート

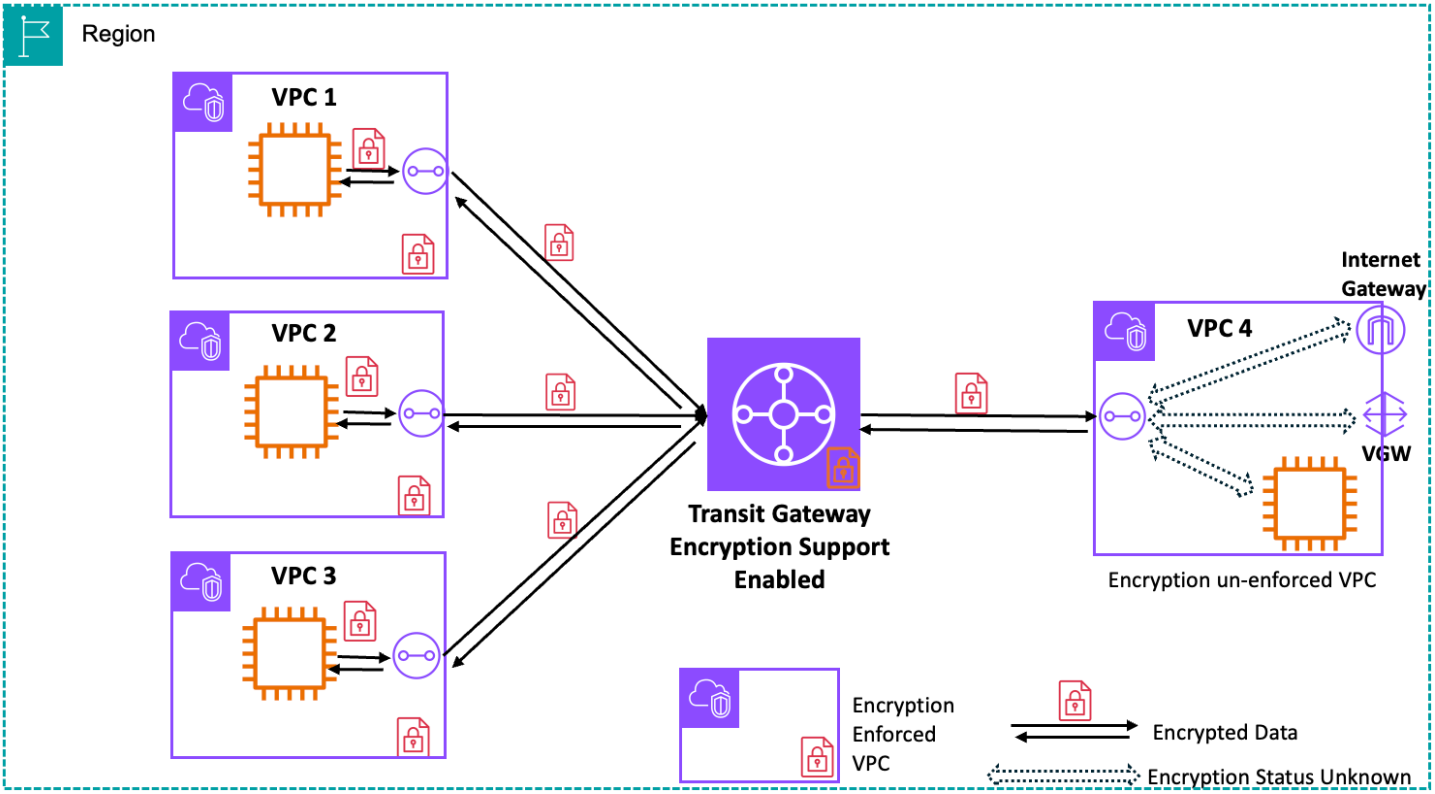
2 つの VPC 間で VPC ピアリングを使用して転送時の暗号化を確実に実行するには、2 つの VPC が同じリージョンに存在し、暗号化コントロールが除外なく強制モードで有効になっている必要があります。暗号化が強制された VPC を、別のリージョンに存在する、または強制モードで暗号化コントロールが有効になっていない (除外のない) 別の VPC にピアリングする場合は、ピアリングの除外を作成する必要があります。

2 つの VPC が強制モードで相互にピアリングしている場合、強制モードからモニタリングモードに変更することはできません。VPC 暗号化コントロールでモニタリングモードに変更する前に、まずピアリングの除外を作成する必要があります。

Transit Gateway 暗号化サポート

暗号化コントロールが有効になっている VPC 間のトラフィックを暗号化するには、Transit Gateway で暗号化サポートを明示的に有効にする必要があります。既存の Transit Gateway で暗号化を有効にすると、既存のトラフィックフローが中断されることなく、暗号化されたレーンへの VPC アタッチメントの移行がシームレスかつ自動的に実行されます。強制モード (除外なし) では、Transit

Gateway を介した 2 つの VPC 間のトラフィックは、100% 暗号化されたレーンを通過します。また、Transit Gateway での暗号化によって、異なる暗号化コントロールモードでも同様に 2 つの VPC を接続できます。これは、暗号化が強制されていない VPC に接続する暗号化コントロールを VPC に強制するときに使用する必要があります。このようなシナリオでは、VPC 間トラフィックを含む、暗号化が強制された VPC 内のすべてのトラフィックが暗号化されます。VPC 間トラフィックは、暗号化が強制された VPC 内のリソースと Transit Gateway の間で暗号化されます。さらに、暗号化は、トラフィックが強制されていない VPC で送信されるリソースに依存し、暗号化が保証されません (VPC は強制モードではないため)。すべての VPC は同じリージョンに存在する必要があります (詳細はこちらをご覧ください)。



- この図では、VPC 1、VPC 2、VPC3 の暗号化コントロールが強制モードで実行され、暗号化コントロールがモニタリングモードで実行されている VPC 4 に接続されます。
- VPC1、VPC2、VPC3 間のすべてのトラフィックが暗号化されます。
- 具体的には、VPC 1 のリソースと VPC 4 のリソース間のトラフィックが、Nitro System ハードウェアによって提供される暗号化を使用する Transit Gateway まで暗号化されます。さらに、暗号化ステータスは VPC 4 のリソースに依存し、暗号化される保証はありません。

Transit Gateway 暗号化サポートの詳細については、Transit Gateway ドキュメントを参照してください。

アカウントレベルで VPC 暗号化コントロールを有効にする

アカウントレベルの VPC 暗号化コントロールを使用すると、アカウント内のすべての VPC に適用される暗号化コントロールモードとリソースの除外を設定できます。これにより、各 VPC を個別に設定することなく、暗号化ポリシーを一元管理できます。

アカウントレベルの暗号化コントロールは、VPC レベルのコントロールと同じ 8 つのリソース除外をサポートします。インターネットゲートウェイ、NAT ゲートウェイ、Egress-Only インターネットゲートウェイ、VPC ピアリング、Virtual Private ゲートウェイ、Lambda、VPC Lattice、Elastic File System です。

アカウントレベルの暗号化コントロールモード

アカウントレベルで次のモードを設定できます。

アンマネージド型

VPC 暗号化コントロールは現在のモードのままで、VPC レベルで管理できます。

モニタリングモードの試行

アカウント内のすべての VPC のモニタリングモードへの移行を試みます。作成をブロックせずに、暗号化されていないリソースを識別します。

強制モードの試行

アカウント内のすべての VPC の強制モードへの移行を試みます。暗号化されていないリソースの作成をブロックします。

アカウントレベルの暗号化コントロールの状態

アカウントレベルの VPC 暗号化コントロールは、次のいずれかの状態になります。

default-state

アカウントレベルの VPC 暗号化コントロールが有効になっていません。

transitions-in-progress

アカウント内の VPC を指定されたモードへ移行しています。

transitions-partially-successful

1 つ以上の VPC を指定されたモードに移行できませんでした。

transitions-successful

すべての VPC を指定されたモードに正常に移行しました。

transitions-failed

すべての VPC を指定されたモードに移行できませんでした。

組織レベルで VPC 暗号化コントロールを有効にする

AWS Organizations を使用して組織内のアカウントを管理している場合は、AWS Organizations 宣言型ポリシーを使用して、組織内のアカウントに VPC 暗号化コントロールを適用できます。VPC 暗号化コントロールの宣言型ポリシーの詳細については、「AWS Organizations ユーザーガイド」の「サポートされている宣言型ポリシー」を参照してください。

料金

料金については、「Amazon VPC の料金」を参照してください。

AWS CLI コマンドリファレンス

セットアップと設定

- aws ec2 create-vpc-encryption-control
- aws ec2 modify-vpc-encryption-control
- aws ec2 modify-account-vpc-encryption-control
- aws ec2 describe-account-vpc-encryption-control
- aws ec2 tgw modify-transit-gateway

モニタリングとトラブルシューティング

- aws ec2 describe-vpc-encryption-controls
- aws ec2 get-vpc-resources-blocking-encryption-enforcement
- aws ec2 create-flow-logs
- aws ec2 describe-flow-logs
- aws logs query

クリーンアップ

- aws ec2 delete-vpc-encryption-control

その他のリソース

詳細なセットアップ手順については、ブログ記事「VPC 暗号化コントロールのご紹介: リージョンの VPC 内および VPC 間での転送中の暗号化の強制」を参照してください。

API 情報の詳細については、「EC2 API リファレンスガイド」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制御するために役立つ 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、追加費用なしで使用できる AWS のサービスです。

内容

- 対象者
- ID で認証する
- ポリシーを使用してアクセスを管理する
- Amazon VPC で IAM を使用する方法
- Amazon VPC ポリシーの例
- Amazon VPC の ID とアクセスのトラブルシューティング
- AWSAmazon Virtual Private Cloud の 管理ポリシー
- VPC のサービスにリンクされたロールの使用

対象者

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」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」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」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。ユーザーは、IAM ユーザー の AWS アカウントのルートユーザー として、または IAM ロールを引き受けることによって、認証される 必要があります。

AWS IAM アイデンティティセンター (IAM アイデンティティセンター)、シングルサインオン認証、Google/Facebook 認証情報などの ID ソースからの認証情報を使用して、フェデレーテッドアイデンティティとしてサインインできます。サインインの詳細については、「AWS サインイン ユーザーガイド」の「AWS アカウントにサインインする方法」を参照してください。

プログラムによるアクセスの場合、AWS はリクエストに暗号で署名するための SDK と CLI を提供します。詳細については、「IAM ユーザーガイド」の「API リクエストに対する AWS 署名バージョン 4」を参照してください。

AWS アカウント のルートユーザー

AWS アカウントを作成すると、すべての AWS のサービスとリソースに対する完全なアクセス権を持つ AWS アカウント ルートユーザーと呼ばれる 1 つのサインイン ID を使用して開始します。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザー認証情報を必要とするタスクについては、「IAM ユーザーガイド」の「ルートユーザー認証情報が必要なタスク」を参照してください。

IAM ユーザーとグループ

IAM ユーザーは、特定の個人やアプリケーションに対する特定のアクセス許可を持つアイデンティティです。長期認証情報を持つ IAM ユーザーの代わりに一時的な認証情報を使用することをお勧めします。詳細は「IAM ユーザーガイド」の「人間のユーザーが一時的な認証情報を使用して AWS にアクセスするには ID プロバイダーとのフェデレーションの使用が必要です」を参照してください。

IAM グループは、IAM ユーザーの集合を指定し、大量のユーザーに対するアクセス許可の管理を容易にします。詳細については、「IAM ユーザーガイド」の「IAM ユーザーに関するユースケース」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アイデンティであり、一時的な認証情報を提供します。ユーザーから IAM ロール (コンソール) に切り替える、または AWS CLI や AWS API オペレーションを呼び出すことで、ロールを引き受けることができます。詳細については、「IAM ユーザーガイド」の「ロールを引き受けるための各種方法」を参照してください。

IAM ロールは、フェデレーションユーザーアクセス、一時的な IAM ユーザーのアクセス許可、クロスアカウントアクセス、クロスサービスアクセス、および Amazon EC2 で実行するアプリケーションに役立ちます。詳細については、IAM ユーザーガイド の IAM でのクロスアカウントリソースアクセス 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 でアクセスを制御するには、ポリシーを作成して AWS ID またはリソースにアタッチします。ポリシーは、アイデンティティやリソースとの関連付けに伴うアクセス許可を定義します。AWS は、プリンシパルがリクエストを行うときに、これらのポリシーを評価します。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、ポリシーを使用して、どのプリンシパルがどのリソースに対して、どのような条件でアクションを実行できるかを定義することで、誰が何にアクセスできるかを指定します。

デフォルトでは、ユーザーやロールにアクセス許可はありません。IAM 管理者は IAM ポリシーを作成してロールに追加し、このロールをユーザーが引き受けられるようにします。IAM ポリシーは、オペレーションの実行方法を問わず、アクセス許可を定義します。

アイデンティティベースのポリシー

アイデンティティベースのポリシーは、アイデンティティ (ユーザー、グループ、またはロール) 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、アイデンティティがどのリソースに対してどのような条件下でどのようなアクションを実行できるかを制御します。アイデンティティベースポリシーの作成方法については、IAM ユーザーガイド の カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する 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、インラインポリシー (単一の ID に直接埋め込む) または管理ポリシー (複数の ID にアタッチされたスタンドアロンポリシー) にすることができます。管理ポリシーと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いずれかを選択する」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、リソースに添付する JSON ポリシードキュメントです。例としては、IAM ロール信頼ポリシーや Amazon S3 バケットポリシーなどがあります。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。リソースベースのポリシーでは、プリンシパルを指定する必要があります。

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は IAM の AWSマネージドポリシーは使用できません。

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アクセス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。

他のポリシータイプ

AWS は、より一般的なポリシータイプで付与された最大数のアクセス許可を設定できる、追加のポリシータイプをサポートしています。

- アクセス許可の境界 – アイデンティティベースのポリシーで IAM エンティティに付与することのできるアクセス許可の数の上限を設定します。詳細については、「IAM ユーザーガイド」の「IAM エンティティのアクセス許可境界」を参照してください。

- サービスコントロールポリシー (SCP) - AWS Organizations 内の組織または組織単位の最大のアクセス許可を指定します。詳細については、「AWS Organizations ユーザーガイド」の「サービスコントロールポリシー」を参照してください。
- リソースコントロールポリシー (RCP) – は、アカウント内のリソースで利用できる最大数のアクセス許可を定義します。詳細については、「AWS Organizations ユーザーガイド」の「リソースコントロールポリシー (RCP)」を参照してください。
- セッションポリシー – ロールまたはフェデレーションユーザーの一時セッションを作成する際にパラメータとして渡される高度なポリシーです。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アクセス許可を理解するのがさらに難しくなります。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

- アクション
- リソース
- 条件キー
- Amazon VPC リソースベースのポリシー
- タグに基づいた承認
- IAM ロール

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 ポリシーエレメントのリファレンス」を参照してください。

アクション

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ようなリソースにどのような条件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` 要素に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が記述されます。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。例えば、`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、`ec2:CreateVpc` アクションへのアクセス許可を付与します。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。例えば、`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、次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確認するには、「Service Authorization Reference」の「Actions defined by Amazon EC2」を参照してください。

リソース

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。

Resource JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。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、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すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

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}
```

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。複数リソースを単一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 リソースのタイプとその ARN のリストを確認するには、「Service Authorization Reference」の「Resource types defined by Amazon EC2」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んなリソースにどんな条件でアクションを実行できるかということです。

Condition 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。イコールや未満などの条件演算子を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。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 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。Amazon VPC 条件キーのリストを確認するには、「Service Authorization Reference」の「Condition keys for Amazon EC2」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 でのクロスアカウントリソースアクセス」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、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「リソース作成時にタグ付けする許可の付与」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」を参照してください。

IAM ロール

IAM ロール は、特定の権限を持つ、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

サービスリンクロールは、AWS サービスが他のサービスのリソースにアクセスしてお客様の代わりにアクションを完了することを許可します。サービスリンクロールは 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

この機能により、ユーザーに代わってサービスがサービスロールを引き受けることが許可されます。この役割により、サービスがお客様に代わって他のサービスのリソースにアクセスし、アクションを完了することが許可されます。サービスロールはIAM アカウントに表示され、アカウントによって所有されます。つまり、IAM 管理者はこの役割の権限を変更できます。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、「the section called “CloudWatch Logs へのフローログ発行のための IAM ロール”」を参照してください。

Amazon VPC ポリシーの例

デフォルトでは、IAM 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。AWS マネジメントコンソール、AWS CLI、または 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、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 ロールに、そのポリシーをアタッチします。

これらサンプルの、JSON ポリシードキュメントを使用して、IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーの作成 (コンソール)」を参照してください。

内容

- ポリシーに関するベストプラクティス
- Amazon VPC コンソールを使用する
- パブリックサブネットを持つ VPC を作成する
- VPC リソースの変更と削除
- セキュリティグループの管理
- セキュリティグループルールの管理
- 特定のサブネット内にインスタンスを起動する
- 特定の VPC 内にインスタンスを起動する
- VPC とサブネットへのパブリックアクセスをブロックする
- その他の Amazon VPC ポリシーの例

ポリシーに関するベストプラクティス

ID ベースのポリシーは、ユーザーのアカウント内で誰かが Amazon VPC リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションでは、AWS アカウント に費用が発生する場合があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS マネージドポリシーの使用を開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードへのアクセス許可の付与を開始するには、多くの一般的なユースケースのためにアクセス許可を付与する AWS マネージドポリシーを使用します。これらは AWS アカウントで使用できます。ユースケースに固有の AWS カスタマー管理ポリシーを定義して、アクセス許可を絞り込むことをお勧めします。詳細については、IAM ユーザーガイド の AWS マネージドポリシー または ジョブ機能の AWS マネージドポリシー を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアク

ションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイド の IAM でのポリシーとアクセス許可 を参照してください。

- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。また、CloudFormation などの特定の AWS のサービス を介して使用する場合、条件を使用してサービスアクションへのアクセスを許可することもできます。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。
- IAM アクセスアナライザー を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM アクセスアナライザー は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイド の IAM Access Analyzer でポリシーを検証する を参照してください。
- 多要素認証 (MFA) を要求する – AWS アカウントで IAM ユーザーまたはルートユーザーを要求するシナリオがある場合は、セキュリティを強化するために MFA をオンにします。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイド の MFA を使用した安全な API アクセス を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイド の IAM でのセキュリティのベストプラクティス を参照してください。

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厳しく制限されたアイデンティティベースポリシーを作成すると、そのポリシーを添付したエンティティ (IAM ロール) に対して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ロールに付与しますが、リソースを作成、更新、削除することはできません。

JSON

```
{
```

```
    "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 のみを呼び出すロールには、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、ロールが実行する必要がある API オペレーションに一致するアクションのみへのアクセスが許可されます。

パブリックサブネットを持つ VPC を作成する

次の例では、ロールが 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。ロールは、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、ロールは、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。

JSON

```
{
   "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、ロールは、Amazon VPC コンソールで VPC を作成することもできます。

VPC リソースの変更と削除

ロール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。例えば、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ロールに許可します。また、このポリシーでは、ロールがタグ 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。ロールは、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。

JSON

```
{
    "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、ロールがセキュリティグループを管理することを許可します。1 番目のステートメントでは、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ロールに許可します。2 番目のステートメントでは、ロール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。3 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ロールに許可します。4 番目のステートメントでは、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することをロールに許可します。5 番目のステートメントは、VPC にセキュリティグループを作成することをロールに許可します。

> **Note**
>
> AWS CloudFormation サービスでは、このポリシーを使用して、必須タグを含むセキュリティグループを作成することはできません。タグを必要とする ec2:CreateSecurityGroup アクションの条件を削除すると、このポリシーが機能します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RevokeSecurityGroupIngress",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:UpdateSecurityGroupRuleDescriptionsEgress",
```

```
                "ec2:RevokeSecurityGroupEgress",
                "ec2:DeleteSecurityGroup",
                "ec2:ModifySecurityGroupRules",
                "ec2:UpdateSecurityGroupRuleDescriptionsIngress"
            ],
            "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"
                },
                "ForAnyValue:StringEquals": {
                    "aws:TagKeys": "Stack"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:security-group/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateSecurityGroup"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeVpcs",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": "*"
        },
```

```
        {
            "Effect": "Allow",
            "Action": "ec2:CreateSecurityGroup",
            "Resource": "arn:aws:ec2:*:*:vpc/*"
        }
    ]
}
```

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ロール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。

ロール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ロールに付与します。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アクセス許可をロールに付与します。これにより、ロール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。

JSON

```
{
    "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:us-east-1:123456789012:security-group/*",
            "Condition": {
                "ArnEquals": {
                    "ec2:Vpc": "arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-1234567890abcdef0"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSecurityGroupRules",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:ModifySecurityGroupRules"
            ],
            "Resource": "arn:aws:ec2:us-east-1:123456789012:security-group-rule/
*"
        }
    ]
}
```

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アクセス許可をロールに付与します。このポリシーは、サブネットの ARN およびセキュリティグループの ARN を指定することで許可を与えます。ロール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、ロールにそのアクセス許可が付与さ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、ロールには、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アクセス許可が必要です。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:us-east-1::image/ami-*",
                "arn:aws:ec2:us-east-1:123456789012:instance/*",
                "arn:aws:ec2:us-
east-1:123456789012:subnet/subnet-1234567890abcdef0",
                "arn:aws:ec2:us-east-1:123456789012:network-interface/*",
                "arn:aws:ec2:us-east-1:123456789012:volume/*",
                "arn:aws:ec2:us-east-1:123456789012:key-pair/*",
                "arn:aws:ec2:us-east-1:123456789012:security-
group/sg-0abcdef1234567890"
            ]
        }
    ]
}
```

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アクセス許可をロールに付与します。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アクセス許可をロールに付与します。

JSON

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

```
            "Resource": "arn:aws:ec2:us-east-1:123456789012:subnet/*",
            "Condition": {
                "ArnEquals": {
                    "ec2:Vpc": "arn:aws:ec2:us-
east-1:123456789012:vpc/vpc-1234567890abcdef0"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:us-east-1::image/ami-*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/department": "dev"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:us-east-1:123456789012:instance/*",
                "arn:aws:ec2:us-east-1:123456789012:volume/*",
                "arn:aws:ec2:us-east-1:123456789012:network-interface/*",
                "arn:aws:ec2:us-east-1:123456789012:key-pair/*",
                "arn:aws:ec2:us-east-1:123456789012:security-group/*"
            ]
        }
    ]
}
```

VPC とサブネットへのパブリックアクセスをブロックする

次のポリシー例は、VPC ブロックパブリックアクセス (BPA) 機能を使用して VPC およびサブネット内のリソースに対するパブリックアクセスをブロックするための許可をロールに付与します。

例 1 - VPC BPA アカウント全体の設定と VPC BPA の除外に対する読み取り専用アクセスを許可します。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VPCBPAReadOnlyAccess",
      "Action": [
        "ec2:DescribeVpcBlockPublicAccessOptions",
        "ec2:DescribeVpcBlockPublicAccessExclusions"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

例 2 - VPC BPA アカウント全体の設定と VPC BPA の除外に対する読み取りおよび書き込みのフルアクセスを許可します。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VPCBPAFullAccess",
      "Action": [
        "ec2:DescribeVpcBlockPublicAccessOptions",
        "ec2:DescribeVpcBlockPublicAccessExclusions",
        "ec2:ModifyVpcBlockPublicAccessOptions",
        "ec2:CreateVpcBlockPublicAccessExclusion",
        "ec2:ModifyVpcBlockPublicAccessExclusion",
        "ec2:DeleteVpcBlockPublicAccessExclusion"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

例 3 - VPC BPA の設定の変更と除外の作成を除くすべての EC2 API に対するアクセスを許可します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EC2FullAccess",
            "Action": [
                "ec2:*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Sid": "VPCBPAPartialAccess",
            "Action": [
                "ec2:ModifyVpcBlockPublicAccessOptions",
                "ec2:CreateVpcBlockPublicAccessExclusion"
            ],
            "Effect": "Deny",
            "Resource": "*"
        }
    ]
}
```

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。

- マネージドプレフィックスリスト
- トラフィックのミラーリング
- Transit Gateway
- VPC エンドポイントおよび VPC エンドポイントサービス (AWS PrivateLink)
- VPC ピアリング接続

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
- iam:PassRole を実行する権限がありません
- 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

Amazon VPC でアクションを実行する権限がない

AWS マネジメントコンソール 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は、管理者に問い合わせてサポートを依頼する必要があります。サインイン認証情報を提供した担当者が管理者です。

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 IAM ロールに属していた場合に発生します。

```
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 を実行する権限がありません

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は、ポリシーを更新して Amazon VPC にロールを渡せるようにする必要があります。

一部の AWS のサービス では、新しいサービスロールやサービスリンク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、このアクションをサービスが実行するには、サービスロールから付与されたアクセス許可が必要です。Mary 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、Mary のポリシーを更新してメアリーに `iam:PassRole` アクションの実行を許可する必要があります。

サポートが必要な場合は、AWS 管理者に問い合わせてください。サインイン認証情報を提供した担当者が管理者です。

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。ロールの引き受けを委託するユーザーを指定できます。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」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 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「サードパーティが所有する AWS 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におけるロールとリソースベースのポリシーの使用方法の違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。

AWSAmazon Virtual Private Cloud の 管理ポリシー

AWS マネージドポリシーは、AWS が作成および管理するスタンドアロンポリシーです。AWS マネージドポリシーは、多くの一般的なユースケースに対してアクセス許可を提供するように設計されているため、ユーザー、グループ、ロールへのアクセス許可の割り当てを開始できます。

AWS マネージドポリシーは、ご利用の特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることにご注意ください。これは、すべての AWS ユーザーが使用できるようになるのを避けるためです。ユースケースに固有のカスタマー管理ポリシーを定義して、アクセス許可を絞り込むことをお勧めします。

AWS マネージドポリシーで定義されたアクセス許可は変更できません。AWS が AWS マネージドポリシーに定義されているアクセス許可を更新すると、更新はポリシーがアタッチされているすべてのプリンシパルアイデンティティ (ユーザー、グループ、ロール) に影響します。新しい AWS のサービス を起動するか、既存のサービスで新しい API オペレーションが使用可能になると、AWS が AWS マネージドポリシーを更新する可能性が最も高くなります。

詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」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。

このポリシーの許可を確認するには、「AWS マネージドポリシーリファレンス」の「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。

このポリシーの許可を確認するには、「AWS マネージドポリシーリファレンス」の「AmazonVPCReadOnlyAccess」を参照してください。

AWS 管理ポリシー: AmazonVPCCrossAccountNetworkInterfaceOperations

`AmazonVPCCrossAccountNetworkInterfaceOperations` ポリシーを IAM IDにアタッチできます。このポリシーは、ID がネットワークインターフェイスを作成し、クロスアカウントリソースにアタッチするためのアクセス許可を付与します。

このポリシーの許可を確認するには、「AWS マネージドポリシーリファレンス」の「AmazonVPCCrossAccountNetworkInterfaceOperations」を参照してください。

AWS マネージドポリシー: AWSServiceRoleForNATGateway

AWSServiceRoleForNATGateway ポリシーを IAM IDにアタッチできます。このポリシーにより、ユーザーに代わってリージョン NAT ゲートウェイを自動的にスケールする ID を許可するアクセス許可を付与します。

このポリシーのアクセス許可を確認するには、「AWS マネージドポリシーリファレンス」の「AWSServiceRoleForNATGateway」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 “AmazonVPCFullAccess” - 既存ポリシーへの更新	AWSIPAMServiceRolePolicy マネージドポリシー (ec2:ModifyManagedPrefixList、ec2:DescribeManagedPrefixLists、および ec2:GetManagedPrefixListEntries) に追加されたアクションで、IPAM がマネージドプレフィックスリストを変更および読み取りできるようになりました。	2025 年 10 月 31 日
the section called “AWSServiceRoleForNATGateway ” - 新しいポリシー	新しい AWSServiceRoleForNATGateway ポリシーにより、リージョン NAT ゲートウェイを自動的にスケールする ID を許可します。	2025 年 11 月 19 日
the section called “AmazonVPCFullAccess” – 既存ポリシーへの更新	VPC とセキュリティグループの関連付けの関連付け実行、関連付けの解除、および表示を可能にする AssociateSecurityGroupVpc、D	2024 年 12 月 9 日

変更	説明	日付
	escribeSecurityGroupVpcAssociations、および DisassociateSecurityGroupVpc アクションが追加されました。	
the section called “AmazonVPCReadOnlyAccess” – 既存ポリシーへの更新	VPC とセキュリティグループの関連付けを表示できる DescribeSecurityGroupVpcAssociations アクションが追加されました。	2024 年 12 月 9 日
the section called “AmazonVPCFullAccess” – 既存ポリシーへの更新	VPC で使用可能なセキュリティグループを取得できる GetSecurityGroupsForVpc アクションを追加しました。	2024 年 2 月 8 日
the section called “AmazonVPCReadOnlyAccess” – 既存ポリシーへの更新	VPC で使用可能なセキュリティグループを取得できる GetSecurityGroupsForVpc アクションを追加しました。	2024 年 2 月 8 日
the section called “AmazonVPCCrossAccountNetworkInterfaceOperations” – 既存ポリシーへの更新	ネットワークインターフェイスに関連付けられた IPv6 アドレスを管理できる AssignIpv6Addresses アクション および UnassignIpv6Addresses アクションが追加されました。	2023 年 9 月 25 日
the section called “AmazonVPCReadOnlyAccess” – 既存ポリシーへの更新	セキュリティグループルールを表示できる DescribeSecurityGroupRules アクションが追加されました。	2021 年 8 月 2 日

変更	説明	日付
the section called “AmazonVPCFullAccess” – 既存ポリシーへの更新	セキュリティグループルールを表示、変更できる DescribeSecurityGroupRules および ModifySecurityGroupRules アクションが追加されました。	2021 年 8 月 2 日
the section called “AmazonVPCFullAccess” – 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
the section called “AmazonVPCReadOnlyAccess” – 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 のサービスにリンクされたロールの使用

Amazon VPC は、AWS Identity and Access Management (IAM) のサービスにリンクされたロールを使用します。サービスリンクロールは、VPC に直接リンクされた一意のタイプの IAM ロールです。サービスにリンクされたロールは、VPC が事前定義しており、このサービスがお客様の代わりにその他の AWS サービスを呼び出すのに必要なアクセス許可がすべて含まれています。

サービスにリンクされたロールを使用することで、必要なアクセス許可を手動で追加する必要がなくなるため、VPC の設定が簡単になります。VPC は、サービスにリンクされたロールのアクセス許可を定義します。特に定義されている場合を除き、VPC のみがそのロールを引き受けることができます。定義される許可は信頼ポリシーと許可ポリシーに含まれており、その許可ポリシーを他の IAM エンティティにアタッチすることはできません。

サービスリンクロールを削除するには、最初に関連リソースを削除する必要があります。これにより、リソースにアクセスするための許可を意図せず削除することが防止され、VPC リソースが保護されます。

サービスにリンクされたロールをサポートする他のサービスについては、「IAM と連携する AWS サービス」を参照して、[サービスにリンクされたロール] 列が [はい] になっているサービスを探してください。サービスリンクロールに関するドキュメントをサービスで表示するには、リンクで [はい] を選択します。

VPC のサービスにリンクされたロールのアクセス許可

VPC は、AWSServiceRoleForNATGateway という名前のサービスにリンクされたロールを使用します。このサービスにリンクされたロールにより、Amazon VPC はユーザーに代わって Elastic IP アドレスを割り当ててリージョン NAT ゲートウェイを自動的にスケールし、リクエストに応じて既存の Elastic IP をリージョン NAT ゲートウェイに関連付け/関連付け解除したり、ネットワークインターフェイスを記述して既存のインフラストラクチャを識別し、新しいアベイラビリティーゾーンに自動的に拡張したりできます。

AWSServiceRoleForNATGateway サービスリンクロールは、次のサービスを信頼してロールを引き受けます。

- `ec2-nat-gateway.amazonaws.com`

AWSNATGatewayServiceRolePolicy という名前のロール許可ポリシーは、指定されたリソースで VPC が以下のアクションを完了できるようにします。

- アクション: サービスマネージド EIP の `AllocateAddress` を使用して、ユーザーに代わり EIP を割り当てます。サービスマネージド EIP により、サービスマネージドタグと ReleaseAddress による後続のタグ付けが自動的に処理されます。
- アクション: 既存の Elastic IP アドレスで `AssociateAddress` を使用して、リクエストに応じリージョン NAT ゲートウェイに手動で関連付けます。
- アクション: 既存の Elastic IP アドレスで `DisassociateAddress` を使用して、リクエストに応じリージョン NAT ゲートウェイから削除します。
- アクション: `DescribeAddresses` を使用して、関連付けで顧客が指定した EIP からパブリック IP アドレス情報を取得します。
- アクション: 既存のネットワークインターフェイスで、`DescribeNetworkInterface` を使用してインフラストラクチャが存在するアベイラビリティーゾーンを自動的に識別し、新しいゾーンに自動的にスケールアウトします。

ユーザー、グループ、またはロールにサービスリンクロールの作成、編集、または削除を許可するには、アクセス許可を設定する必要があります。詳細についてはIAM ユーザーガイド の「サービスにリンクされた役割のアクセス許可」を参照してください。

VPC のサービスにリンクされたロールの作成

サービスリンクロールを手動で作成する必要はありません。AWS マネジメントコンソール、AWS CLI、または AWS API で「リージョン」アベイラビリティーモードを使用して NAT ゲートウェイを作成すると、VPC によってサービスにリンクされたロールが作成されます。

> **Important**
>
> このサービスリンクロールはこのロールでサポートされている機能を使用する別のサービスでアクションが完了した場合にアカウントに表示されます。また、2017 年 1 月 1 日以前に VPC サービスを使用していた場合、サービスにリンクされたロールのサポートが開始された時点で、VPC が [AWSServiceRoleForNATGateway] ロールをアカウントに作成済みです。詳細については、「AWS アカウントに新しいロールが表示される」を参照してください。

このサービスリンクロールを削除した後で再度作成する必要が生じた場合は同じ方法でアカウントにロールを再作成できます。「リージョン」アベイラビリティーモードで NAT ゲートウェイを作成すると、VPC によってサービスにリンクされたロールが再度作成されます。

また、AWSServiceRoleForNATGateway ユースケースでサービスにリンクされたロールを作成するには、IAM コンソールを使用できます。AWS CLI または AWS API では、`ec2-nat-gateway.amazonaws.com` サービス名を使用してサービスリンクロールを作成します。詳細については、IAM ユーザーガイドの「サービスリンクロールの作成」を参照してください。このサービスリンクロールを削除しても、同じ方法でロールを再作成できます。

VPC のサービスにリンクされたロールの編集

VPC では、AWSServiceRoleForNATGateway のサービスにリンクされたロールを編集することはできません。サービスリンクロールの作成後は、さまざまなエンティティがロールを参照する可能性があるため、ロール名を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「サービスリンクロールの編集」を参照してください。

VPC のサービスにリンクされたロールの削除

サービスリンクロールを必要とする機能やサービスが不要になった場合は、ロールを削除することをお勧めします。そうすることで、使用していないエンティティがアクティブにモニタリングされたり、メンテナンスされたりすることがなくなります。ただし、手動で削除する前に、サービスリンクロールのリソースをクリーンアップする必要があります。

> **Note**
>
> リソースを削除する際に、VPC のサービスでロールが使用されていると、削除に失敗することがあります。失敗した場合は数分待ってから操作を再試行してください。

AWSServiceRoleForNATGateway が使用する VPC リソースを削除するには

- デプロイされたすべてのリージョンで、すべてのリージョン NAT ゲートウェイを削除します。

サービスリンクロールを IAM で手動削除するには

IAM コンソール、AWS CLI、または AWS API を使用して、AWSServiceRoleForNATGateway サービスリンクロールを削除します。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの削除」を参照してください。

VPC のサービスにリンクされたロールをサポートするリージョン

VPC は、サービスが利用可能なすべてのリージョンで、サービスにリンクされたロールの使用をサポートしています。詳細については、「AWS リージョンとエンドポイント」を参照してください。

VPC は、サービスを利用できるすべてのリージョンで、サービスにリンクされたロールの使用をサポートしているわけではありません。AWSServiceRoleForNATGateway ロールは、以下のリージョンで使用できます。

リージョン名	リージョン識別子	VPC でのサポート
米国東部 (バージニア北部)	us-east-1	はい
米国東部 (オハイオ)	us-east-2	はい
米国西部 (北カリフォルニア)	us-west-1	はい

リージョン名	リージョン識別子	VPC でのサポート
米国西部 (オレゴン)	us-west-2	はい
アフリカ (ケープタウン)	af-south-1	はい
アジアパシフィック (香港)	ap-east-1	はい
アジアパシフィック (台北)	ap-east-2	はい
アジアパシフィック (ジャカルタ)	ap-southeast-3	はい
アジアパシフィック (ムンバイ)	ap-south-1	はい
アジアパシフィック (ハイデラバード)	ap-south-2	はい
アジアパシフィック (大阪)	ap-northeast-3	はい
アジアパシフィック (ソウル)	ap-northeast-2	はい
アジアパシフィック (シンガポール)	ap-southeast-1	はい
アジアパシフィック (シドニー)	ap-southeast-2	はい
アジアパシフィック (東京)	ap-northeast-1	はい
アジアパシフィック (メルボルン)	ap-southeast-4	はい
アジアパシフィック (マレーシア)	ap-southeast-5	はい
アジアパシフィック (ニュージーランド)	ap-southeast-6	はい
アジアパシフィック (タイ)	ap-southeast-7	はい
カナダ (中部)	ca-central-1	はい
カナダ西部 (カルガリー)	ca-west-1	はい
欧州 (フランクフルト)	eu-central-1	はい
欧州 (チューリッヒ)	eu-central-2	はい

リージョン名	リージョン識別子	VPC でのサポート
欧州 (アイルランド)	eu-west-1	はい
欧州 (ロンドン)	eu-west-2	はい
欧州 (ミラノ)	eu-south-1	はい
欧州 (スペイン)	eu-south-2	はい
欧州 (パリ)	eu-west-3	はい
欧州 (ストックホルム)	eu-north-1	はい
イスラエル (テルアビブ)	il-central-1	はい
中東 (バーレーン)	me-south-1	はい
中東 (アラブ首長国連邦)	me-central-1	はい
中東 (サウジアラビア)	me-west-1	はい
メキシコ (中部)	mx-central-1	はい
南米 (サンパウロ)	sa-east-1	はい
AWS GovCloud (US-East)	us-gov-east-1	いいえ
AWS GovCloud (米国西部)	us-gov-west-1	いいえ

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 Virtual Private Cloud は、AWS グローバルネットワークセキュリティによって保護されています。AWSセキュリティサービスと AWS がインフラストラクチャを保護する方法については、AWS クラウドセキュリティ を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、セキュリティの柱 - AWS Well-Architected Framework の インフラストラクチャ保護 を参照してください。

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。クライアントは次をサポートする必要があります。

- Transport Layer Security (TLS)。TLS 1.2 が必須で、TLS 1.3 をお勧めします。
- DHE (楕円ディフィー・ヘルマン鍵共有) や ECDHE (楕円曲線ディフィー・ヘルマン鍵共有) などの完全前方秘匿性 (PFS) による暗号スイート。これらのモードは Java 7 以降など、ほとんどの最新システムでサポートされています。

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。

AWS PrivateLink を使用して、VPC 内のリソースが、サービスが VPC で直接ホストされているかのようにプライベート IP アドレスを使用して AWS のサービス に接続することを許可します。したがって、AWS のサービス へのアクセスにインターネットゲートウェイまたは NAT デバイスを使用する必要はありません。

ネットワークトラフィックの制御

EC2 インスタンスなど、VPC 内のリソースへのネットワークトラフィックを制御するには、以下のオプションを検討してください。

- VPC へのネットワークアクセスを制御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使用します。必要に応じて、ネットワーク ACL 使用すると、ステートレスできめの粗いネットワーク制御を行うことができます。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ー処理を実行して、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。ネットワーク ACL は、セカンダリ制御 (特定のトラフィックのサブセットを拒否するなど) または高レベルのサブネットガードレールとして効果的に使用できます。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、万が一正しいセキュリティグループがない状態でインスタンスが起動された場合に、深層防御として活用できます。
- 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は、踏み台ホストまたは NAT ゲートウェイを使用します。

- 接続要件を満たす最小限のネットワークルートでサブネットルートテーブルを設定します。
- 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、変更管理のための特別な 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への変更を容易に監査することができます。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使用したりするなど、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 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します。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監視します。
- AWS Security Hub CSPM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。
- AWS Network Firewall を使用して、VPC 内のサブネットを一般的なネットワークの脅威から保護します。

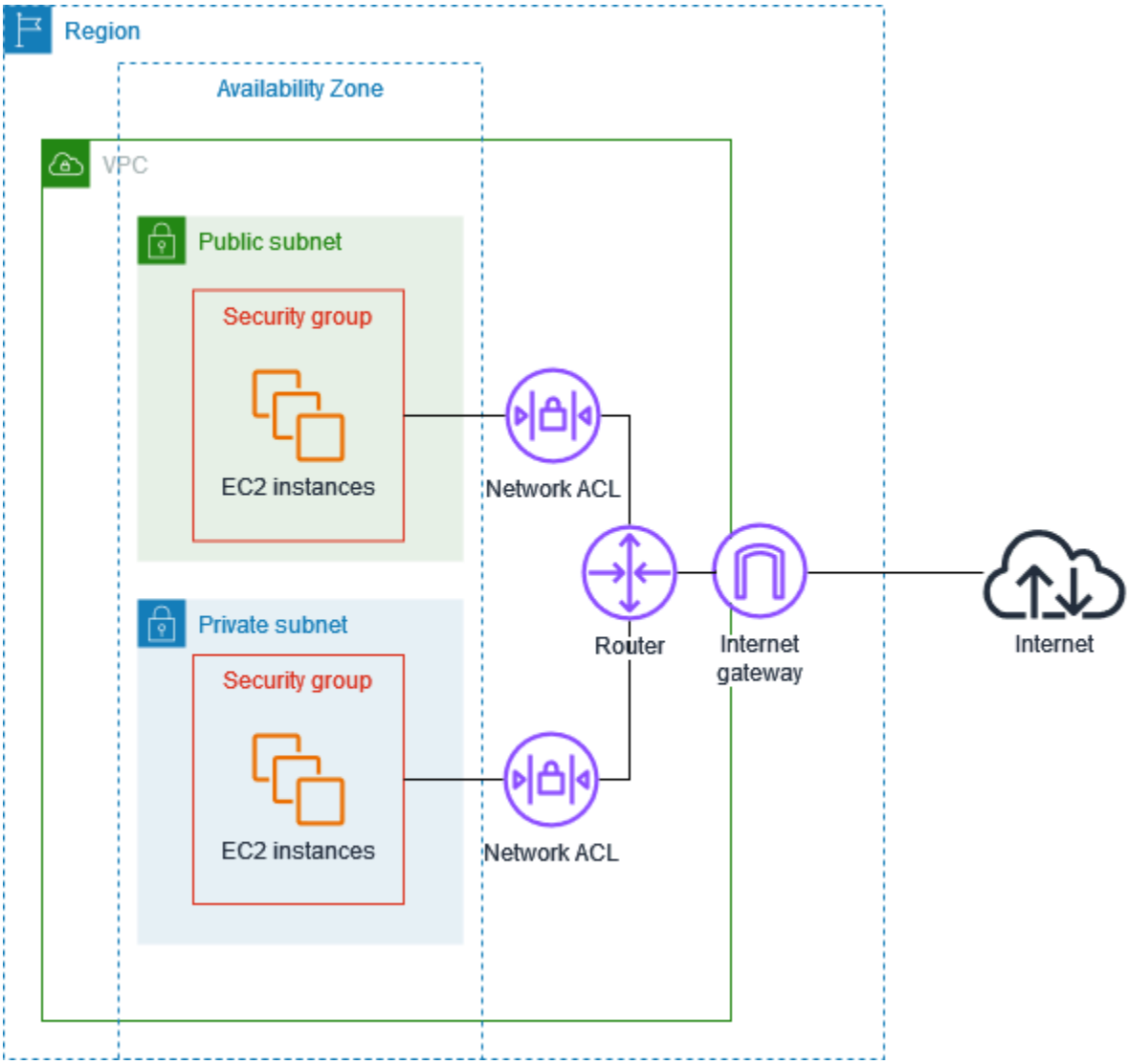
セキュリティグループとネットワーク ACL を比較する

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。

特性	セキュリティグループ	ネットワーク ACL
オペレーションのレベル	インスタンスレベル	サブネットレベル
範囲	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適用されます	関連付けられたサブネット内のすべてのインスタンスに適用されます
ルールタイプ	ルールのみを許可します	許可ルールと拒否ルール
ルールの評価	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	トラフィックの一致が見つかるまで昇順でルールを評価します

特性	セキュリティグループ	ネットワーク ACL
返されるトラフィック	自動許可 (ステートフル)	明示的に許可する必要があります (ステートレス)

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。



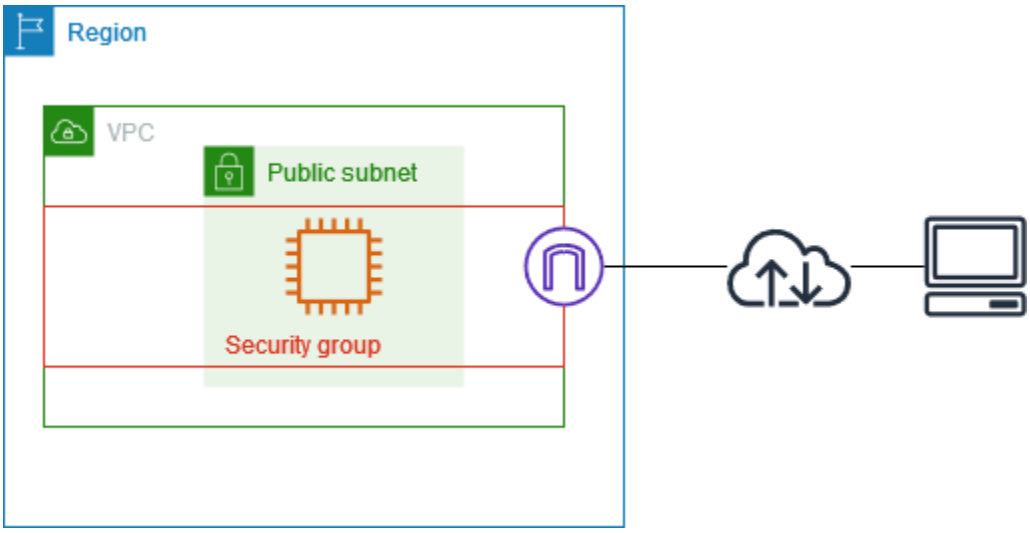
セキュリティグループのみを使用してインスタンスを保護できます。ただし、ネットワーク ACL は追加の防御レイヤーとして追加できます。詳細については、「例: サブネットのインスタンスへのアクセス制御」を参照してください。

セキュリティグループを使用して AWS リソースへのトラフィックを制御する

セキュリティグループは、関連付けられたリソースに到達できるトラフィックおよびリソースから発信できるトラフィックを制御します。例えば、セキュリティグループを EC2 インスタンスに関連付けると、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が制御されます。

VPC を作成すると、デフォルトのセキュリティグループが使用されます。VPC ごとに追加のセキュリティグループを作成し、それぞれに独自のインバウンドルールとアウトバウンドルールを設定できます。インバウンドルールごとに、送信元、ポート範囲、プロトコルを指定できます。アウトバウンドルールごとに、送信先、ポート範囲、プロトコルを指定できます。

次の図は、サブネット、インターネットゲートウェイ、セキュリティグループを備えた VPC を示しています。サブネットには EC2 インスタンスが含まれています。セキュリティグループは、インスタンスに割り当てられます。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ます。インスタンスに到達するトラフィックは、セキュリティグループのルールで許可されているトラフィックだけです。例えば、ネットワークからインスタンスへの ICMP トラフィックを許可するルールがセキュリティグループに含まれている場合は、お使いのコンピュータからインスタンスに ping を送信できます。SSH トラフィックを許可するルールがセキュリティグループに含まれていない場合、SSH を使用してインスタンスに接続することはできません。



内容

- セキュリティグループの基本
- セキュリティグループの例
- セキュリティグループのルール
- VPC のデフォルトセキュリティグループ
- VPC 用の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ルールを設定する
- 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を複数の VPCに関連付ける
- AWS Organizations とセキュリティグループを共有する

料金

セキュリティグループは追加料金なしで使用できます。

セキュリティグループの基本

- セキュリティグループ VPC 関連付け機能を使用してセキュリティグループを同じリージョン内の他の VPC に関連付ける場合は、セキュリティグループをセキュリティグループと同じ VPC 内に作成されたリソースに割り当てるか、他の VPC リソースに割り当てることができます。1 つのリソースに複数のセキュリティ グループを割り当てることもできます。
- 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。
  - 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名では大文字と小文字が区別されません。
  - 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、sg- で開始できません。
- セキュリティグループはステートフルです。例えば、インスタンスからリクエストを送信した場合、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関係なく、インスタンスに到達が許可されます。許可されたインバウンドトラフィックへのレスポンスは、アウトバウンドルールに関係なく、インスタンスを離れることができます。

- セキュリティグループでは、以下で送受信されるトラフィックはフィルターされません。
  - 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 ECS タスクメタデータエンドポイント
  - Windows インスタンスのライセンスアクティベーション
  - Amazon Time Sync Service
  - デフォルトの VPC ルーターによる予約済み IP アドレス
- 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」を参照してください。

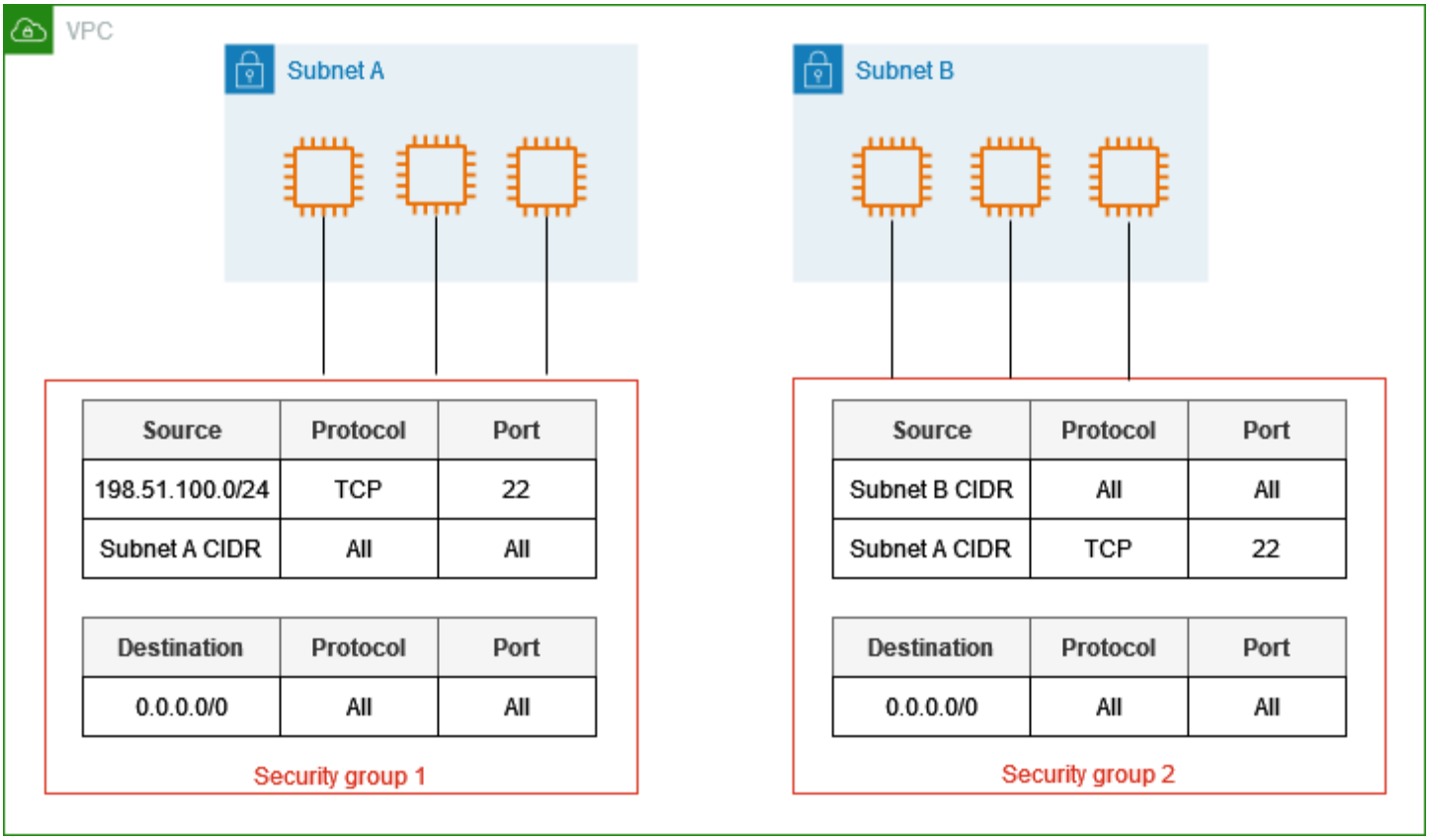
ベストプラクティス

- 特定の IAM プリンシパルのみにセキュリティグループの作成と変更を許可します。
- エラーのリスクを減らすために、必要最小限の数のセキュリティグループを作成してください。各セキュリティグループを使用して、同様の機能とセキュリティ要件を持つリソースへのアクセスを管理します。
- EC2 インスタンスにアクセスできるようにするために、ポート 22 (SSH) または 3389 (RDP) のインバウンドルールを追加する場合は、特定の IP アドレスの範囲のみを許可する必要があります。0.0.0.0/0 (IPv4) と ::/ (IPv6) を指定すると、指定したプロトコルを使用して、誰でも任意の IP アドレスからインスタンスにアクセスできるようになります。
- 広い範囲のポートを開かないでください。各ポートからのアクセスが、それを必要とする送信元または宛先に制限されていることを確認します。
- 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 を作成することを検討してください。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」を参照してください。

セキュリティグループの例

次の図は、2 つのセキュリティグループと 2 つのサブネットを持つ VPC を示しています。サブネット A のインスタンスは、接続要件が同じであるため、セキュリティグループ 1 に関連付けられます。サブネット B のインスタンスは、接続要件が同じであるため、セキュリティグループ 2 に関連付けられます。セキュリティグループのルールでは、トラフィックを次のように許可します。

- セキュリティグループ 1 の最初のインバウンドルールは、指定されたアドレス範囲 (例えば、独自のネットワーク内の範囲) からサブネット A のインスタンスへの SSH トラフィックを許可します。
- セキュリティグループ 1 の 2 番目のインバウンドルールは、サブネット A のインスタンスが任意のプロトコルとポートを使用して相互に通信することを許可します。
- セキュリティグループ 2 の最初のインバウンドルールは、サブネット B のインスタンスが任意のプロトコルとポートを使用して相互に通信することを許可します。
- セキュリティグループ 2 の 2 番目のインバウンドルールは、サブネット A のインスタンスが SSH を使用してサブネット B のインスタンスと通信することを許可します。
- どちらのセキュリティグループも、すべてのトラフィックを許可するデフォルトのアウトバウンドルールを使用します。



セキュリティグループのルール

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リソー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。特定のソースまたは送信先へのアクセス権を付与できます。

内容

- セキュリティグループのルールの基本
- セキュリティグループルールの構成要素
- セキュリティグループの参照
- セキュリティグループのサイズ
- 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のルールの基本

セキュリティグループのルール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。
- 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、インバウンドトラフィックは許可されません。
- セキュリティグループを最初に作成するとき、リソースからの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設定されます。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、アウトバウンドトラフィックは許可されません。
- 複数のセキュリティグループをリソースに関連付けると、各セキュリティグループのルールが集約されて、アクセス許可の判断に使用する 1 つのルールセットが形成されます。
- 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リソースにこの変更が自動的に適用されます。手順については、「セキュリティグループのルールを設定する」を参照してください。
- 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。詳細については、「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。

制限

セキュリティグループは、「VPC+2 IP アドレス」(「Amazon Route 53 デベロッパーガイド」の「Amazon Route 53 Resolver」を参照) または AmazonProvidedDNS と呼ばれることがある Route 53 Resolver から送受信される DNS リクエストをブロックできません。Route 53 Resolver 経由の DNS リクエストをフィルタリングするには、Route 53 Resolver DNS Firewall を使用します。

セキュリティグループルールの構成要素

以下は、インバウンドおよびアウトバウンドセキュリティグループルールの構成要素です。

- 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。例えば、ICMP エコー要求にはタイプ 8、ICMPv6 エコー要求にはタイプ 128 を使用します。詳細については、「Amazon EC2 ユーザーガイド」の「Ping/ICMP のルール」を参照してください。
- Source or destination (送信元または送信先): 許可する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)。次のいずれかを指定します。
  - 単一の IPv4 アドレス。/32 プレフィクス長を使用する必要があります。例えば、203.0.113.1/32。
  - 単一の IPv6 アドレス。/128 プレフィクス長を使用する必要があります。例えば、2001:db8:1234:1a00::123/128。
  - CIDR ブロック表記の IPv4 アドレスの範囲。例えば、203.0.113.0/24。
  - CIDR ブロック表記の IPv6 アドレスの範囲。例えば、2001:db8:1234:1a00::/64。
  - プレフィクスリストの ID。例えば、pl-1234abc1234abc123 です。詳細については、「マネージドプレフィックスリスト」を参照してください。
  - セキュリティグループの ID。例えば、sg-1234567890abcdef0 です。詳細については、「the section called “セキュリティグループの参照”」を参照してください。
- (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。

例については、「Amazon EC2 ユーザーガイド」の「さまざまなユースケースのセキュリティグループのルール」を参照してください。

セキュリティグループの参照

ルールのソースまたは宛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。インスタンスは、指定されたプロトコルとポート経由で、インスタンスのプライベート IP アドレスを使用して、指定された方向の通信を行うことができます。

例えば、以下は、セキュリティグループ sg-0abcdef1234567890 を参照するセキュリティグループのインバウンドルールを表しています。このルールは、sg-0abcdef1234567890 に関連付けられたインスタンスからのインバウンド SSH トラフィックを許可します。

ソース	プロトコル	ポート範囲
sg-0abcdef1234567890	TCP	22

セキュリティグループルール内のセキュリティグループを参照するときは、以下の点に注意してください。

- 次のいずれかに該当する場合、別のセキュリティグループのインバウンドルールのセキュリティグループを参照できます:
  - セキュリティグループは同じ VPC に関連付けられます。
  - セキュリティグループが関連付けられている VPC 間にピアリング接続があります。
  - セキュリティグループが関連付けられている VPC 間にトランジットゲートウェイがあります。
- 次のいずれかに該当する場合、アウトバウンドルールのセキュリティグループを参照できます:
  - セキュリティグループは同じ VPC に関連付けられます。
  - セキュリティグループが関連付けられている VPC 間にピアリング接続があります。
- 参照されるセキュリティグループからのルールが、このグループを参照するセキュリティグループに追加されていない。
- インバウンドルールの場合は、セキュリティグループに関連付けられた EC2 インスタンスが、参照されるセキュリティグループに関連付けられた EC2 インスタンスのネットワークインターフェイスからのプライベート IP アドレスからのインバウンドトラフィックを受信できる。
- アウトバウンドルールの場合は、セキュリティグループに関連付けられた EC2 インスタンスが、参照されるセキュリティグループに関連付けられた EC2 インスタンスのネットワークインターフェイスからのプライベート IP アドレスへのアウトバウンドトラフィックを送信できる。
- `AuthorizeSecurityGroupIngress`、`AuthorizeSecurityGroupEgress`、`RevokeSecurityGrou` および `RevokeSecurityGroupEgress` の各アクションでは、参照されるセキュリティグループに対しては認可されません セキュリティグループが存在するかどうかが確認されるだけです。この結果は以下のようになります。
  - IAM ポリシーで、これらのアクションについて参照されるセキュリティグループを指定しても効果はありません。
  - 参照されるセキュリティグループが別のアカウントによって所有されている場合、所有者アカウントはこれらのアクションについて CloudTrail イベントを受信しません。

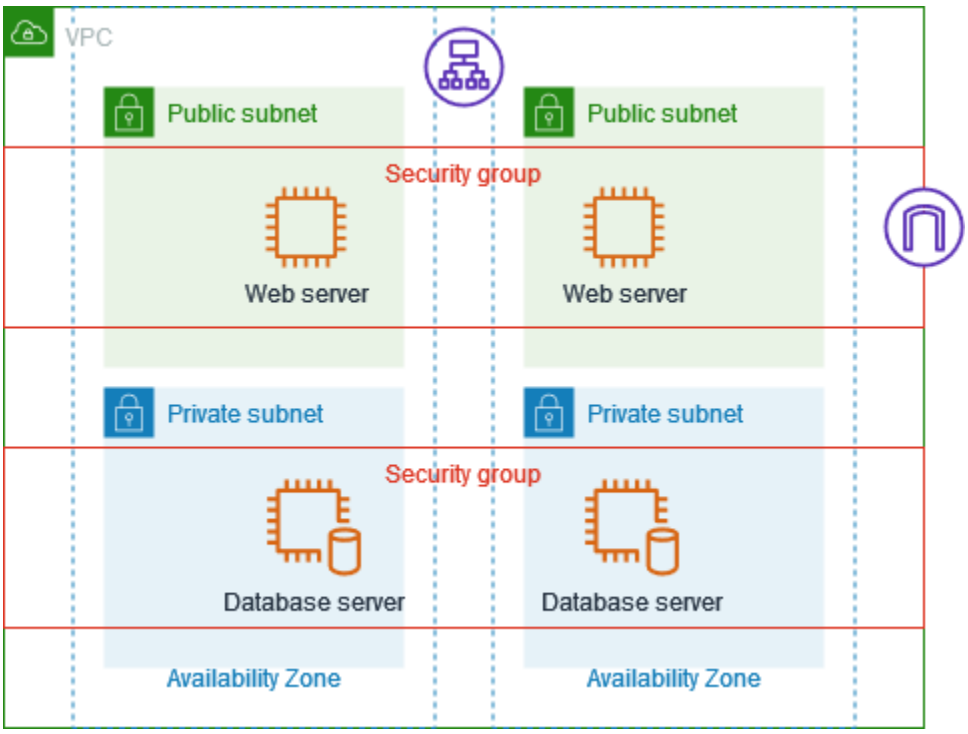
制限

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される必要があります。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。

例

次の図は、2 つのアベイラビリティーゾーン、1 つのインターネットゲートウェイ、1 つの Application Load Balancer のサブネットを使用する VPC を示しています。各アベイラビリティーゾーンには、ウェブサーバー用のパブリックサブネットと、データベースサーバー用のプライベートサブネットがあります。ロードバランサー、ウェブサーバー、データベースサーバーには個別のセキュリティグループがあります。以下のセキュリティグループルールを作成して、トラフィックを許可します。

- ロードバランサーのセキュリティグループにルールを追加して、インターネットからの HTTP および HTTPS トラフィックを許可します。送信元は 0.0.0.0/0 です。
- ウェブサーバーのセキュリティグループにルールを追加して、ロードバランサーからの HTTP および HTTPS トラフィックのみを許可します。送信元はロードバランサーのセキュリティグループです。
- データベースサーバーのセキュリティグループにルールを追加して、ウェブサーバーからのデータベースリクエストを許可します。送信元はウェブサーバーのセキュリティグループです。



セキュリティグループのサイズ

各ルールがセキュリティグループごとに設定できるルールの最大数にカウントされる方法は、ソースまたは送信先のタイプに応じて判断されます。

- CIDR ブロックを参照するルールは、1 個のルールとしてカウントされます。
- 別のセキュリティグループを参照するルールは、参照されるセキュリティグループのサイズにかかわらず、1 個のルールとしてカウントされます。
- カスタマーマネージドプレフィックスリストを参照するルールは、プレフィックスリストの最大サイズにならってカウントされます。例えば、プレフィックスリストの最大サイズが 20 の場合、このプレフィックスリストを参照するルールも 20 個のルール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を参照するルールは、プレフィックスリストのウェイトにならってカウントされます。例えば、プレフィックスリストの最大サイズが 10 の場合、このプレフィックスリストを参照するルールも 10 個のルールとしてカウントされます。詳細については、「the section called “使用可能な AWS マネージドプレフィックスリスト”」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、または別のアカウントで共有されている VPC を使用している場合、VPC のセキュリティグループルールは、そのピア VPC または共有 VPC

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リソースと、参照するセキュリティグループに関連付けられているリソースが、相互に通信できるようになります。詳細については、「Amazon VPC ピアリングガイド」の「セキュリティグループの更新とピアセキュリティグループの参照」を参照してください。

ピア VPC または共有 VPC 内のセキュリティグループを参照するセキュリティグループルールがあるとき、参照されているセキュリティグループが削除された場合、またはVPC ピアリング接続が削除された場合、そのルールは古くなったものとしてとマークされます。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。

VPC のデフォルトセキュリティグループ

デフォルトの VPC および作成した VPC には、デフォルトのセキュリティグループが適用されます。デフォルトセキュリティグループの名前は「default」です。

デフォルトのセキュリティグループを使用する代わりに、特定のリソース、またはリソースグループ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。ただし、作成時に何らかのリソースとセキュリティグループを関連付けない場合は、デフォルトのセキュリティグループが関連付けられます。例えば、EC2 インスタンス起動時にセキュリティグループを指定しない場合、インスタンスにはデフォルトの VPC 用セキュリティグループが関連付けられます。

デフォルトセキュリティグループの基本

- 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` のエラーが発生します。

デフォルトのルール

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インバウンドルールについて説明します。

ソース	プロトコル	ポート範囲	説明
sg-1234567890abcdef0	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すべてのリソースからのインバウンドトラフィックを許

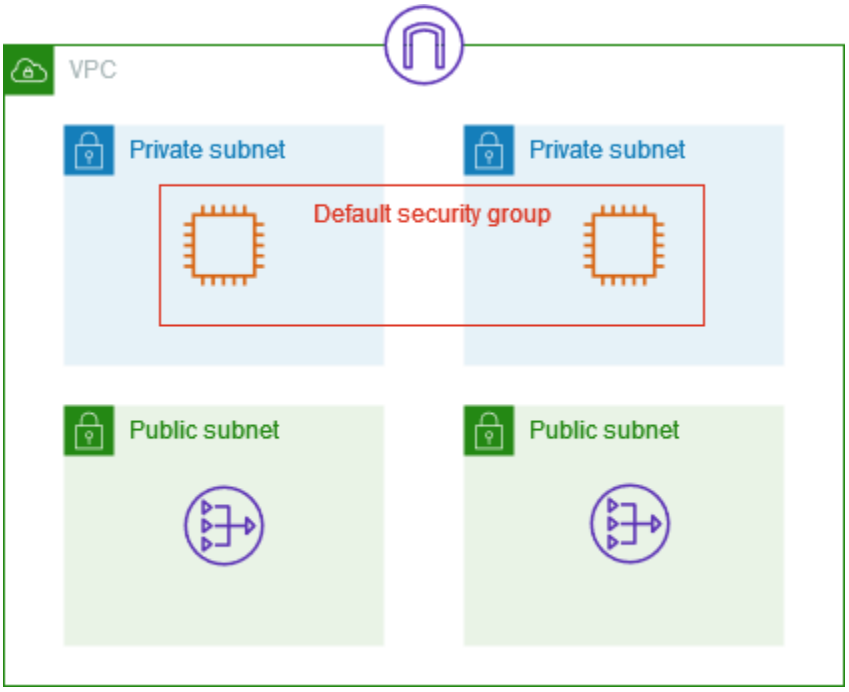
ソース	プロトコル	ポート範囲	説明
			可します。ソースは、このセキュリティグループの ID です。

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アウトバウンドルールについて説明します。

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	すべてのアウトバウンド IPv4 トラフィックを許可します。
::/0	すべて	すべて	すべてのアウトバウンド IPv6 トラフィックを許可します。このルールは、VPC に IPv6 CIDR ブロックが関連付けられている場合にのみ追加されます。

例

次の図は、デフォルトのセキュリティグループ、インターネットゲートウェイ、NAT ゲートウェイを備えた VPC を示しています。デフォルトのセキュリティにはデフォルトルールのみが含まれており、VPC で実行されている 2 つの EC2 インスタンスに関連付けられています。このシナリオでは、各インスタンスはすべてのポートとプロトコルで他のインスタンスからのインバウンドトラフィックを受信できます。デフォルトのルールでは、インスタンスはインターネットゲートウェイまたは NAT ゲートウェイからのトラフィックを受信できません。インスタンスが追加のトラフィックを受信する必要がある場合は、必要なルールを含むセキュリティグループを作成し、その新しいセキュリティグループをデフォルトのセキュリティグループではなくインスタンスに関連付けることをお勧めします。



VPC 用のセキュリティグループを作成するには

仮想プライベートクラウド (VPC) には、デフォルトのセキュリティグループが備わっています。追加のセキュリティグループを作成することができます。セキュリティグループは、作成時に対象とした VPC のリソースにのみ使用できます。

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リソースを離れ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。ルールは、セキュリティグループ作成時に、または後で追加することができます。詳細については、「セキュリティグループのルール」を参照してください。

必要な アクセス許可

作業を開始する前に、必要なアクセス許可があることを確認してください。詳細については次を参照してください:

- セキュリティグループの管理
- セキュリティグループルールの管理

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 [セキュリティグループ] をクリックします。
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は、その名前と説明を変更することはできません。
5. [VPC] では、セキュリティグループを関連付けるリソースを作成する VPC を選択します。
6. (任意) インバウンドルールを追加するには、[インバウンドルール] を選択します。ルールごとに、[ルールを追加] を選択し、プロトコル、ポート、および送信元を指定します。詳細については、「セキュリティグループのルールを設定する」を参照してください。
7. (任意) アウトバウンドルールを追加するには、[アウトバウンドルール] を選択します。ルールごとに、[ルールを追加] を選択し、プロトコル、ポート、および送信先を指定します。
8. (オプション) タグを追加するには、[Add new tag] (新しいタグを追加) を選択し、そのタグのキーと値を入力します。
9. [セキュリティグループの作成] を選択してください。

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。

または、既存のセキュリティグループをコピーすることで、新しいセキュリティグループを作成することができます。セキュリティグループをコピーすると、元のセキュリティグループと同じインバウンドルールとアウトバウンドルールが自動的に追加され、元のセキュリティグループと同じ VPC が使用されます。新しいセキュリティグループの名前と説明を入力できます。任意で別の VPC を選択でき、また必要に応じてインバウンドルールとアウトバウンドルールを変更できます。ただし、セキュリティグループはあるリージョンから別のリージョンにはコピーできません。

既存のセキュリティグループに基づい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、[セキュリティグループ] を選択します。
3. セキュリティグループを選択します。
4. [アクション]、[新しいセキュリティグループにコピー] の順に選択します。
5. セキュリティグループの名前と説明を入力します。
6. (任意) 必要に応じて別の VPC を選択します。
7. (任意) 必要に応じてセキュリティグループのルールを追加、削除、または編集します。
8. [セキュリティグループの作成] を選択してください。

セキュリティグループのルールを設定する

セキュリティグループを作成したら、そのセキュリティグループルールを追加、更新、削除できます。ルールを追加、更新、または削除すると、変更はそのセキュリティグループに関連付けられているリソースに自動的に適用されます。

必要なアクセス許可

作業を開始する前に、必要な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、「セキュリティグループルールの管理」を参照してください。

プロトコルとポート

- コンソールでは、事前定義されたタイプを選択すると、プロトコルとポート範囲が指定されます。ポート範囲を入力するには、[カスタム TCP] または [カスタム UDP] のいずれかのカスタムタイプを選択する必要があります。
- AWS CLI では、`--protocol` オプションおよび `--port` オプションを使用して、単一のポートを持つ単一のルールを追加できます。複数のルール、またはポート範囲を持つルールを追加するには、代わりに `--ip-permissions` オプションを使用します。

送信元と送信先

- コンソールでは、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、以下を指定できます。
  - [カスタム] – IPv4 CIDR ブロック、IPv6 CIDR ブロック、セキュリティグループ、またはプレフィックスリスト。
  - [Anywhere-IPv4] – 0.0.0.0/0 IPv4 CIDR ブロック。
  - [Anywhere-IPv6] – ::/0 IPv6 CIDR ブロック。
  - [マイ IP] – ローカルコンピュータのパブリック IPv4 アドレス。
- AWS CLI では、`--cidr` オプションを使用して IPv4 CIDR ブロックを指定するか、`--source-group` オプションを使用してセキュリティグループを指定できます。プレフィックスリストまたは IPv6 CIDR ブロックを指定するには、`--ip-permissions` オプションを使用します。

> **Warning**
>
> [Anywhere-IPv4] を選択すると、すべての IPv4 アドレスからのトラフィックが許可されます。[Anywhere-IPv6] を選択すると、すべての IPv6 アドレスからのトラフィックが許可され

ます。リソースへのアクセスが必要な特定の IP アドレス範囲のみを許可するのがベストプラクティスです。

コンソールを使用してセキュリティグループを設定するには

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 [セキュリティグループ] をクリックします。
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を編集するには、[アクション] または [インバウンドルール] タブから [インバウンドルールを編集] を選択します。

   a. ルールを追加するには、[ルールを追加] を選択し、ルールのタイプ、プロトコル、ポート、および送信元を入力します。

      タイプが 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。その他のタイプについては、プロトコルとポート範囲は自動的に設定されます。

   b. ルールを更新するには、必要に応じてプロトコル、説明、送信元を変更します。ただし、送信元のタイプを変更することはできません。例えば、送信元が IPv4 CIDR ブロックの場合、IPv6 CIDR ブロック、プレフィックスリスト、またはセキュリティグループを指定することはできません。

   c. ルールを削除するには、[削除] ボタンを選択します。
5. アウトバウンドルールを編集するには、[アクション] または [アウトバウンドルール] タブから [アウトバウンドルールを編集] を選択します。

   a. ルールを追加するには、[ルールを追加] を選択し、ルールのタイプ、プロトコル、ポート、および送信先を入力します。オプションとして説明を入力することもできます。

      タイプが 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。その他のタイプについては、プロトコルとポート範囲は自動的に設定されます。

   b. ルールを更新するには、必要に応じてプロトコル、説明、送信元を変更します。ただし、送信元のタイプを変更することはできません。例えば、送信元が IPv4 CIDR ブロックの場

合、IPv6 CIDR ブロック、プレフィックスリスト、またはセキュリティグループを指定することはできません。

c. ルールを削除するには、[削除] ボタンを選択します。

6. [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のルールを設定するには

- 追加 – 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。
- 削除 – revoke-security-group-ingress コマンドおよび revoke-security-group-egress コマンドを使用します。
- 変更 – modify-security-group-rules、update-security-group-rule-descriptions-ingress、および update-security-group-rule-descriptions-egress コマンドを使用します。

セキュリティグループを削除する

作成したセキュリティグループが用済みになったら、削除することができます。

要件

- このセキュリティグループをリソースに関連付けることはできません。
- 別のセキュリティグループのルールでこのセキュリティグループを参照することはできません。
- このセキュリティグループは、VPC のデフォルトのセキュリティグループにはできません。

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

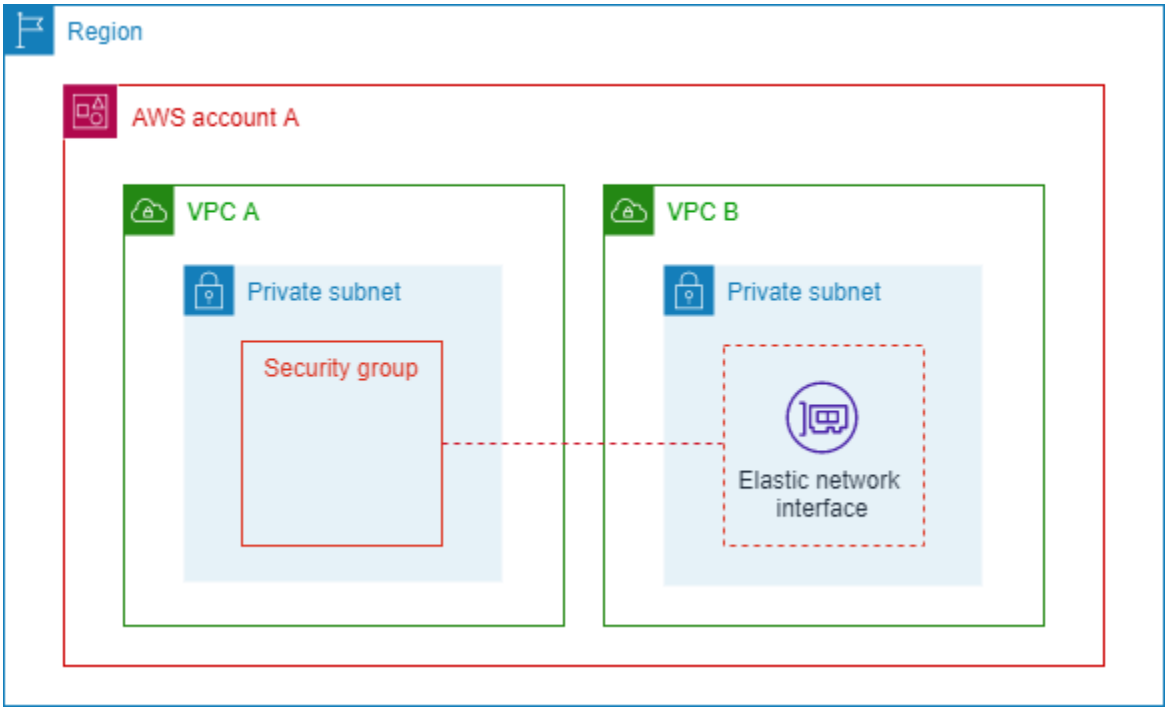
1. Amazon VPC コンソール (https://console.aws.amazon.com/vpc/) を開きます。
2. ナビゲーションペインで [セキュリティグループ] をクリックします。
3. セキュリティグループを選択して、[アクション]、[セキュリティグループを削除] を選択します。
4. 複数のセキュリティグループを選択した場合は、確認を求められます。一部のセキュリティグループを削除できない場合は、削除されるかどうかを示す、各セキュリティグループのステータスが表示されます。削除を確認するには、「Delete」と入力します。
5. [削除] を選択します。

AWS CLI 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 コマンドを使用します。

セキュリティグループを複数の VPCに関連付ける

ネットワークセキュリティ要件を共有する複数の VPC でワークロードを実行している場合は、セキュリティグループの VPC の関連付け機能を使用して、セキュリティグループを同じリージョン内の複数の VPC に関連付けることができます。これにより、アカウント内の複数の VPC のために、セキュリティグループを 1 か所で管理および維持できます。



上記の図は、2 つの VPC がある AWS アカウント A を示しています。各 VPC には、プライベートサブネットで実行されているワークロードがあります。この場合、VPC A および B サブネットのワークロードは同じネットワークトラフィック要件を共有するため、アカウント A はセキュリティグループの VPC の関連付け機能を使用して VPC A のセキュリティグループを VPC B に関連付けることができます。関連付けられたセキュリティグループに加えられた更新は、VPC B サブネットのワークロードへのトラフィックに自動的に適用されます。

セキュリティグループの VPC の関連付け機能の要件

- セキュリティグループを VPC に関連付けるには、VPC を所有するか、または VPC サブネットの 1 つを共有する必要があります。
- VPC とセキュリティグループは同じ AWS リージョンに存在する必要があります。

- デフォルトのセキュリティグループを別の VPC に関連付けたり、セキュリティグループをデフォルトの VPC に関連付けたりすることはできません。
- セキュリティグループの所有者と VPC 所有者の両方が、セキュリティグループの VPC の関連付けを表示できます。

この機能をサポートするサービス

- Amazon API Gateway (REST API のみ)
- AWS Auto Scaling
- CloudFormation
- Amazon EC2
- Amazon EFS
- Amazon EKS
- Amazon FSx
- AWS PrivateLink
- Amazon Route 53
- Elastic Load Balancing
  - Application Load Balancer
  - Network Load Balancer

セキュリティグループを別の VPC に関連付ける

このセクションでは、AWS マネジメントコンソールと AWS CLI を使用してセキュリティグループを VPC に関連付ける方法について説明します。

AWS Management Console

セキュリティグループを別の VPC と関連付けるには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 左のナビゲーションペインで、[セキュリティグループ] を選択します。
3. セキュリティグループを選択して、詳細を表示します。
4. [VPC の関連付け] タブを選択します。
5. [Associate VPC] を選択します。

6. [VPC ID] で、セキュリティグループに関連付ける VPC を選択します。
7. [Associate VPC] を選択します。

Command line

セキュリティグループを別の VPC と関連付けるには

1. associate-security-group-vpc との VPC の関連付けを作成します。
2. describe-security-group-vpc-associations との VPC の関連付けのステータスを確認し、ステータスが `associated` になるまで待ちます。

これで、VPC がセキュリティグループに関連付けられました。

VPC をセキュリティグループに関連付けると、VPC 内にインスタンスを起動して、この新しいセキュリティグループを選択すること、このセキュリティグループを既存のセキュリティグループルールで参照することなどが可能になります。

別の VPC からセキュリティグループの関連付けを解除する

このセクションでは、AWS マネジメントコンソールと AWS CLI を使用して VPC からセキュリティグループの関連付けを解除する方法について説明します。セキュリティグループの削除を目標としている場合、この操作を実行することが考えられます。セキュリティグループは、関連付けられている場合は削除できません。セキュリティグループの関連付けを解除できるのは、そのセキュリティグループを使用する関連付けられた VPC にネットワークインターフェイスがない場合のみです。

AWS Management Console

VPC からセキュリティグループの関連付けを解除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 左のナビゲーションペインで、[セキュリティグループ] を選択します。
3. セキュリティグループを選択して、詳細を表示します。
4. [VPC の関連付け] タブを選択します。
5. [VPC の関連付けを解除] を選択します。
6. [VPC ID] で、セキュリティグループから関連付けを解除する VPC を選択します。
7. [VPC の関連付けを解除] を選択します。

8. VPC の関連付けタブで関連付け解除の [ステータス] を表示し、ステータスが `disassociated` になるまで待ちます。

Command line

VPC からセキュリティグループの関連付けを解除するには

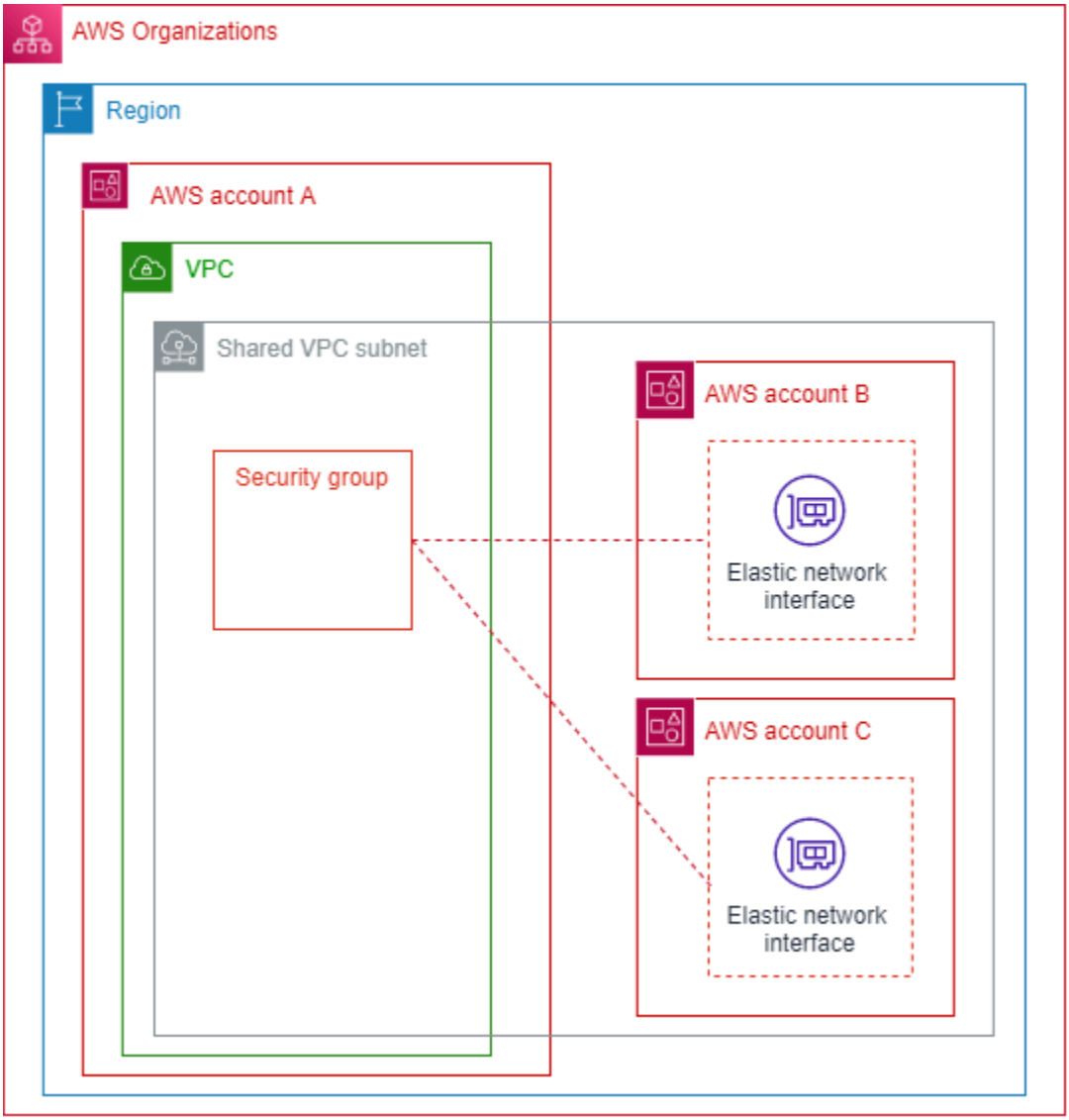
1. disassociate-security-group-vpc を使用して VPC の関連付けを解除します。
2. describe-security-group-vpc-associations との VPC の関連付け解除のステータスを確認し、ステータスが `disassociated` になるまで待ちます。

これで、VPC とセキュリティグループの関連付けが解除されました。

AWS Organizations とセキュリティグループを共有する

共有セキュリティグループ機能を使用すると、同じ AWS リージョンにある他の AWS Organizations アカウントとセキュリティグループを共有し、それらのアカウントでセキュリティグループを使用できるようになります。

次の図は、共有セキュリティグループ機能を使用して、AWS Organizations 内のアカウント間のセキュリティグループ管理を簡素化する方法を示しています:



この図は、同じ Organization の一部である 3 つのアカウントを示しています。アカウント A は VPC サブネットをアカウント B および C と共有します。アカウント A は、共有セキュリティグループ機能を使用して、アカウント B および C とセキュリティグループを共有します。その後、アカウント B と C は、共有サブネットでインスタンスを起動する際に、そのセキュリティグループを使用します。これにより、アカウント A はセキュリティグループを管理できます。セキュリティグループに対する更新は、アカウント B と C が共有 VPC サブネットで実行しているリソースに適用されます。

共有セキュリティグループ機能の要件

- この機能は、AWS Organizations の同じ Organizations 内のアカウントでのみ使用できます。AWS Organizations で [リソース共有] が有効になっている必要があります。

- セキュリティグループを共有するアカウントは、VPC とセキュリティグループの両方を所有している必要があります。
- デフォルトのセキュリティグループを共有することはできません。
- デフォルトの VPC にあるセキュリティグループを共有することはできません。
- 参加者アカウントは共有 VPC にセキュリティグループを作成できますが、それらのセキュリティグループを共有することはできません。
- IAM プリンシパルがセキュリティグループを AWS RAM と共有するには、最小限の許可セットが必要です。共有セキュリティグループを共有および使用するために必要な許可が IAM プリンシパルに付与されるよう、`AmazonEC2FullAccess` および `AWSResourceAccessManagerFullAccess` マネージド IAM ポリシーを使用します。カスタム IAM ポリシーを使用する場合は、`c2:PutResourcePolicy` および `ec2:DeleteResourcePolicy` アクションが必要です。これらはアクセス許可のみの IAM アクションです。IAM プリンシパルにこれらのアクセス許可が付与されていない場合、AWS RAM を使用してセキュリティグループを共有しようとすると エラーが発生します。

この機能をサポートするサービス

- Amazon API Gateway
- Amazon EC2
- Amazon ECS
- Amazon EFS
- Amazon EKS
- Amazon EMR
- Amazon FSx
- Amazon ElastiCache
- AWS Elastic Beanstalk
- AWS Glue
- Amazon MQ
- Amazon SageMaker AI
- Elastic Load Balancing
  - Application Load Balancer
  - Network Load Balancer

この機能が既存のクォータに及ぼす影響

セキュリティグループのクォータが適用されます。ただし、[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] のクォータでは、参加者が Elastic Network Interface (ENI) で所有グループと共有グループの両方を使用する場合、所有者と参加者のクォータの最小値が適用されます。

クォータがこの機能によってどのように影響を受けるかを示す例:

- 所有者アカウントのクォータ: インターフェイスあたり 4 つのセキュリティグループ
- 参加者アカウントのクォータ: インターフェイスあたり 5 つのセキュリティグループ
- 所有者は、グループ SG-O1、SG-O2、SG-O3、SG-O4、SG-O5 を参加者と共有します。参加者は、VPC に既に独自のグループを持っています: SG-P1、SG-P2、SG-P3、SG-P4、SG-P5。
- 参加者が ENI を作成し、自分の所有グループのみを使用する場合、5 つのセキュリティグループ (SG-P1、SG-P2、SG-P3、SG-P4、SG-P5) すべてを関連付けることができます。なぜなら、これが参加者のクォータだからです。
- 参加者が ENI を作成し、その ENI で共有グループを使用する場合、関連付けることができるグループは最大 4 つのみです。この場合、このような ENI のクォータは、所有者と参加者のクォータの最小値です。考えられる有効な設定は次のようになります:
  - SG-O1、SG-P1、SG-P2、SG-P3
  - SG-O1、SG-O2、SG-O3、SG-O4

セキュリティグループを共有する

このセクションでは、AWS マネジメントコンソールと AWS CLI を使用して、Organization 内の他のアカウントとセキュリティグループを共有する方法について説明します。

AWS Management Console

セキュリティグループを共有するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 左のナビゲーションペインで、[セキュリティグループ] を選択します。
3. セキュリティグループを選択して、詳細を表示します。
4. [共有] タブを選択します。
5. [セキュリティグループを共有] を選択します。
6. [リソースの共有の作成] を選択します。AWS RAM コンソールが開いたら、セキュリティグループのリソース共有を作成します。

7. リソース共有の [名前] を入力します。
8. [リソース - オプション] で、[セキュリティグループ] を選択します。
9. [セキュリティグループ] をクリックします。セキュリティグループをデフォルトのセキュリティグループにしたり、デフォルトの VPC に関連付けたりすることはできません。
10. [次へ] を選択します。
11. プリンシパルによる実行が許可されているアクションを確認し、[次へ] を選択します。
12. [プリンシパル - オプション] で [自分の組織内でのみ共有を許可] を選択します。
13. [プリンシパル] で、次のいずれかのプリンシパルタイプを選択し、適切な数値を入力します:

    - [AWS アカウント]: Organization 内のアカウントのアカウント番号。
    - [Organization]: AWS Organization ID。
    - [組織単位 (OU)]: Organization 内の OU の ID。
    - [IAM ロール]: IAM ロールの ARN。ロールを作成したアカウントは、このリソース共有を作成するアカウントと同じ Organization のメンバーである必要があります。
    - [IAM ユーザー]: IAM ユーザーの ARN。ユーザーを作成したアカウントは、このリソース共有を作成するアカウントと同じ Organization のメンバーである必要があります。
    - [サービスプリンシパル]: セキュリティグループをサービスプリンシパルと共有することはできません。

14. [Add] (追加) を選択します。
15. [次へ] を選択します。
16. [リソースの共有の作成] を選択します。
17. [共有リソース] で、`Associated` の [ステータス] が表示されるまで待ちます。セキュリティグループの関連付けに障害が発生した場合は、上記の制限のいずれかが原因である可能性があります。詳細ページでセキュリティグループの詳細と [共有中] タブを表示して、セキュリティグループを共有できない理由に関連するメッセージを確認します。
18. VPC コンソールのセキュリティグループリストに戻ります。
19. 共有したセキュリティグループを選択します。
20. [共有] タブを選択します。そこで AWS RAM リソースが表示されるはずです。表示されない場合、リソース共有の作成が失敗した可能性があり、再作成する必要がある場合があります。

Command line

セキュリティグループを共有するには

1. AWS RAM と共有するセキュリティグループのリソース共有を最初に作成する必要があります。AWS CLI を使用して AWS RAM とのリソース共有を作成するステップについては、「AWS RAM ユーザーガイド」の「AWS RAM でのリソース共有の作成」を参照してください。
2. 作成されたリソース共有の関連付けを表示するには、get-resource-share-associations を使用します。

これで、セキュリティグループが共有されました。同じ VPC 内の共有サブネットで EC2 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選択できます。

セキュリティグループの共有を停止する

このセクションでは、AWS マネジメントコンソールと AWS CLI を使用して、Organization 内の他のアカウントとセキュリティグループの共有を停止する方法について説明します。

AWS Management Console

セキュリティグループの共有を停止するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 左のナビゲーションペインで、[セキュリティグループ] を選択します。
3. セキュリティグループを選択して、詳細を表示します。
4. [共有] タブを選択します。
5. セキュリティグループのリソース共有を選択し、[共有を停止] を選択します。
6. [はい、共有を停止します] を選択します。

Command line

セキュリティグループの共有を停止するには

delete-resource-share を使用してリソース共有を削除します。

セキュリティグループは共有されなくなりました。所有者がセキュリティグループの共有を停止すると、次のルールが適用されます:

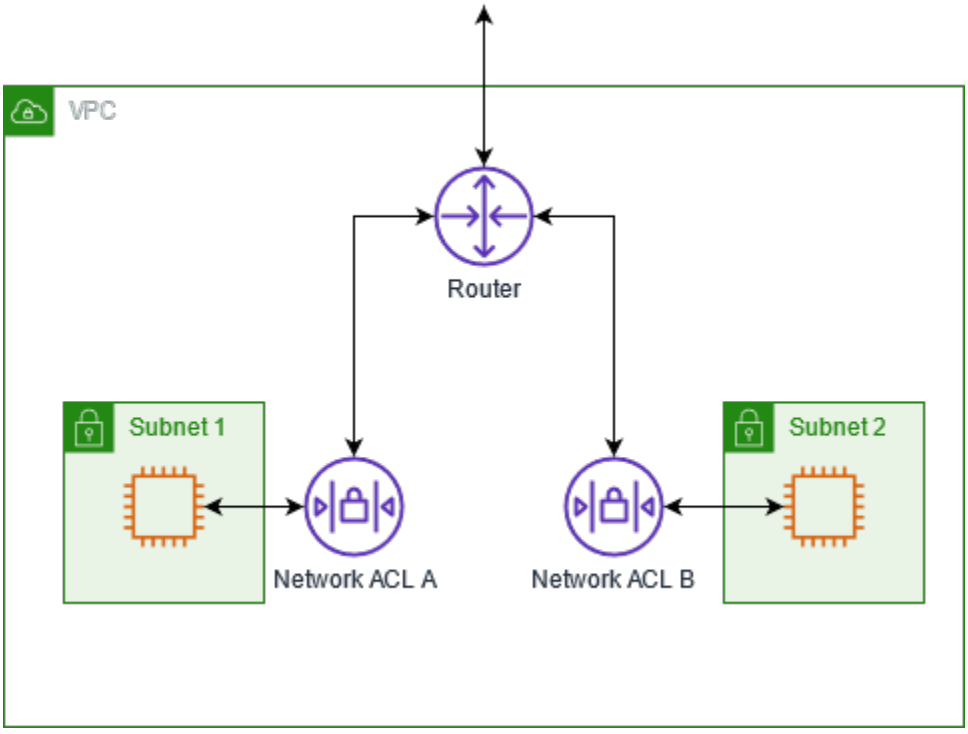
- 既存の参加者の Elastic Network Interface (ENI) は、共有解除されたセキュリティグループに対して行われたセキュリティグループルールの更新を引き続き取得します。共有の解除による影響は、参加者が共有解除されたグループと新しい関連付けを作成できなくなることだけです。
- 参加者は、共有解除されたセキュリティグループを所有している ENI に関連付けることができなくなります。
- 参加者は、共有解除されたセキュリティグループに引き続き関連付けられている ENI を記述および削除できます。
- 共有解除されたセキュリティグループに関連付けられている ENI を参加者が引き続き持っている場合、所有者は、共有解除されたセキュリティグループを削除できません。所有者は、参加者がすべての ENI からセキュリティグループの関連付けを解除 (セキュリティグループを削除) した後にのみ、セキュリティグループを削除できます。
- 参加者は、共有されていないセキュリティグループに関連付けられた ENI を使用して新しい EC2 インスタンスを起動することはできません。

ネットワークアクセスコントロールリストを使用して、サブネットのトラフィックを制御する

ネットワークアクセスコントロールリスト (ACL) は、サブネットレベルで特定のインバウンドまたはアウトバウンドのトラフィックを許可または拒否します。VPC のデフォルトのネットワーク ACL を使用するか、セキュリティグループルールと同様のルールを使用して VPC 用のカスタムネットワーク ACL を作成できます。カスタムネットワーク ACL は、もう 1 つのセキュリティレイヤーを VPC に追加します。

ネットワーク ACL は追加料金なしで使用できます。

次の図は、2 つのサブネットを持つ VPC を示しています。各サブネットにはネットワーク ACL があります。トラフィックが (ピアリングされた VPC、VPN 接続、インターネットなどから) VPC に入ると、ルーターはこのトラフィックを宛先に送信します。ネットワーク ACL A は、サブネット 1 を宛先とするトラフィックのうち、サブネット 1 への送信を許可するトラフィックと、サブネット 1 以外を宛先とするトラフィックのうち、サブネット 1 からの送信を許可するトラフィックを決定します。同様に、ネットワーク ACL B は、どのトラフィックがサブネット 2 に出入りできるかを決定します。



セキュリティグループとネットワーク ACL の違いについては、「セキュリティグループとネットワーク ACL を比較する」を参照してください。

内容

- ネットワーク ACL の基本
- ネットワーク ACL ルール
- VPC のデフォルトのネットワーク ACL
- VPC のカスタムネットワーク ACL
- パス MTU 検出とネットワーク ACL
- VPC のネットワーク ACL を作成する
- VPC のネットワーク ACL の関連付けを管理する
- VPC のネットワーク ACL を削除する
- 例: サブネットのインスタンスへのアクセス制御

ネットワーク ACL の基本

開始する前にネットワーク ACL について知っておくべき基本的な情報を以下に示します。

ネットワーク ACL の関連付け

- 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。
- カスタムネットワーク ACL を作成して、それをサブネットに関連付けることで、サブネットレベルで特定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を許可または拒否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 ルール

- ネットワーク ACL には、インバウンドルールとアウトバウンドルールがあります。ネットワーク ACL ごとに設定できるルールの数にはクォータ (または制限) が設定されています。各ルールでは、トラフィックを許可または拒否できます。各ルールには 1 から 32,766 までの番号が設定されます。トラフィックを許可するか拒否するかを決定する際は、最も低い番号のルールから順にルールを評価します トラフィックがルールに一致すると、そのルールが適用され、追加のルールは評価されません。まずは増分 (例えば 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、必要になったときに後で新しいルールを挿入できます。
- ネットワーク ACL ルールは、トラフィックがサブネット内でルーティングされるときではなく、サブネットに出入りするときに評価されます。
- NACL はステートレスです。つまり、以前に送受信されたトラフィックに関する情報は保存されません。例えば、サブネットへの特定のインバウンドトラフィックを許可する NACL ルールを作成しても、そのトラフィックへの応答は自動的には許可されません。これは、セキュリティグループの仕組みとは対照的です。セキュリティグループはステートフルです。つまり、以前に送受信されたトラフィックに関する情報が保存されます。例えば、セキュリティグループが EC2 インスタンスへのインバウンドトラフィックを許可している場合、アウトバウンドセキュリティグループのルールにかかわらず、レスポンスは自動的に許可されます。

制限事項

- ネットワーク ACL にはクォータ (制限とも呼ばれます) が設定されています。詳細については、「ネットワーク ACL」を参照してください。

- ネットワーク ACL では、Route 53 Resolver (VPC+2 IP アドレスまたは AmazonProvidedDNS とも呼ばれます) で送受信される DNS リクエストをブロックすることはできません。Route 53 Resolver 経由の DNS リクエストをフィルタリングするために、Route 53 Resolver DNS Firewall を有効化できます。
- ネットワーク ACL では、インスタンスメタデータサービス (IMDS) へのトラフィックをブロックすることはできません。IMDS へのアクセスを管理するには、「Amazon EC2 ユーザーガイド」の「インスタンスメタデータオプションの設定」を参照してください。
- ネットワーク ACL では、以下で送受信されるトラフィックはフィルターされません。
  - 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 ECS タスクメタデータエンドポイント
  - Windows インスタンスのライセンスアクティベーション
  - Amazon Time Sync Service
  - デフォルトの VPC ルーターによる予約済み IP アドレス

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。

ロールの例については、「the section called “例: サブネットのインスタンスへのアクセス制御”」を参照してください。

考慮事項

- ネットワーク ACL あたりのルールの数には、クォータ (制限とも呼ばれます) があります。詳細については、「Amazon VPC クォータ」を参照してください。
- 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。変更は短期間で有効になります。
- 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。
- 広い範囲のポートを開く必要があるが、その範囲内の特定のポートは拒否する場合は、拒否ルールを追加する必要があります。拒否ルールには、より広範なポートトラフィックを許可するルールよりも少ない数を指定してください。
- ネットワーク ACL のルールを同時に追加および削除する場合は、注意が必要です。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、許可されている数より多くのエントリを追加した場合 (Amazon VPC クォータ を参照)、削除対象として選択されたエントリは削除され、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 へのアクセスや VPC からのアクセスが妨げられる可能性があります。

VPC のデフォルトのネットワーク ACL

仮想プライベートクラウド (VPC) には、デフォルトのネットワーク ACL が自動的に付属します。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 には、ルール番号がアスタリスク (*) であるルールも含まれます。これらのルールにより、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。

ルールを追加したり、デフォルトの番号付きルールを削除したりすることで、デフォルトのネットワーク ACL を変更できます。ルール番号がアスタリスクになっているルールは削除できません。

デフォルトのインバウンドルール

次の表は、デフォルトのネットワーク ACL のデフォルトのインバウンドルールを示しています。IPv6 のルールは、関連付けられた IPv6 CIDR ブロックを使用して VPC を作成するか、IPv6 CIDR ブロックを VPC に関連付ける場合にのみ追加されます。ただし、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、すべてのインバウンド IPv6 トラフィックを許可するルールは追加されません。

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	すべての IPv4 トラフィック	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
*	すべてのトラフィック	すべて	すべて	0.0.0.0/0	DENY
*	すべての IPv6 トラフィック	すべて	すべて	::/0	DENY

デフォルトのアウトバウンドルール

次の表は、デフォルトネットワーク ACL のデフォルトのアウトバウンドルールを示しています。IPv6 のルールは、関連付けられた IPv6 CIDR ブロックを使用して VPC を作成するか、IPv6 CIDR ブロックを VPC に関連付ける場合にのみ追加されます。ただし、デフォルトのネットワーク ACL のアウトバウンドルールを変更した場合、IPv6 ブロックを VPC に関連付けるときにすべてのアウトバウンド IPv6 トラフィックを許可するルールは追加されません。

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	すべてのトラフィック	すべて	すべて	0.0.0.0/0	許可

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
101	すべての IPv6 トラフィック	すべて	すべて	::/0	許可
*	すべてのトラフィック	すべて	すべて	0.0.0.0/0	DENY
*	すべての IPv6 トラフィック	すべて	すべて	::/0	DENY

VPC のカスタムネットワーク ACL

カスタムネットワーク ACL を作成して、それをサブネットに関連付けることで、サブネットレベルで特定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を許可または拒否することができます。詳細については、「the section called “ネットワーク ACL の作成”」を参照してください。

各ネットワーク ACL には、デフォルトのインバウンドルールと、ルール番号がアスタリスク (*) であるデフォルトの送信ルールが含まれます。これらのルールにより、パケットが他のどのルールにも一致しない場合は拒否されます。

ルールを追加または削除することで、ネットワーク ACL を変更できます。ルール番号がアスタリスクになっているルールは削除できません。

追加するルールごとに、応答トラフィックを許可するインバウンドルールまたはアウトバウンドルールが必要です。適切な一時ポートの範囲を選択する方法については、「一時ポート」を参照してください。

インバウンドルールの例

次の表は、ネットワーク ACL のインバウンドルールの例を示しています。IPv6 のルールは、VPC に関連付けられた IPv6 CIDR ブロックがある場合にのみ追加されます。IPv4 トラフィックと IPv6 トラフィックは個別に評価されます。したがって、IPv4 トラフィックのルールは IPv6 トラフィックには適用されません。対応する IPv4 ルールの横に IPv6 ルールを追加したり、最後の IPv4 ルールの後に IPv6 ルールを追加したりもできます。

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に対して、最も小さい番号のルールから順に評価されます。例えば、HTTPS ポート (443) 宛ての IPv4 トラフィックがあるとします。パケットがルール 100 または 105 と一致しません。これは、サブネットへのトラフィックを許可するルール 110 と一致します。パケットの宛先がポート 139 (NetBIOS) だった場合、番号付きルールのいずれにも一致しないため、IPv4 トラフィックの * ルールによって最終的にパケットが拒否されます。

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。

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
140	カスタム TCP	TCP	32768-65535	0.0.0.0/0	許可	(サブネットから発信されたリクエストの場合) 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。
145	カスタム TCP	TCP	32768-65535	::/0	許可	(送信元がサブネットであるリクエストの場合) 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。
*	すべてのトラフィック	すべて	すべて	0.0.0.0/0	DENY	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の例

次の表は、カスタムネットワーク ACL のアウトバウンドルールの例を示しています。IPv6 のルールは、VPC に関連付けられた IPv6 CIDR ブロックがある場合にのみ追加されます。IPv4 トラフィック

と IPv6 トラフィックは個別に評価されます。したがって、IPv4 トラフィックのルールは IPv6 トラフィックには適用されません。対応する IPv4 ルールの横に IPv6 ルールを追加したり、最後の IPv4 ルールの後に IPv6 ルールを追加したりもできます。

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。
120	カスタム TCP	TCP	*1024-655 5*	*192.0.2.(/24*	許可	ホーム ネットワークからの SSH トラフィックへのアウトバンド応答を許可します。

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
140	カスタム 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: ウェブページの提供) 。
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: ウェブページの提供) 。
*	すべてのトラフィック	すべて	すべて	0.0.0.0/0	DENY	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。
*	すべてのトラフィック	すべて	すべて	::/0	DENY	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。

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、インスタンスのオペレーティング システム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。

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、以下を参照してください。

- Network ACLs for your Application Load Balancer
- Network ACLs for your Network Load Balancer

- Network ACLs for your Classic Load Balancer

到達可能性に関する問題のトラブルシューティング

Reachability Analyzer は静的な設定分析ツールです。Reachability Analyzer を使用して、VPC 内の 2 つのリソース間のネットワーク到達可能性を分析およびデバッグできます。Reachability Analyzer は、これらのリソースに到達可能な場合は、リソース間にある仮想パスのホップバイホップの詳細を生成し、そうでない場合はブロッキングコンポーネントを識別します。例えば、欠落した、または誤って設定されたネットワーク ACL のルールを特定できます。

詳細については、「Reachability Analyzer Guide」(到達可能性アナライザーガイド) を参照してください。

パス MTU 検出とネットワーク ACL

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、ホストが受信側ホストの MTU またはパス上のデバイスの MTU よりも大きなパケットを送信すると、受信側ホストまたはデバイスはそのパケットを削除します。次に、「`Destination Unreachable: Fragmentation Needed and Don't Fragment was Set`」という ICMP メッセージ (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。ホストが受信側ホストの MTU またはパス上のデバイスの MTU よりも大きなパケットを送信すると、受信側ホストまたはデバイスはそのパケットを削除します。次に、「`ICMPv6 Packet Too Big (PTB)`」という ICMP メッセージ (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、「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。

VPC のネットワーク ACL を作成する

次のタスクは、ネットワーク ACL を作成し、ルールをネットワーク ACL に追加してから、ネットワーク ACL をサブネットに関連付ける方法を示します。

タスク

- ステップ 1: ネットワーク ACL を作成する
- ステップ 2: ルールを追加する
- ステップ 3: サブネットをネットワーク ACL に関連付ける
- (オプション) Firewall Manager を使用してネットワーク ACL を管理する

ステップ 1: ネットワーク ACL を作成する

VPC のカスタムネットワーク ACL を作成できます。カスタムネットワーク ACL の初期ルールは、すべてのインバウンドトラフィックとアウトバウンドトラフィックをブロックします。新しいカスタムネットワーク ACL は、デフォルトではサブネットに関連付けられていないため、明示的にサブネットに関連付ける必要があります。

コンソールを使用してネットワーク ACL を作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [Network ACLs] を選択します。
3. [ネットワーク ACL の作成] を選択します。
4. (オプション) [名前] に、ネットワーク ACL の名前を入力します。
5. [VPC] で、VPC を選択します。
6. (オプション) [タグ] で、[タグの追加] を選択して、タグのキーと値を指定します。
7. [ネットワーク ACL の作成] を選択します。

コマンドラインを使用してネットワーク ACL を作成するには

- create-network-acl (AWS CLI)
- New-EC2NetworkAcl (AWS Tools for Windows PowerShell)

ステップ 2: ルールを追加する

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許可または拒否するルールを追加できます。

最も小さい番号のルールから順にルールを処理します。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。

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。

コンソールを使用してネットワーク ACL にルールを追加するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [Network ACLs] を選択します。
3. ネットワーク ACL を選択します。
4. インバウンドルールを追加するには、次の手順を実行します。
   a. [Inbound rules] (インバウンドルール) タブを開きます。
   b. [インバウンドのルールの編集] ページで、[新しいルールの追加 を選択します。
   c. まだ使用されていないルール番号、タイプ、プロトコル、ポート範囲、ソース、トラフィックを許可または拒否するかどうかを入力します。一部のタイプでは、プロトコルとポートが自動的に入力されます。ポート範囲の入力を求められた場合は、ポート番号またはポート範囲 (49152-65535 など) を入力します。

      リストにないプロトコルを使用するには、タイプの [カスタムプロトコル] をクリックし、プロトコルを選択します。詳細については、「IANA プロトコル番号」を参照してください。
   d. [Save changes] (変更の保存) をクリックします。
5. アウトバウンドルールを追加するには、次の手順を実行します。
   a. [Outbound rules] (アウトバウンドルール) タブを選択します。
   b. [アウトバウンドのルールの編集]、[ルールの追加] の順に選択します。

c. まだ使用されていないルール番号、タイプ、プロトコル、ポート範囲、ソース、トラフィックを許可または拒否するかどうかを入力します。一部のタイプでは、プロトコルとポートが自動的に入力されます。ポート範囲の入力を求められた場合は、ポート番号またはポート範囲 (49152-65535 など) を入力します。

   リストにないプロトコルを使用するには、タイプの [カスタムプロトコル] をクリックし、プロトコルを選択します。詳細については、「IANA プロトコル番号」を参照してください。

d. [Save changes] (変更の保存) をクリックします。

コマンドラインを使用してネットワーク ACL にルールを追加するには

- create-network-acl-entry (AWS CLI)
- New-EC2NetworkAclEntry (AWS Tools for Windows PowerShell)

コマンドラインを使用してネットワーク ACL のルールを置き換えるには

- replace-network-acl-entry (AWS CLI)
- Set-EC2NetworkAclEntry (AWS Tools for Windows PowerShell)

コマンドラインを使用してネットワーク ACL からルールを削除するには

- delete-network-acl-entry (AWS CLI)
- Remove-EC2NetworkAclEntry (AWS Tools for Windows PowerShell)

ステップ 3: サブネットをネットワーク ACL に関連付ける

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。

(オプション) Firewall Manager を使用してネットワーク ACL を管理する

AWS Firewall Manager は、複数のアカウントおよび複数のサブネット間でネットワーク ACL の管理およびメンテナンスタスクを簡略化します。Firewall Manager を使用して、組織内のアカウントとサブネットをモニタリングし、定義したネットワーク ACL の設定を自動的に適用できます。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自動的に保護する新しいサブネットを頻繁に追加する場合に特に便利です。

Firewall Manager のネットワーク ACL ポリシーでは、単一の管理者アカウントを使用して、組織全体で使用するネットワーク ACL で定義する最小ルールセットを設定、モニタリング、および管理できます。組織内のどのアカウントとサブネットが Firewall Manager ポリシーの範囲内にあるかを指定します。Firewall Manager は、対象範囲内のサブネットのネットワーク ACL のコンプライアンスステータスを報告します。また、Firewall Manager を設定して、コンプライアンス違反のネットワーク ACL の修復を自動化することもできます。

詳細については、「AWS Firewall Manager デベロッパーガイド」で以下のリソースを参照してください。

- AWS Firewall Manager の前提条件
- AWS Firewall Manager ネットワーク ACL ポリシーのセットアップ
- Firewall Manager でネットワーク ACL ポリシーを使用する

VPC のネットワーク ACL の関連付けを管理する

各サブネットは 1 つのネットワーク ACL に関連付けられます。サブネットを初めて作成すると、VPC のデフォルトのネットワーク ACL に関連付けられます。カスタムネットワーク ACL を作成し、それを 1 つ以上のサブネットに関連付け、以前のネットワーク ACL の関連付けを置き換えることができます。

タスク

- ネットワーク ACL の関連付けを記述する
- ネットワーク ACL に関連付けられたサブネットを変更する

- サブネットに関連付けられたネットワーク ACL を変更する

ネットワーク ACL の関連付けを記述する

サブネットに関連付けられているネットワーク ACL を記述したり、ネットワーク ACL に関連付けられているサブネットを記述したりすることもできます。

コンソールを使用してサブネットに関連付けられたネットワーク ACL を記述するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[サブネット] を選択してください。
3. サブネットを選択します。
4. [ネットワーク ACL] タブを選択します。

AWS CLI を使用してサブネットに関連付けられたネットワーク ACL を記述するには

指定されたサブネットに関連付けられているネットワーク ACL を一覧表示するには、次の describe-network-acls コマンドを使用します。

```
aws ec2 describe-network-acls --filters Name=association.subnet-
id,Values=subnet-0d2d1b81e0bc9c6d4 --query NetworkAcls[*].NetworkAclId
```

以下は出力の例です。

```
[
    "acl-03701d1f82d8c3fd6"
]
```

コンソールを使用してネットワーク ACL に関連付けられたサブネットを記述するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [Network ACLs] を選択します。
3. ネットワーク ACL を選択します。
4. [サブネットの関連付け] タブを選択します。

AWS CLI を使用してネットワーク ACL に関連付けられたサブネットを記述するには

指定されたネットワーク ACL に関連付けられたサブネットを一覧表示するには、次の describe-network-acls コマンドを使用します。

```
aws ec2 describe-network-acls --network-acl-ids acl-060415a18fcc9afde --query
 NetworkAcls[*].Associations[].SubnetId
```

以下は出力の例です。

```
[
    "subnet-0d2d1b81e0bc9c6d4",
    "subnet-0e990c67809773b19",
    "subnet-0eb17d85f5dfd33b1",
    "subnet-0e01d500780bb7468"
]
```

ネットワーク ACL に関連付けられたサブネットを変更する

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。サブネットをカスタム ネットワーク ACL から関連付け解除すると、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。変更はしばらくすると有効になります。

ネットワーク ACL に関連付けられたサブネットを変更するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [Network ACLs] を選択します。
3. ネットワーク ACL を選択します。
4. [アクション]、[サブネットの関連付けの編集] を選択します。
5. [選択済みサブネット] からサブネットを削除します。
6. [Save changes] (変更の保存) をクリックします。

サブネットに関連付けられたネットワーク ACL を変更する

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、初期設定では VPC のデフォルトのネットワーク ACL に関連付けられます。カスタムネットワーク ACL を作成する場合は、ネットワーク ACL を 1 つ以上のサブネットに関連付けることで、ネットワーク ACL ルールを適用します。

サブネットのネットワーク ACL を変更すると、変更はしばらくすると有効になります。

サブネットに関連付けられたネットワーク ACL を変更するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインで、[サブネット] を選択してください。
3. サブネットを選択します。
4. [アクション]、[ネットワーク ACL の関連付けの編集] の順に選択します。
5. ネットワーク ACL ID で、サブネットに関連付けるネットワーク ACL を選択し、選択したネットワーク ACL のインバウンドルールとアウトバウンドルールを確認します。
6. [保存] を選択します。

コマンドラインを使用してネットワーク ACL の関連付けを置き換えるには

- replace-network-acl-association (AWS CLI)
- Set-EC2NetworkAclAssociation (AWS Tools for Windows PowerShell)

VPC のネットワーク ACL を削除する

ネットワーク ACL の使用が終了したら、削除できます。サブネットが関連付けられている場合は、ネットワーク ACL を削除できません。デフォルトのネットワーク ACL は削除できません。

コンソールを使用してネットワーク ACL からサブネットの関連付けを削除するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [Network ACLs] を選択します。[関連付け] 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。関連付けられたサブネットがない場合、この列は - になります。
3. ネットワーク ACL を選択します。
4. [アクション]、[サブネットの関連付けの編集] を選択します。
5. サブネットの関連付けを削除します。
6. [Save changes] (変更の保存) をクリックします。

コマンドラインを使用して、関連付けを含むネットワーク ACL を記述するには

- describe-network-acls (AWS CLI)
- Get-EC2NetworkAcl (AWS Tools for Windows PowerShell)

コマンドラインを使用してネットワーク ACL の関連付けを置き換えるには

- replace-network-acl-association (AWS CLI)
- Set-EC2NetworkAclAssociation (AWS Tools for Windows PowerShell)

コンソールを使用してネットワーク ACL を削除するには

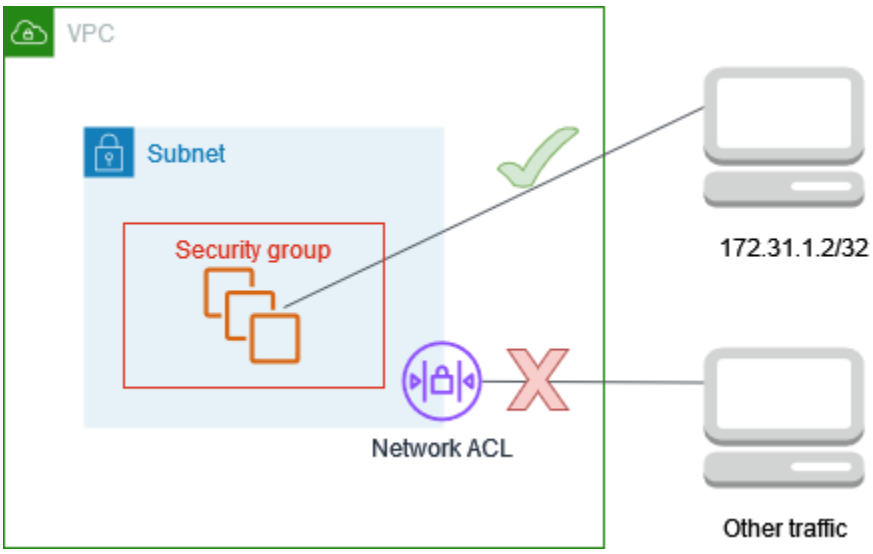
1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ナビゲーションペインの [Network ACLs] を選択します。
3. ネットワーク ACL を選択します。
4. [アクション]、[ネットワーク ACL の削除] の順に選択します。
5. 確認を求められたら、**delete**と入力し、[削除] を選択します。

コマンドラインを使用してネットワーク ACL を削除するには

- 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、サブネットにあるインスタンスは相互に通信できます。これらのインスタンスは信頼できるリモートコンピューターからアクセスして管理タスクを実行できます。リモートコンピュータは、図に示すようにローカル ネットワーク内のコンピュータである場合もあれば、別のサブネットまたは VPC 内のインスタンスである場合もあります。サブネットのネットワーク ACL ルールとインスタンスのセキュリティグループルールにより、リモートコンピューターの IP アドレスからのアクセスが許可されます。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。



ネットワーク ACL を使用すると、ネットワーク ACL を防御のバックアップレイヤーとして利用しながら、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柔軟に変更できるようになります。

例えば、セキュリティグループを誤って更新して任意の場所からのインバウンド SSH アクセスを許可してしまった場合でも、ネットワーク ACL はアクセスをリモートコンピュータの IP アドレス範囲のみに制限します。ネットワーク ACL は、他の IP アドレスからのインバウンド SSH トラフィックを拒否します。

ネットワーク ACL ルール

次は、サブネットに関連付けられたネットワーク ACL のインバウンドルールの例を示します。これらの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。

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	DENY	他のすべてのインバ

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
						ウンドトラフィックを拒否します。

以下は、サブネットに関連付けられたネットワーク ACL の送信ルールの例です。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を含める必要があります。

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。

セキュリティグループのルール

以下は、インスタンスに関連付けられたセキュリティ グループのインバウンドルールの例です。これらのルールは、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適用されます。インスタンスに関連付けられたキーペアのプライベートキーを持つユーザーは、SSH を使用してリモートコンピュータからインスタンスに接続できます。

プロトコルのタイプ	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インスタンス間の通信を許可します。
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。

以下は、インスタンスに関連付けられたセキュリティグループのアウトバウンドバウンドルールの例です。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。

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インスタンス間の通信を許可します。

ネットワーク ACL とセキュリティグループの違い

次の表は、ネットワーク ACL とセキュリティグループの基本的な違いをまとめたものです。

特性	ネットワーク ACL	セキュリティグループ
オペレーションのレベル	サブネットレベル	インスタンスレベル
範囲	関連付けられたサブネット内のすべてのインスタンスに適用されます	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適用されます
ルールタイプ	許可ルールと拒否ルール	ルールのみを許可します
ルールの評価	トラフィックの一致が見つかるまで昇順で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める前に、すべてのルールを評価します
返されるトラフィック	明示的に許可する必要があります (ステートレス)	自動許可 (ステートフル)

Amazon Virtual Private Cloud での耐障害性

AWS グローバルインフラストラクチャは AWS リージョン およびアベイラビリティーゾーンを中心に構築されています。AWS リージョン には、低レイテンシー、高スループットで、高度な冗長ネットワークを使用して接続され、物理的に独立および隔離された複数の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ー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、フォールトトレランス、および拡張性が優れています。

AWS リージョン は主要な構成要素であり、それぞれが、物理的に分離され隔離された複数のアベイラビリティーゾーンを格納する、個別の地理的位置を表しています。これらのアベイラビリティーゾーンは、低レイテンシーで高スループットかつ高度に冗長なネットワークファブリックを介して接続されており、相互のシームレスな通信とデータ転送が可能です。

アベイラビリティーゾーンのアーキテクチャは、従来の単一または複数のデータセンター構成よりもはるかに堅牢にかつ耐障害性が高くなるように設計されており、重要な差別化要因となっています。リージョン内の複数のアベイラビリティーゾーンにリソースを分散することにより、アプリケーションとデータベースを、サービスを中断することなくゾーン間で自動的にフェイルオーバーするように設計することができます。このレベルの冗長性と高可用性は、ミッションクリティカルなワークロー

ドにとって不可欠な要件であり、組織は回復力のあるクラウドネイティブソリューションを構築することが可能になります。

さらに、AWS のインフラストラクチャのスケールと世界的な展開力により、ユーザーは、エンドユーザーに近い場所にアプリケーションをデプロイし、レイテンシーを削減してユーザーエクスペリエンス全般を高めることができます。また、世界中の複数のリージョンを利用できるため、特定の規制やビジネスニーズによって要請される地理的な境界内でデータを保存し処理することにより、データ主権とコンプライアンスを効果的に実現できます。

AWS のグローバルなインフラストラクチャを使用することで、組織は、要件の変化やビジネスニーズの進化に柔軟に対応できる、可用性、耐障害性、スケーラビリティに優れたクラウド環境を構築できます。こうした堅牢な基盤は、最新のクラウドベースのアプリケーションやサービスを最適に実装するために欠かせない要素です。

AWS リージョン とアベイラビリティーゾーン (AZ) 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。

ワークロードのレジリエンス要件を満たすように VPC を設定できます。詳細については次を参照してください:

- レジリエンシーのパターンとトレードオフを理解する (AWS アーキテクチャブログ)
- ネットワークトポロジーの計画 (AWS Well-Architected フレームワーク)
- Amazon Virtual Private Cloud の接続オプション (AWS ホワイトペーパー)

Amazon Virtual Private Cloud のコンプライアンス検証

新しい AWS サインアップエクスペリエンスは、規制対象ワークロード向けには設計されていません。新しい AWS サインアップエクスペリエンスを使用しているものの、規制対象ワークロードに AWS を使用する場合は、AWS (高度) にサインアップしたり、AWS 環境の高度な機能をアクティブ化したりできます。

AWS のサービスが特定のコンプライアンスプログラムの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、関心のあるコンプライアンスプログラムを選択してください。一般的な情報については、AWSコンプライアンスプログラム を参照してください。

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

AWS の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、お客様のデータの機密性や貴社のコンプライアンス目的、適用可能な法律および規制によって決定されます。AWS のサービス を使用する際のコンプライアンス責任の詳細については、「AWS セキュリティドキュメント」を参照してください。

VPC とサブネットへのパブリックアクセスをブロックする

VPC Block Public Access (BPA) は、AWS アカウント全体にわたって VPC リソースへのパブリックインターネットアクセスを防止する一元的なセキュリティ機能です。これにより、セキュリティ要件への準拠が保証されるとともに特定の例外や監査機能に対する柔軟性が提供されます。

VPC BPA 機能には次のモードがあります:

- [双方向]: このリージョンのインターネットゲートウェイとエグレスのみのインターネットゲートウェイとの間のすべてのトラフィック (除外された VPC とサブネットを除く) がブロックされます。
- [イングレスのみ]: このリージョンの VPC に対するすべてのインターネットトラフィック (除外される VPC またはサブネットを除く) がブロックされます。NAT ゲートウェイとエグレスのみのインターネットゲートウェイとの間のトラフィックのみが許可されます。なぜなら、これらのゲートウェイはアウトバウンド接続の確立のみを許可するからです。

また、ブロックしないトラフィックのために、この機能で「除外」を作成することもできます。除外は、単一の VPC またはサブネットに適用できるモードであり、アカウントの VPC BPA モードから除外し、双方向または Egress-Only のアクセスを許可します。

除外では、次のいずれかのモードを使用できます:

- [双方向]: 除外された VPC とサブネットとの間のすべてのインターネットトラフィックが許可されます。
- [エグレスのみ]: 除外された VPCとサブネットからの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ます。除外された VPC とサブネットに対するインバウンドインターネットトラフィックはブロックされます。これは、VPC BPA が [双方向] に設定されている場合にのみ適用されます。

内容

- VPC BPA の基本
- VPC BPA の影響を評価し、VPC BPA をモニタリングする

- 高度な例

VPC BPA の基本

このセクションでは、VPC BPA をサポートするサービスや、VPC BPA の使用方法など、VPC BPA に関する重要な詳細について説明します。

内容

- リージョン別の可用性
- AWS サービスへの影響とサポート
- VPC BPA の制限事項
- IAM ポリシーを使用して VPC BPA に対するアクセスを制御する
- アカウントのために VPC BPA 双方向モードを有効にする
- VPC BPA モードをイングレスのみに変更する
- 除外を作成および削除する
- 組織レベルで VPC BPA を有効にする

リージョン別の可用性

VPC BPA は、GovCloud および中国リージョンを含むすべての商用 AWS リージョンで利用できます。

また、このガイドでは、Network Access Analyzer および Reachability Analyzer と VPC BPA の併用についても説明します。Network Access Analyzer と Reachability Analyzer は、すべての商用リージョンで利用できるわけではありません。Network Access Analyzer と Reachability Analyzer を利用できるリージョンについては、「Network Access Analyzer ガイド」の「Limitations」と「Reachability Analyzer ガイド」の「Considerations」を参照してください。

AWS サービスへの影響とサポート

次のリソースとサービスは VPC BPA をサポートし、これらのサービスとリソースに対するトラフィックは VPC BPA の影響を受けます。

- [インターネットゲートウェイ]: 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ブロックされます。

- [エグレスのみのインターネットゲートウェイ]: すべてのアウトバウンドトラフィックがブロックされます。エグレスのみのインターネットゲートウェイは、インバウンドトラフィックを許可しません。
- Gateway Load Balancer (GWLB): GWLB エンドポイントを含むサブネットが除外されていても、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はすべてブロックされます。
- [NAT ゲートウェイ]: 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ブロックされます。NAT ゲートウェイには、インターネット接続のためのインターネットゲートウェイが必要です。
- [インターネット向け Network Load Balancer]: 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ブロックされます。インターネット向け Network Load Balancer には、インターネット接続のためのインターネットゲートウェイが必要です。
- [インターネット向け Application Load Balancer]: 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ブロックされます。インターネット向け Application Load Balancer には、インターネット接続のためのインターネットゲートウェイが必要です。
- Amazon CloudFront VPC オリジン: 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ブロックされます。
- Direct Connect: パブリック仮想インターフェイス (パブリック IPv4 またはグローバルユニキャスト IPv6 アドレス) を使用する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はブロックされます。このトラフィックは、接続にインターネットゲートウェイ (または Egress-Only インターネットゲートウェイ) を使用します。
- AWS Global Accelerator: ターゲットがインターネットからアクセス可能かどうかにかかわらず、VPC へのインバウンドトラフィックはブロックされます。
- AWS Network Firewall: ファイアウォールエンドポイントを含むサブネットが除外されていない限り、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はすべてブロックされます。
- AWS Wavelength キャリア ゲートウェイ: 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ブロックされます。

次のサービスやリソースについてのトラフィックなど、プライベート接続に関連するトラフィックは、VPC BPA によってブロックされず、影響も受けません。

- AWS Client VPN
- AWS CloudWAN
- AWS Outposts ローカルゲートウェイ
- AWS Site-to-Site VPN

- トランジットゲートウェイ
- AWS Verified Access

> **Important**
>
> - VPC Ingress Routing を使用してサブネット内の EC2 インスタンスで実行されるアプライアンス (サードパーティーのセキュリティツールやモニタリングツールなど) を介して送受信トラフィックをルーティングする場合は、VPC Block Public Access からワークロードサブネットを除外する必要があります。ワークロードサブネットは、トラフィックがアプライアンスを介してルーティングされるインスタンスを含むサブネットです。アプライアンスを含むサブネットを除外する必要はありません。
> - VPC 内のリソースから Route 53 Resolver など、VPC で実行されている他のサービスにプライベートに送信されるトラフィックは、VPC 内のインターネットゲートウェイを通過しないため、VPC BPA がオンになっている場合でも許可されます。これらのサービスは、ユーザーに代わって VPC 外のリソースに対してリクエストを実行する可能性があります (DNS クエリを解決するなど)。他のセキュリティコントロールを通じて緩和されない場合、VPC 内のリソースのアクティビティに関する情報が公開される可能性があります。
> - インターネット向けロードバランサーがあり、そのサブネットの 1 つのみに対して VPC BPA 除外を作成した場合、ロードバランサーは引き続き除外されたサブネットでパブリックトラフィックを受信し、除外されていないサブネットのターゲットにプライベートにルーティングすることができます。VPC BPA がターゲットへのパブリックアクセスを完全にブロックするには、ロードバランサーサブネットが除外されていないことを確認してください。

VPC BPA の制限事項

VPC BPA のイングレスのみのモードは、NAT ゲートウェイとエグレスのみのインターネットゲートウェイが許可されていないローカルゾーン (LZ) ではサポートされていません。

IAM ポリシーを使用して VPC BPA に対するアクセスを制御する

VPC BPA 機能に対するアクセスを許可/拒否する IAM ポリシーの例については、「VPC とサブネットへのパブリックアクセスをブロックする」を参照してください。

アカウントのために VPC BPA 双方向モードを有効にする

VPC BPA 双方向モードは、このリージョンのインターネットゲートウェイとエグレスのみのインターネットゲートウェイとの間のすべてのトラフィックをブロックします (除外された VPC とサブネットを除く)。除外の詳細については、「除外を作成および削除する」を参照してください。

> **Important**
>
> 本番アカウントで VPC BPA を有効にする前に、インターネットアクセスを必要とするワークロードを徹底的に確認することを強くお勧めします。

> **Note**
>
> - アカウントの VPC とサブネットで VPC BPA を有効にするには、その VPC とサブネットを所有している必要があります。
> - 現在 VPC サブネットを他のアカウントと共有している場合、サブネット所有者によって強制適用される VPC BPA モードも参加者のトラフィックに適用されますが、参加者は共有サブネットに影響を及ぼす VPC BPA の設定を制御することはできません。

AWS マネジメントコンソール

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 左のナビゲーションペインで、[設定] を選択します。
3. [パブリックアクセスの設定を編集] を選択します。
4. [ブロックパブリックアクセスをオンにする] と [双方向] を選択し、[変更を保存] を選択します。
5. [ステータス] が [オン] に変わるまで待ちます。VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。

VPC BPA の [双方向] モードがオンになりました。

AWS CLI

1. VPC BPA をオンにします。

```
aws ec2 --region us-east-2 modify-vpc-block-public-access-options --internet-
gateway-block-mode block-bidirectional
```

VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。

2. VPC BPA のステータスを表示します。

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-options
```

VPC BPA モードをイングレスのみに変更する

VPC BPA のイングレスのみのモードは、このリージョンの VPC に対するすべてのインターネットトラフィックをブロックします (除外される VPC またはサブネットを除く)。NAT ゲートウェイとエグレスのみのインターネットゲートウェイとの間のトラフィックのみが許可されます。なぜなら、これらのゲートウェイはアウトバウンド接続の確立のみを許可するからです。

AWS マネジメントコンソール

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 左のナビゲーションペインで、[設定] を選択します。
3. [パブリックアクセスの設定を編集] を選択します。
4. 方向を Ingress-only に変更します。
5. 変更を保存し、ステータスが更新されるまで待ちます。VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。

AWS CLI

1. VPC BPA ブロックの方向を変更します:

```
aws ec2 --region us-east-2 modify-vpc-block-public-access-options --internet-
gateway-block-mode block-ingress
```

VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。

2. VPC BPA のステータスを表示します。

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-options
```

除外を作成および削除する

VPC BPA の除外は、単一の VPC またはサブネットに適用できるモードです。有効にした場合、対象の VPC またはサブネットがアカウントの VPC BPA モードから除外され、双方向または Egress-Only のアクセスが許可されます。アカウントで VPC BPA が有効になっていない場合でも VPC とサブネットのために VPC BPA の除外を作成して、VPC BPA がオンになっているときに除外に対するトラフィックの中断が発生しないようにできます。VPC の除外は、VPC 内のすべてのサブネットに自動的に適用されます。

最大 50 個の除外を作成できます。制限の引き上げをリクエストする方法については、「Amazon VPC クォータ」の「VPC BPA exclusions per account」を参照してください。

AWS マネジメントコンソール

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 左のナビゲーションペインで、[設定] を選択します。
3. [パブリックアクセスをブロックする] タブの [除外] で、次のいずれかを実行します。
   - 除外を削除するには、削除する除外項目を選択し、[アクション] > [除外の削除] を選択します。
   - 除外を作成するには、[除外を作成] を選択し、次のステップに進みます。
4. ブロック方向を選択します。
   - [双方向]: 除外された VPC とサブネットとの間のすべてのインターネットトラフィックを許可します。
   - [エグレスのみ]: 除外された VPCとサブネットからのアウトバウンドインターネットトラフィックを許可します。除外された VPC とサブネットに対するインバウンドインターネットトラフィックをブロックします。この設定は、VPC BPA が [双方向] に設定されている場合に適用されます。
5. [VPC] または [サブネット] を選択します。
6. [除外を作成] を選択します。
7. [除外ステータス] が [アクティブ] に変わるまで待ちます。変更を確認するには、除外テーブルを更新する必要がある場合があります。

除外が作成されました。

AWS CLI

1. 除外の許可の方向を変更します:

```
aws ec2 --region us-east-2 create-vpc-block-public-access-exclusion --subnet-
id subnet-id --internet-gateway-exclusion-mode allow-bidirectional
```

2. 除外ステータスが更新されるまでに時間がかかる場合があります。除外のステータスを表示するには:

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-exclusions --
exclusion-ids exclusion-id
```

組織レベルで VPC BPA を有効にする

AWS Organizations を使用して組織内のアカウントを管理している場合は、AWSOrganizations 宣言型ポリシーを使用して、組織内のアカウントに VPC BPA を適用できます。VPC BPA 宣言型ポリシーの詳細については、「AWS Organizations ユーザーガイド」の「サポートされている宣言型ポリシー」を参照してください。

> **Note**
>
> - VPC BPA 宣言型ポリシーを使用して、除外を許可するかどうかを設定できますが、ポリシーを使用して除外を作成することはできません。除外を作成するには、VPC を所有するアカウントで除外を作成する必要があります。VPC BPA の除外の作成方法の詳細については、「除外を作成および削除する」を参照してください。
> - VPC BPA 宣言型ポリシーが有効になっている場合、[パブリックアクセスをブロックする] では、[宣言型ポリシーによる管理] と表示され、アカウントレベルで VPC BPA 設定を変更することはできません。

VPC BPA の影響を評価し、VPC BPA をモニタリングする

このセクションには、VPC BPA をオンにする前に VPC BPA の影響を評価する方法に関する情報と、VPC BPA をオンにした後にトラフィックがブロックされるかどうかをモニタリングする方法に関する情報が含まれています。

内容

- Network Access Analyzer を使用して VPC BPA の影響を評価する
- フローログを使用して VPC BPA の影響をモニタリングする
- CloudTrail を使用して除外の削除を追跡する
- Reachability Analyzer を使用して接続がブロックされていることを検証する

Network Access Analyzer を使用して VPC BPA の影響を評価する

このセクションでは、VPC BPA を有効にしてアクセスをブロックする前に、Network Access Analyzer を使用して、インターネットゲートウェイを使用するアカウントのリソースを表示します。この分析を使用して、アカウントで VPC BPA をオンにし、トラフィックをブロックした場合の影響を理解します。

> **Note**
>
> - Network Access Analyzer は IPv6 をサポートしていないため、Egress-Only インターネットゲートウェイのアウトバウンド IPv6 トラフィックに対する VPC BPA の潜在的な影響を表示するために使用することはできません。
> - Network Access Analyzer で実行する分析には料金がかかります。詳細については、「Network Access Analyzer ガイド」の「料金」を参照してください。
> - Network Access Analyzer が利用できるリージョンについては、「Network Access Analyzer ガイド」の「Limitations」を参照してください。

AWS マネジメントコンソール

1. https://console.aws.amazon.com/networkinsights/ で AWS Network Insights コンソールを開きます。
2. [Network Access Analyzer] を選択します。
3. [ネットワークアクセススコープを作成] を選択します。
4. [Assess impact of VPC Block Public Access] を選択し、[次へ]を選択します。
5. テンプレートは、アカウントのインターネットゲートウェイとの間のトラフィックを分析するように既に設定されています。これは、[ソース] と [宛先] で確認できます。
6. [次へ] を選択します。
7. [ネットワークアクセススコープを作成] を選択します。

8. 先ほど作成したスコープを選択し、[分析] を選択します。
9. 分析が完了するまで待ちます。
10. 分析の検出結果を表示します。[検出結果] の各行には、アカウントのインターネットゲートウェイとの間のネットワーク内でパケットが沿うことができるネットワークパスが表示されます。この場合、VPC BPA をオンにし、これらの検出結果に表示される VPC やサブネットのいずれも VPC BPA の除外として設定されていない場合、それらの VPC やサブネットに対するトラフィックは制限されます。
11. 各検出結果を分析して、VPC 内のリソースに対する VPC BPA の影響を理解します。

影響分析が完了しました。

AWS CLI

1. ネットワークアクセススコープを作成します:

```
aws ec2 create-network-insights-access-scope --region us-east-2 --match-paths
 "Source={ResourceStatement={ResourceTypes=["AWS::EC2::InternetGateway"]}}"
 "Destination={ResourceStatement={ResourceTypes=["AWS::EC2::InternetGateway"]}}"
```

2. スコープ分析を開始します:

```
aws ec2 start-network-insights-access-scope-analysis  --region us-east-2 --
network-insights-access-scope-id nis-id
```

3. 分析の結果を取得します:

```
aws ec2 get-network-insights-access-scope-analysis-findings  --region us-east-2
 --network-insights-access-scope-analysis-id nisa-0aa383a1938f94cd1 --max-items
 1
```

その結果、アカウント内のすべての VPC のインターネットゲートウェイとの間のトラフィックが表示されます。結果は「検出結果」として整理されます。"FindingId": "AnalysisFinding-1" は、これが分析の最初の結果であることを示します。複数の検出結果があり、それぞれが VPC BPA をオンにすることで影響を受けるトラフィックフローを示しています。最初の検出結果は、トラフィックがインターネットゲートウェイ ("SequenceNumber": 1) で開始され、NACL ("SequenceNumber": 2)、セキュリティグループ ("SequenceNumber": 3) の順に渡され、インスタンス ("SequenceNumber": 4) で終了したことを示しています。

4. 検出結果を分析して、VPC 内のリソースに対する VPC BPA の影響を理解します。

影響分析が完了しました。

フローログを使用して VPC BPA の影響をモニタリングする

VPC フローログは、VPC の Elastic Network Interface 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。この機能を使用して、インスタンスのネットワークインターフェイスに到達しないように VPC BPA によってブロックされたトラフィックをモニタリングできます。

フローログの使用 のステップを使用して、VPC のフローログを作成します。

フローログを作成する際には、フィールド `reject-reason` を含むカスタム形式を使用してください。

フローログを表示する際に、VPC BPA が原因で ENI に対するトラフィックが拒否された場合、フローログエントリに BPA の `reject-reason` が表示されます。

VPC フローログについての標準の制限に加えて、VPC BPA に固有の次の制限に留意してください。

- VPC BPA のフローログには、スキップされたレコードは含まれません。
- VPC BPA のフローログには、フローログに `bytes` フィールドを含めた場合であっても、`bytes` は含まれません。

CloudTrail を使用して除外の削除を追跡する

このセクションでは、AWS CloudTrail を使用して VPC BPA の除外の削除をモニタリングおよび追跡する方法について説明します。

AWS マネジメントコンソール

https://console.aws.amazon.com/cloudtrailv2/ の AWS CloudTrail コンソールの [リソースタイプ] > `AWS::EC2::VPCBlockPublicAccessExclusion` にアクセスして、[CloudTrail イベント履歴] で、削除された除外を確認できます。

AWS CLI

除外の削除に関連するイベントを表示するには、`lookup-events` コマンドを使用します:

```
aws cloudtrail lookup-events --lookup-attributes
 AttributeKey=ResourceType,AttributeValue=AWS::EC2::VPCBlockPublicAccessExclusion
```

Reachability Analyzer を使用して接続がブロックされていることを検証する

VPC Reachability Analyzer を使用して、VPC BPA の設定を含むネットワーク設定を踏まえて、特定のネットワークパスに到達できるかどうかを評価できます。

Reachability Analyzer を利用できるリージョンについては、「Reachability Analyzer ガイド」の「Considerations」を参照してください。

AWS マネジメントコンソール

1. https://console.aws.amazon.com/networkinsights/home#ReachabilityAnalyzer で AWS Network Insights コンソールを開きます。
2. [パスの作成と分析] を選択します
3. [ソースタイプ] で、[インターネットゲートウェイ] を選択し、[ソースドロップダウン] からトラフィックをブロックするインターネットゲートウェイを選択します。
4. [宛先タイプ] で、[インスタンス] を選択し、[宛先] ドロップダウンからトラフィックをブロックするインスタンスを選択します。
5. [パスの作成と分析] を選択します
6. 分析が完了するまで待ちます。数分かかる場合があります。
7. 完了すると、[可達のステータス] が [到達不可能] となり、[パスの詳細] に VPC_BLOCK_PUBLIC_ACCESS_ENABLED がこの到達可能性に関する問題の原因であることが示されます。

AWS CLI

1. トラフィックをブロックするインターネットゲートウェイの ID (ソース) と、トラフィックをブロックするインスタンスの ID (宛先) を使用してネットワークパスを作成します。

   ```
   aws ec2 --region us-east-2 create-network-insights-path --source igw-id --
   destination instance-id --protocol TCP
   ```

2. ネットワークパスで分析を開始します:

```
aws ec2 --region us-east-2 start-network-insights-analysis --network-insights-
path-id nip-id
```

3. 分析の結果を取得します:

```
aws ec2 --region us-east-2 describe-network-insights-analyses --network-
insights-analysis-ids nia-id
```

4. 到達可能性の欠如について、VPC_BLOCK_PUBLIC_ACCESS_ENABLED が ExplanationCode であることを確認します。

高度な例

このセクションでは、VPC ブロックパブリックアクセス機能がさまざまなシナリオでどのように機能するかを理解するのに役立つ高度な例を示します。各シナリオはその前のシナリオに依拠するため、ステップを順番に完了することが重要です。

> **Important**
>
> 本番アカウントでは、この例を実行しないでください。本番アカウントで VPC BPA を有効にする前に、インターネットアクセスを必要とするワークロードを徹底的に確認することを強くお勧めします。

> **Note**
>
> VPC BPA の機能を完全に理解するには、アカウントに特定のリソースが必要です。このセクションでは、この機能の仕組みを完全に理解するために必要なリソースをプロビジョニングするために使用できる CloudFormation テンプレートを提供します。CloudFormation テンプレートを使用してプロビジョニングするリソースと、Network Access Analyzer と Reachability Analyzer を使用して実行する分析にはコストがかかります。このセクションのテンプレートを使用する場合は、この例を完了したら、クリーンアップのステップを完了してください。

内容

- CloudFormation テンプレートをデプロイする (オプション)

- Network Access Analyzer を使用して VPC BPA の影響を表示する
- シナリオ 1 - VPC BPA が有効になっていないインスタンスに接続する
- シナリオ 2 - 双方向モードで VPC BPA を有効にする
- シナリオ 3 - VPC BPA を Ingress-Only モードに変更する
- シナリオ 4 - 除外を作成する
- シナリオ 5 - 除外モードを変更する
- シナリオ 6 - VPC BPA モードを変更する
- クリーンアップ

CloudFormation テンプレートをデプロイする (オプション)

この機能の仕組みのデモには、VPC、サブネット、インスタンス、および他のリソースが必要です。このデモをより簡単に完了できるように、このデモのシナリオのために必要なリソースを迅速にスピンアップするために使用できる CloudFormation テンプレートが以下に提供されています。このステップはオプションであり、このセクションのシナリオの図のみを表示できます。

> **Note**
>
> - NAT ゲートウェイやパブリック IPv4 アドレスのコストなど、CloudFormation テンプレートを使用してこのセクションで作成するリソースに関連するコストがかかります。余分なコストがかからないよう、クリーンアップのステップを完了して、この例のために作成されたすべてのリソースを削除してください。
> - この CloudFormation テンプレートは、VPC BPA に必要な基盤となるリソースを作成しますが、VPC BPA 機能自体は有効にしません。ここでデプロイされるリソースは、VPC BPA 機能を個別に有効にした後、VPC BPA 機能を理解しテストするのに役立ちます。

このテンプレートによって以下のリソースがアカウントに作成されます。

- Egress-only 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
- NAT ゲートウェイ
- 2 つのパブリックサブネット
- 1 つのプライベートサブネット

- パブリックおよびプライベート IPv4 アドレスを持つ 2 つの EC2 インスタンス
- IPv6 アドレスとプライベート IPv4 アドレスを持つ 1 つの EC2 インスタンス
- プライベート IPv4 アドレスのみを持つ 1 つの EC2 インスタンス
- SSH および ICMP インバウンドトラフィックが許可され、すべてのアウトバウンドトラフィックが許可されているセキュリティグループ
- VPC フローログ
- サブネット B の 1 つの EC2 Instance Connect エンドポイント

以下のテンプレートをコピーし、.yaml ファイルに保存します。

```
AWSTemplateFormatVersion: '2010-09-09'
Description: Creates a VPC with public and private subnets, NAT gateway, and EC2
 instances for VPC BPA.

Parameters:
  InstanceAMI:
    Description: ID of the Amazone Machine Image (AMI) to use with the instances
 launched by this template
    Type: AWS::EC2::Image::Id
  InstanceType:
    Description: EC2 Instance type to use with the instances launched by this template
    Type: String
    Default: t2.micro

Resources:

  # VPC
  VPCBPA:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: 10.0.0.0/16
      EnableDnsHostnames: true
      EnableDnsSupport: true
      InstanceTenancy: default
      Tags:
        - Key: Name
          Value: VPC BPA

  # VPC IPv6 CIDR
  VPCBPAIpv6CidrBlock:
```

```
    Type: AWS::EC2::VPCCidrBlock
    Properties:
      VpcId: !Ref VPCBPA
      AmazonProvidedIpv6CidrBlock: true

  # EC2 Key Pair
  VPCBPAKeyPair:
    Type: AWS::EC2::KeyPair
    Properties:
      KeyName: vpc-bpa-key

  # Internet Gateway
  VPCBPAInternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: VPC BPA Internet Gateway

  VPCBPAInternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId: !Ref VPCBPA
      InternetGatewayId: !Ref VPCBPAInternetGateway

  # Egress-Only Internet Gateway
  VPCBPAEgressOnlyInternetGateway:
    Type: AWS::EC2::EgressOnlyInternetGateway
    Properties:
      VpcId: !Ref VPCBPA

  # Subnets
  VPCBPAPublicSubnetA:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPCBPA
      CidrBlock: 10.0.1.0/24
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: VPC BPA Public Subnet A

  VPCBPAPublicSubnetB:
    Type: AWS::EC2::Subnet
```

```
    Properties:
      VpcId: !Ref VPCBPA
      CidrBlock: 10.0.2.0/24
      MapPublicIpOnLaunch: true
      Tags:
        - Key: Name
          Value: VPC BPA Public Subnet B

  VPCBPAPrivateSubnetC:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPCBPA
      CidrBlock: 10.0.3.0/24
      MapPublicIpOnLaunch: false
      Ipv6CidrBlock: !Select [0, !GetAtt VPCBPA.Ipv6CidrBlocks]
      AssignIpv6AddressOnCreation: true
      Tags:
        - Key: Name
          Value: VPC BPA Private Subnet C

  # NAT Gateway
  VPCBPANATGateway:
    Type: AWS::EC2::NatGateway
    Properties:
      AllocationId: !GetAtt VPCBPANATGatewayEIP.AllocationId
      SubnetId: !Ref VPCBPAPublicSubnetB
      Tags:
        - Key: Name
          Value: VPC BPA NAT Gateway

  VPCBPANATGatewayEIP:
    Type: AWS::EC2::EIP
    Properties:
      Domain: vpc
      Tags:
        - Key: Name
          Value: VPC BPA NAT Gateway EIP

  # Route Tables
  VPCBPAPublicRouteTable:
    Type: AWS::EC2::RouteTable
    Properties:
      VpcId: !Ref VPCBPA
      Tags:
```

```
      - Key: Name
        Value: VPC BPA Public Route Table

VPCBPAPublicRoute:
  Type: AWS::EC2::Route
  DependsOn: VPCBPAInternetGatewayAttachment
  Properties:
    RouteTableId: !Ref VPCBPAPublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref VPCBPAInternetGateway

VPCBPAPublicSubnetARouteTableAssoc:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref VPCBPAPublicSubnetA
    RouteTableId: !Ref VPCBPAPublicRouteTable

VPCBPAPublicSubnetBRouteTableAssoc:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    SubnetId: !Ref VPCBPAPublicSubnetB
    RouteTableId: !Ref VPCBPAPublicRouteTable

VPCBPAPrivateRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPCBPA
    Tags:
      - Key: Name
        Value: VPC BPA Private Route Table

VPCBPAPrivateRoute:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref VPCBPAPrivateRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref VPCBPANATGateway

VPCBPAPrivateSubnetCRoute:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref VPCBPAPrivateRouteTable
    DestinationIpv6CidrBlock: ::/0
    EgressOnlyInternetGatewayId: !Ref VPCBPAEgressOnlyInternetGateway
```

```
  VPCBPAPrivateSubnetCRouteTableAssociation:
    Type: AWS::EC2::SubnetRouteTableAssociation
    Properties:
      SubnetId: !Ref VPCBPAPrivateSubnetC
      RouteTableId: !Ref VPCBPAPrivateRouteTable

  # EC2 Instances Security Group
  VPCBPAInstancesSecurityGroup:
    Type: AWS::EC2::SecurityGroup
    Properties:
      GroupName: VPC BPA Instances Security Group
      GroupDescription: Allow SSH and ICMP access
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0
        - IpProtocol: icmp
          FromPort: -1
          ToPort: -1
          CidrIp: 0.0.0.0/0
      VpcId: !Ref VPCBPA
      Tags:
        - Key: Name
          Value: VPC BPA Instances Security Group

  # EC2 Instances
  VPCBPAInstanceA:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref InstanceAMI
      InstanceType: t2.micro
      KeyName: !Ref VPCBPAKeyPair
      SubnetId: !Ref VPCBPAPublicSubnetA
      SecurityGroupIds:
        - !Ref VPCBPAInstancesSecurityGroup
      Tags:
        - Key: Name
          Value: VPC BPA Instance A

  VPCBPAInstanceB:
    Type: AWS::EC2::Instance
    Properties:
```

```
      ImageId: !Ref InstanceAMI
      InstanceType: !Ref InstanceType
      KeyName: !Ref VPCBPAKeyPair
      SubnetId: !Ref VPCBPAPublicSubnetB
      SecurityGroupIds:
        - !Ref VPCBPAInstancesSecurityGroup
      Tags:
        - Key: Name
          Value: VPC BPA Instance B

  VPCBPAInstanceC:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref InstanceAMI
      InstanceType: !Ref InstanceType
      KeyName: !Ref VPCBPAKeyPair
      SubnetId: !Ref VPCBPAPrivateSubnetC
      SecurityGroupIds:
        - !Ref VPCBPAInstancesSecurityGroup
      Tags:
        - Key: Name
          Value: VPC BPA Instance C

  VPCBPAInstanceD:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: !Ref InstanceAMI
      InstanceType: !Ref InstanceType
      KeyName: !Ref VPCBPAKeyPair
      NetworkInterfaces:
        - DeviceIndex: '0'
          GroupSet:
            - !Ref VPCBPAInstancesSecurityGroup
          SubnetId: !Ref VPCBPAPrivateSubnetC
          Ipv6AddressCount: 1
      Tags:
        - Key: Name
          Value: VPC BPA Instance D

  # Flow Logs IAM Role
  VPCBPAFlowLogRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
```

```
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service: vpc-flow-logs.amazonaws.com
            Action: 'sts:AssumeRole'
      Tags:
        - Key: Name
          Value: VPC BPA Flow Logs Role

  VPCBPAFlowLogPolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyName: VPC-BPA-FlowLogsPolicy
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Action:
              - 'logs:CreateLogGroup'
              - 'logs:CreateLogStream'
              - 'logs:PutLogEvents'
              - 'logs:DescribeLogGroups'
              - 'logs:DescribeLogStreams'
            Resource: '*'
      Roles:
        - !Ref VPCBPAFlowLogRole

  # Flow Logs
  VPCBPAFlowLog:
    Type: AWS::EC2::FlowLog
    Properties:
      ResourceId: !Ref VPCBPA
      ResourceType: VPC
      TrafficType: ALL
      LogDestinationType: cloud-watch-logs
      LogGroupName: /aws/vpc-flow-logs/VPC-BPA
      DeliverLogsPermissionArn: !GetAtt VPCBPAFlowLogRole.Arn
      LogFormat: '${version} ${account-id} ${interface-id} ${srcaddr} ${dstaddr}
 ${srcport} ${dstport} ${protocol} ${packets} ${bytes} ${start} ${end} ${action} ${log-
status} ${vpc-id} ${subnet-id} ${instance-id} ${tcp-flags} ${type} ${pkt-srcaddr}
 ${pkt-dstaddr} ${region} ${az-id} ${sublocation-type} ${sublocation-id} ${pkt-src-aws-
service} ${pkt-dst-aws-service} ${flow-direction} ${traffic-path} ${reject-reason}'
      Tags:
```

```
        - Key: Name
          Value: VPC BPA Flow Logs

  # EC2 Instance Connect Endpoint
  VPCBPAEC2InstanceConnectEndpoint:
    Type: AWS::EC2::InstanceConnectEndpoint
    Properties:
      SecurityGroupIds:
        - !Ref VPCBPAInstancesSecurityGroup
      SubnetId: !Ref VPCBPAPublicSubnetB

Outputs:
  VPCBPAVPCId:
    Description: A reference to the created VPC
    Value: !Ref VPCBPA
    Export:
      Name: vpc-id

  VPCBPAPublicSubnetAId:
    Description: The ID of the public subnet A
    Value: !Ref VPCBPAPublicSubnetA

  VPCBPAPublicSubnetAName:
    Description: The name of the public subnet A
    Value: VPC BPA Public Subnet A

  VPCBPAPublicSubnetBId:
    Description: The ID of the public subnet B
    Value: !Ref VPCBPAPublicSubnetB

  VPCBPAPublicSubnetBName:
    Description: The name of the public subnet B
    Value: VPC BPA Public Subnet B

  VPCBPAPrivateSubnetCId:
    Description: The ID of the private subnet C
    Value: !Ref VPCBPAPrivateSubnetC

  VPCBPAPrivateSubnetCName:
    Description: The name of the private subnet C
    Value: VPC BPA Private Subnet C

  VPCBPAInstanceAId:
    Description: The ID of instance A
```

```
    Value: !Ref VPCBPAInstanceA

  VPCBPAInstanceBId:
    Description: The ID of instance B
    Value: !Ref VPCBPAInstanceB

  VPCBPAInstanceCId:
    Description: The ID of instance C
    Value: !Ref VPCBPAInstanceC

  VPCBPAInstanceDId:
    Description: The ID of instance D
    Value: !Ref VPCBPAInstanceD
```

AWS マネジメントコンソール

1. で CloudFormation コンソールを開きます。。https://console.aws.amazon.com/cloudformation/
2. [スタックを作成] を選択し、.yaml テンプレートファイルをアップロードします。
3. テンプレートを起動するステップを実行します。[イメージ ID] と [インスタンスタイプ] (t2.micro など) を入力する必要があります。また、フローログの作成と CloudWatch へのログ記録の許可のために、CloudFormation が IAM ロールを作成することを許可する必要があります。
4. スタックを起動したら、[イベント] タブで進行状況を表示し、続行する前にスタックが完了していることを確認します。

AWS CLI

1. CloudFormation スタックを作成するには、次のコマンドを実行します:

```
aws cloudformation create-stack --stack-name VPC-BPA-stack --template-body
 file://sampletemplate.yaml --capabilities CAPABILITY_IAM --region us-east-2
```

出力:

```
{
    "StackId": "arn:aws:cloudformation:us-east-2:470889052923:stack/VPC-BPA-
stack/8a7a2cc0-8001-11ef-b196-06386a84b72f"
}
```

2. 進行状況を表示し、続行する前にスタックが完了しているようにします:

```
aws cloudformation describe-stack-events --stack-name VPC-BPA-stack --region us-east-2
```

Network Access Analyzer を使用して VPC BPA の影響を表示する

このセクションでは、Network Access Analyzer を使用して、インターネットゲートウェイを使用するアカウントのリソースを表示します。この分析を使用して、アカウントで VPC BPA をオンにし、トラフィックをブロックした場合の影響を理解します。

Network Access Analyzer が利用できるリージョンについては、「Network Access Analyzer ガイド」の「Limitations」を参照してください。

AWS マネジメントコンソール

1. https://console.aws.amazon.com/networkinsights/ で AWS Network Insights コンソールを開きます。
2. [Network Access Analyzer] を選択します。
3. [ネットワークアクセススコープを作成] を選択します。
4. [Assess impact of VPC Block Public Access] を選択し、[次へ]を選択します。
5. テンプレートは、アカウントのインターネットゲートウェイとの間のトラフィックを分析するように既に設定されています。これは、[ソース] と [宛先] で確認できます。
6. [次へ] を選択します。
7. [ネットワークアクセススコープを作成] を選択します。
8. 先ほど作成したスコープを選択し、[分析] を選択します。
9. 分析が完了するまで待ちます。
10. 分析の検出結果を表示します。[検出結果] の各行には、アカウントのインターネットゲートウェイとの間のネットワーク内でパケットが沿うことができるネットワークパスが表示されます。この場合、VPC BPA をオンにし、これらの検出結果に表示される VPC やサブネットのいずれも VPC BPA の除外として設定されていない場合、それらの VPC やサブネットに対するトラフィックは制限されます。
11. 各検出結果を分析して、VPC 内のリソースに対する VPC BPA の影響を理解します。

影響分析が完了しました。

AWS CLI

1. ネットワークアクセススコープを作成します:

```
aws ec2 create-network-insights-access-scope --match-paths
 "Source={ResourceStatement={ResourceTypes=["AWS::EC2::InternetGateway"]}}"
 "Destination={ResourceStatement={ResourceTypes=["AWS::EC2::InternetGateway"]}}"
 --region us-east-2
```

出力:

```
{
  "NetworkInsightsAccessScope": {
    "NetworkInsightsAccessScopeId": "nis-04cad3c4b3a1d5e3e",
    "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-2:470889052923:network-insights-access-scope/nis-04cad3c4b3a1d5e3e",
    "CreatedDate": "2024-09-30T15:55:53.171000+00:00",
    "UpdatedDate": "2024-09-30T15:55:53.171000+00:00"
  },
  "NetworkInsightsAccessScopeContent": {
    "NetworkInsightsAccessScopeId": "nis-04cad3c4b3a1d5e3e",
    "MatchPaths": [
      {
        "Source": {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::EC2::InternetGateway"
            ]
          }
        }
      },
      {
        "Destination": {
          "ResourceStatement": {
            "ResourceTypes": [
              "AWS::EC2::InternetGateway"
            ]
          }
        }
      }
    ]
  }
```

```
}
```

2. スコープ分析を開始します:

```
aws ec2 start-network-insights-access-scope-analysis --network-insights-access-
scope-id nis-04cad3c4b3a1d5e3e --region us-east-2
```

出力:

```
{
  "NetworkInsightsAccessScopeAnalysis": {
    "NetworkInsightsAccessScopeAnalysisId": "nisa-0aa383a1938f94cd1",
    "NetworkInsightsAccessScopeAnalysisArn": "arn:aws:ec2:us-
east-2:470889052923:network-insights-access-scope-analysis/
nisa-0aa383a1938f94cd",
    "NetworkInsightsAccessScopeId": "nis-04cad3c4b3a1d5e3e",
    "Status": "running",
    "StartDate": "2024-09-30T15:56:59.109000+00:00",
    "AnalyzedEniCount": 0
  }
}
```

3. 分析の結果を取得します:

```
aws ec2 get-network-insights-access-scope-analysis-findings --network-insights-
access-scope-analysis-id nisa-0aa383a1938f94cd1 --region us-east-2 --max-items 1
```

出力:

```
{
  "AnalysisFindings": [
    {
      "NetworkInsightsAccessScopeAnalysisId": "nisa-0aa383a1938f94cd1",
      "NetworkInsightsAccessScopeId": "nis-04cad3c4b3a1d5e3e",
      "FindingId": "AnalysisFinding-1",
      "FindingComponents": [
        {
          "SequenceNumber": 1,
          "Component": {
            "Id": "igw-04a5344b4e30486f1",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:internet-gateway/
igw-04a5344b4e30486f1",
```

```
            "Name": "VPC BPA Internet Gateway"
          },
          "OutboundHeader": {
            "DestinationAddresses": [
              "10.0.1.85/32"
            ]
          },
          "InboundHeader": {
            "DestinationAddresses": [
              "10.0.1.85/32"
            ],
            "DestinationPortRanges": [
              {
                "From": 22,
                "To": 22
              }
            ],
            "Protocol": "6",
            "SourceAddresses": [
              "0.0.0.0/5",
              "100.0.0.0/10",
              "96.0.0.0/6"
            ],
            "SourcePortRanges": [
              {
                "From": 0,
                "To": 65535
              }
            ]
          },
          "Vpc": {
            "Id": "vpc-0762547ec48b6888d",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:vpc/
vpc-0762547ec48b6888d",
            "Name": "VPC BPA"
          }
        },
        {
          "SequenceNumber": 2,
          "AclRule": {
            "Cidr": "0.0.0.0/0",
            "Egress": false,
            "Protocol": "all",
            "RuleAction": "allow",
```

```
            "RuleNumber": 100
          },
          "Component": {
            "Id": "acl-06194fc3a4a03040b",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:network-acl/
acl-06194fc3a4a03040b"
          }
        },
        {
          "SequenceNumber": 3,
          "Component": {
            "Id": "sg-093dde06415d03924",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:security-group/
sg-093dde06415d03924",
            "Name": "VPC BPA Instances Security Group"
          },
          "SecurityGroupRule": {
            "Cidr": "0.0.0.0/0",
            "Direction": "ingress",
            "PortRange": {
              "From": 22,
              "To": 22
            },
            "Protocol": "tcp"
          }
        },
        {
          "SequenceNumber": 4,
          "AttachedTo": {
            "Id": "i-058db34f9a0997895",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:instance/
i-058db34f9a0997895",
            "Name": "VPC BPA Instance A"
          },
          "Component": {
            "Id": "eni-0fa23f2766f03b286",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:network-interface/
eni-0fa23f2766f03b286"
          },
          "InboundHeader": {
            "DestinationAddresses": [
              "10.0.1.85/32"
            ],
            "DestinationPortRanges": [
```

```
              {
                "From": 22,
                "To": 22
              }
            ],
            "Protocol": "6",
            "SourceAddresses": [
              "0.0.0.0/5",
              "100.0.0.0/10",
              "96.0.0.0/6"
            ],
            "SourcePortRanges": [
              {
                "From": 0,
                "To": 65535
              }
            ]
          },
          "Subnet": {
            "Id": "subnet-035d235a762eeed04",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:subnet/
subnet-035d235a762eeed04",
            "Name": "VPC BPA Public Subnet A"
          },
          "Vpc": {
            "Id": "vpc-0762547ec48b6888d",
            "Arn": "arn:aws:ec2:us-east-2:470889052923:vpc/
vpc-0762547ec48b6888d",
            "Name": "VPC BPA"
          }
        }
      ]
    }
  ],
  "AnalysisStatus": "succeeded",
  "NetworkInsightsAccessScopeAnalysisId": "nisa-0aa383a1938f94cd1",
  "NextToken":
 "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ=="
}
```


その結果、アカウント内のすべての VPC のインターネットゲートウェイとの間のトラフィックが表示されます。結果は「検出結果」として整理されます。"FindingId": "AnalysisFinding-1" は、これが分析の最初の結果であることを示します。複数の検出

結果があり、それぞれが VPC BPA をオンにすることで影響を受けるトラフィックフローを示しています。最初の検出結果は、トラフィックがインターネットゲートウェイ ("SequenceNumber": 1) で開始され、NACL ("SequenceNumber": 2)、セキュリティグループ ("SequenceNumber": 3) の順に渡され、インスタンス ("SequenceNumber": 4) で終了したことを示しています。

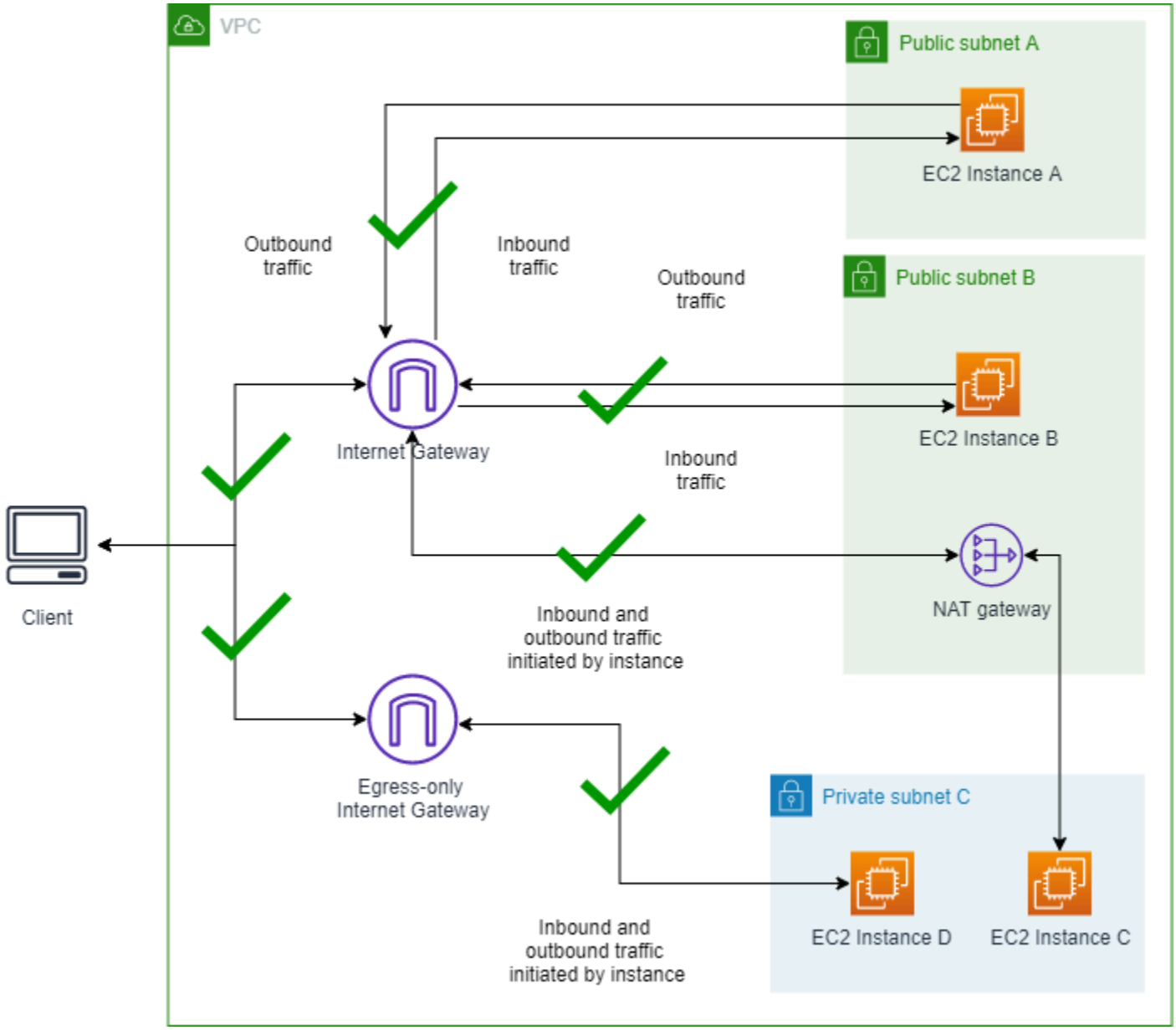
4. 検出結果を分析して、VPC 内のリソースに対する VPC BPA の影響を理解します。

影響分析が完了しました。

シナリオ 1 - VPC BPA が有効になっていないインスタンスに接続する

このセクションでは、パブリックサブネット A と B の EC2 インスタンスにインターネットゲートウェイ経由でインターネットから到達可能であり、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許可します。プライベートサブネットのインスタンス C および D は、NAT ゲートウェイまたは Egress-Only インターネットゲートウェイを介してアウトバウンドトラフィックを開始できますが、インターネットから直接到達することはできません。この設定では、一部のリソースへのインターネットアクセスを提供しながら、他のリソースを保護します。この設定の目的は、ベースラインを設定し、VPC BPA を有効にする前にすべてのインスタンスに到達できるようにし、すべてのインスタンスに接続してパブリック IP アドレスに対して ping を実行することです。

VPC BPA がオンになっていない VPC の図



1.1 インスタンスに接続する

VPC BPA をオフにした状態でインスタンスに接続し、問題なく接続できるようにするには、このセクションを完了します。この例のために CloudFormation を使用して作成されたすべてのインスタンスには、「VVPC BPA Instance A」のような名前が付けられています。

AWS マネジメントコンソール

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. インスタンス A の詳細を開きます。

3. [EC2 Instance Connect] > [EC2 Instance Connect エンドポイントを使用して接続] オプションを使用して、インスタンス A に接続します。
4. 接続 を選択します。インスタンスに正常に接続したら、www.amazon.com に対して ping を実行して、アウトバウンドリクエストをインターネットに送信できることを検証します。
5. インスタンス A への接続に使用したのと同じ方法を使用してインスタンス B、C、D に接続します。各インスタンスから www.amazon.com に対し ping を実行してアウトバウンドリクエストをインターネットに送信できるかどうかを確認します。

AWS CLI

1. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス A に対して Ping を実行します:

```
ping 18.225.8.244
```

出力:

```
Pinging 18.225.8.244 with 32 bytes of data:

Reply from 18.225.8.244: bytes=32 time=51ms TTL=110
Reply from 18.225.8.244: bytes=32 time=61ms TTL=110
```

ping は成功し、トラフィックはブロックされていません。

2. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available. Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #_   ~_  ####_        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
```

```
~~._.   _/
/ /
/m/'
Last login: Fri Sep 27 18:27:57 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING www-amazon-com.customer.fastly.net (18.65.233.187) 56(84) bytes of data.
64 bytes from 18.65.233.187 (18.65.233.187): icmp_seq=15 ttl=58 time=2.06 ms
64 bytes from 18.65.233.187 (18.65.233.187): icmp_seq=16 ttl=58 time=2.26 ms
```

ping は成功し、トラフィックはブロックされていません。

3. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス B に対して Ping を実行します:

```
ping 3.18.106.198
```

出力:

```
Pinging 3.18.106.198 with 32 bytes of data:
Reply from 3.18.106.198: bytes=32 time=83ms TTL=110
Reply from 3.18.106.198: bytes=32 time=54ms TTL=110
```

ping は成功し、トラフィックはブロックされていません。

4. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id  i-08552a0774b5c8f72 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.
Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
/ /
```

```
/m/'
Last login: Fri Sep 27 18:12:27 2024 from 3.16.146.5
[ec2-user@ip-10-0-2-98 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=1 ttl=249 time=1.55 ms
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=2 ttl=249 time=1.67 ms
```

ping は成功し、トラフィックはブロックされていません。

5. インスタンス C に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.
Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
/ /
/m/'
Last login: Thu Sep 19 20:31:26 2024 from 10.0.2.86
[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=1 ttl=248 time=1.75 ms
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=2 ttl=248 time=1.97 ms
64 bytes from server-3-160-24-26.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=3 ttl=248 time=1.08 ms
```

ping は成功し、トラフィックはブロックされていません。

6. インスタンス D に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

出力:

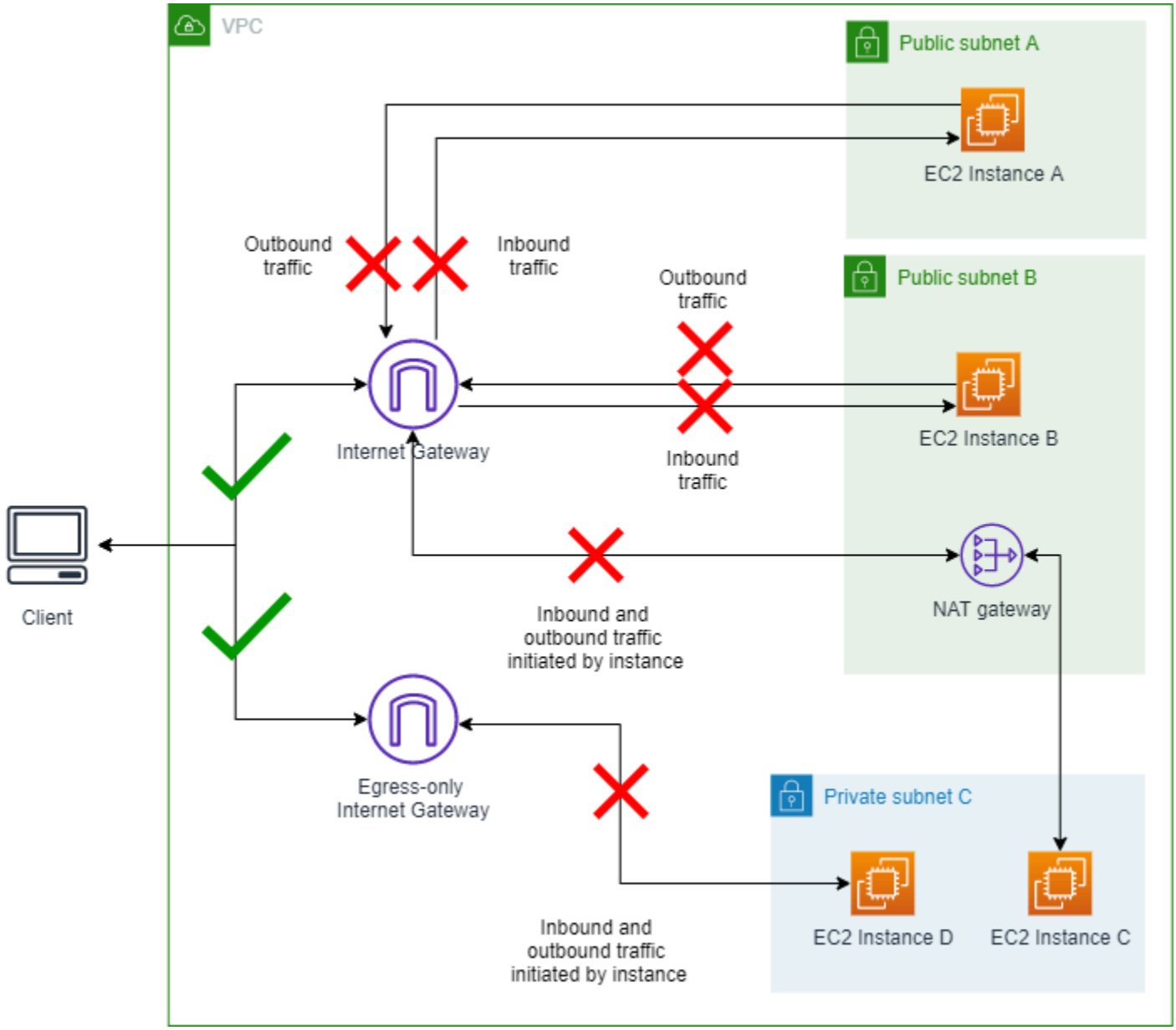
```
The authenticity of host '10.0.3.59 can't be established.
ECDSA key fingerprint is SHA256:c4naBCqbC61/cExDyccEproNU+1HHSpMSzl2J6cOtIZA8g.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.3.59' (ECDSA) to the list of known hosts.
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
_/ _/
_/m/'
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING www.amazon.com(2600:9000:25f3:ee00:7:49a5:5fd4:b121
 (2600:9000:25f3:ee00:7:49a5:5fd4:b121)) 56 data bytes
64 bytes from 2600:9000:25f3:ee00:7:49a5:5fd4:b121
 (2600:9000:25f3:ee00:7:49a5:5fd4:b121): icmp_seq=1 ttl=58 time=1.19 ms
64 bytes from 2600:9000:25f3:ee00:7:49a5:5fd4:b121
 (2600:9000:25f3:ee00:7:49a5:5fd4:b121): icmp_seq=2 ttl=58 time=1.38 ms
```

ping は成功し、トラフィックはブロックされていません。

シナリオ 2 - 双方向モードで VPC BPA を有効にする

このセクションでは、VPC BPA をオンにし、アカウントのインターネットゲートウェイとの間のトラフィックをブロックします。

双方向モードがオンになっている VPC BPA を示す図



2.1 VPC BPA 双方向モードを有効にする

VPC BPA を有効にするには、このセクションを完了します。VPC BPA 双方向モードは、このリージョンのインターネットゲートウェイとエグレスのみのインターネットゲートウェイとの間のすべてのトラフィックをブロックします (除外された VPC とサブネットを除く)。

AWS マネジメントコンソール

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 左のナビゲーションペインで、[設定] を選択します。
3. [パブリックアクセスの設定を編集] を選択します。

4. [ブロックパブリックアクセスをオンにする] と [双方向] を選択し、[変更を保存] を選択します。
5. [ステータス] が [オン] に変わるまで待ちます。VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。

VPC BPA がオンになりました。

AWS CLI

1. modify-vpc-block-public-access-options コマンドを使用して、VPC BPA をオンにします。

```
aws ec2 --region us-east-2 modify-vpc-block-public-access-options --internet-gateway-block-mode block-bidirectional
```

VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。

2. VPC BPA のステータスを表示します。

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-options
```

2.2 インスタンスに接続する

インスタンスに接続するには、このセクションを完了します。

AWS マネジメントコンソール

1. シナリオ 1 で実行したように、インスタンス A とインスタンス B のパブリック IPv4 アドレスに対して Ping を実行します。トラフィックがブロックされます。
2. シナリオ 1 で行ったように、[EC2 Instance Connect] > [EC2 Instance Connect Endpoint] オプションを使用して、インスタンス A に接続します。エンドポイントオプションを使用していることを確認してください。
3. 接続 を選択します。インスタンスに正常に接続したら、www.amazon.com に ping を実行します。すべてのアウトバウンドトラフィックがブロックされます。
4. インスタンス A への接続に使用したのと同じ方法を使用してインスタンス B、C、D に接続し、インターネットにアウトバウンドリクエストを送信できるかテストします。すべてのアウトバウンドトラフィックがブロックされます。

AWS CLI

1. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス A に対して Ping を実行します:

```
ping 18.225.8.244
```

出力:

```
Pinging 18.225.8.244 with 32 bytes of data:

Request timed out.
```

ping が失敗し、トラフィックがブロックされます。

2. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

出力:

```
The authenticity of host '10.0.1.85' can't be established.
ECDSA key fingerprint is SHA256:3zo/gSss+HAZ+7eTyWlOB/Ke04IM+hadjsoLJeRTWBk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.85' (ECDSA) to the list of known hosts.
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #_   ~_  ####_        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~._.   _/
/ /
/m/'
Last login: Fri Sep 27 14:16:53 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping が失敗し、トラフィックがブロックされます。

3. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス B に対して Ping を実行します:

```
ping 3.18.106.198
```

出力:

```
Pinging 3.18.106.198 with 32 bytes of data:
Request timed out.
```

ping が失敗し、トラフィックがブロックされます。

4. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id  i-08552a0774b5c8f72 --region us-
east-2 --connection-type eice
```

出力:

```
The authenticity of host '10.0.2.98' can't be established.
ECDSA key fingerprint is SHA256:0IjXKKyVlDthcCfI0IPIJMUiItAOLYKRNLGTYURnFXo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.98' (ECDSA) to the list of known hosts.
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
/ /
/m/'
Last login: Fri Sep 27 14:18:16 2024 from 3.16.146.5
[ec2-user@ip-10-0-2-98 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping が失敗し、トラフィックがブロックされます。

5. インスタンス C に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
/ /
/m/'
Last login: Tue Sep 24 15:17:56 2024 from 10.0.2.86
[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping が失敗し、トラフィックがブロックされます。

6. インスタンス D に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
```

```
~~..   _/
_/ _/
_/m/'
Last login: Fri Sep 27 16:42:01 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING www.amazon.com(2600:9000:25f3:8200:7:49a5:5fd4:b121
 (2600:9000:25f3:8200:7:49a5:5fd4:b121)) 56 data bytes
```

ping が失敗し、トラフィックがブロックされます。

2.3 オプション: Reachability Analyzer を使用して接続がブロックされていることを確認する

VPC Reachability Analyzer を使用して、VPC BPA の設定を含むネットワーク設定を踏まえて、特定のネットワークパスに到達できるかどうかを把握できます。この例では、以前に試行したのと同じネットワークパスを分析し、VPC BPA が接続に失敗する理由であることを確認します。

AWS マネジメントコンソール

1. https://console.aws.amazon.com/networkinsights/home#ReachabilityAnalyzer の Network Insights コンソールに移動します。
2. [パスの作成と分析] を選択します
3. [ソースタイプ] で、[インターネットゲートウェイ] を選択し、[ソース] ドロップダウンから [VPC BPA インターネットゲートウェイ] のタグが付けられたインターネットゲートウェイを選択します。
4. [送信先タイプ] で、[インスタンス] を選択し、[送信先] ドロップダウンから [VPC BPA インスタンス A] のタグが付けられたインスタンスを選択します。
5. [パスの作成と分析] を選択します
6. 分析が完了するまで待ちます。数分かかる場合があります。
7. 完了すると、[可達のステータス] が [到達不可能] となり、[パスの詳細] に VPC_BLOCK_PUBLIC_ACCESS_ENABLED が原因であることが示されます。

AWS CLI

1. [VPC BPA インターネットゲートウェイ] のタグが付けられたインターネットゲートウェイの ID と、[VPC BPA インスタンス A] のタグが付けられたインスタンスの ID を使用してネットワークパスを作成します。

```
aws ec2 --region us-east-2 create-network-insights-path --source igw-id --
destination instance-id --protocol TCP
```

2. ネットワークパスで分析を開始します:

```
aws ec2 --region us-east-2 start-network-insights-analysis --network-insights-
path-id nip-id
```

3. 分析の結果を取得します:

```
aws ec2 --region us-east-2 describe-network-insights-analyses --network-
insights-analysis-ids nia-id
```

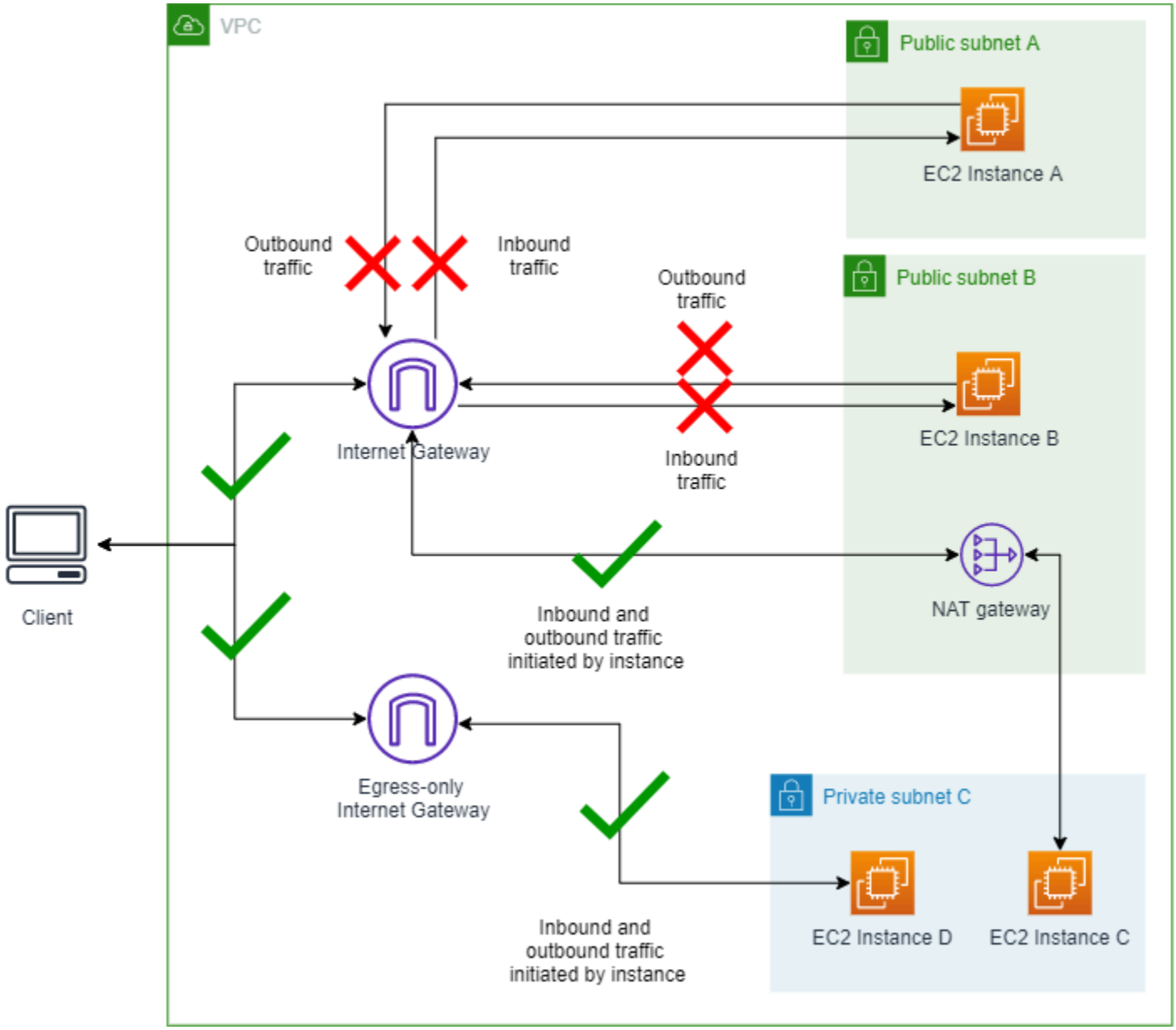
4. 到達可能性の欠如について、VPC_BLOCK_PUBLIC_ACCESS_ENABLED が ExplanationCode であることを確認します。

フローログを使用して VPC BPA の影響をモニタリングすることも可能です。

シナリオ 3 - VPC BPA を Ingress-Only モードに変更する

このセクションでは、VPC BPA トラフィックの方向を変更し、NAT ゲートウェイまたはエグレスのみのインターネットゲートウェイを使用するトラフィックのみを許可します。BPA はインターネットゲートウェイ経由のインバウンドトラフィックをブロックするため、パブリックサブネットの EC2 インスタンス A と B はインターネットから到達できません。プライベートサブネットのインスタンス C と D は、NAT ゲートウェイと Egress-Only インターネットゲートウェイを介してアウトバウンドトラフィックを開始できるため、引き続きインターネットに到達できます。

VPC BPA のイングレスのみのモードがオンになっている図



3.1 モードを「イングレスのみ」に変更する

モードを変更するには、このセクションを完了します。

AWS マネジメントコンソール

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 左のナビゲーションペインで、[設定] を選択します。
3. [パブリックアクセスをブロックする] タブで、[パブリックアクセス設定を編集する] を選択します。

4. VPC コンソールでパブリックアクセスの設定を変更し、方向を [イングレスのみ] に変更します。
5. 変更を保存し、ステータスが更新されるまで待ちます。VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。

AWS CLI

1. VPC BPA モードを変更します:

```
aws ec2 --region us-east-2 modify-vpc-block-public-access-options --internet-gateway-block-mode block-ingress
```

   VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。
2. VPC BPA のステータスを表示します。

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-options
```

3.2 インスタンスに接続する

インスタンスに接続するには、このセクションを完了します。

AWS マネジメントコンソール

1. シナリオ 1 で実行したように、インスタンス A とインスタンス B のパブリック IPv4 アドレスに対して Ping を実行します。トラフィックがブロックされます。
2. シナリオ 1 で実行したように、EC2 Instance Connect を使用してインスタンス A とインスタンス B に接続し、そこから www.amazon.com に対して ping を実行します。インスタンス A または B からインターネット上のパブリックサイトに対して ping を実行することはできず、トラフィックがブロックされます。
3. シナリオ 1 で実行したように、EC2 Instance Connect を使用してインスタンス C とインスタンス D に接続し、そこから www.amazon.com に対して ping を実行します。インスタンス C または D からインターネット上のパブリックサイトに対して ping を実行でき、トラフィックが許可されます。

AWS CLI

1. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス A に対して Ping を実行します:

```
ping 18.225.8.244
```

出力:

```
Pinging 18.225.8.244 with 32 bytes of data:

Request timed out.
```

ping が失敗し、トラフィックがブロックされます。

2. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

出力:

```
The authenticity of host '10.0.1.85' can't be established.
ECDSA key fingerprint is SHA256:3zo/gSss+HAZ+7eTyWlOB/Ke04IM+hadjsoLJeRTWBk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.85' (ECDSA) to the list of known hosts.
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #_   ~_  ####_        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~._.   _/
/ /
/m/'
Last login: Fri Sep 27 14:16:53 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping が失敗し、トラフィックがブロックされます。

3. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス B に対して Ping を実行します:

```
ping 3.18.106.198
```

出力:

```
Pinging 3.18.106.198 with 32 bytes of data:
Request timed out.
```

ping が失敗し、トラフィックがブロックされます。

4. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id i-08552a0774b5c8f72 --region us-
east-2 --connection-type eice
```

出力:

```
The authenticity of host '10.0.2.98 ' can't be established.
ECDSA key fingerprint is SHA256:0IjXKKyVlDthcCfI0IPIJMUiItAOLYKRNLGTYURnFXo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.98' (ECDSA) to the list of known hosts.
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
_/ /
/m/'
Last login: Fri Sep 27 14:18:16 2024 from 3.16.146.5
[ec2-user@ip-10-0-2-98 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping が失敗し、トラフィックがブロックされます。

5. インスタンス C に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'

Last login: Tue Sep 24 15:28:09 2024 from 10.0.2.86

[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com

PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.

64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=1 ttl=248 time=1.84 ms
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=2 ttl=248 time=1.40 ms
```

ping は成功し、トラフィックはブロックされていません。

6. インスタンス D に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 16:48:38 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING www.amazon.com(2600:9000:25f3:5800:7:49a5:5fd4:b121
 (2600:9000:25f3:5800:7:49a5:5fd4:b121)) 56 data bytes
64 bytes from 2600:9000:25f3:5800:7:49a5:5fd4:b121
 (2600:9000:25f3:5800:7:49a5:5fd4:b121): icmp_seq=14 ttl=58 time=1.47 ms
64 bytes from 2600:9000:25f3:5800:7:49a5:5fd4:b121
 (2600:9000:25f3:5800:7:49a5:5fd4:b121): icmp_seq=16 ttl=58 time=1.59 ms
```

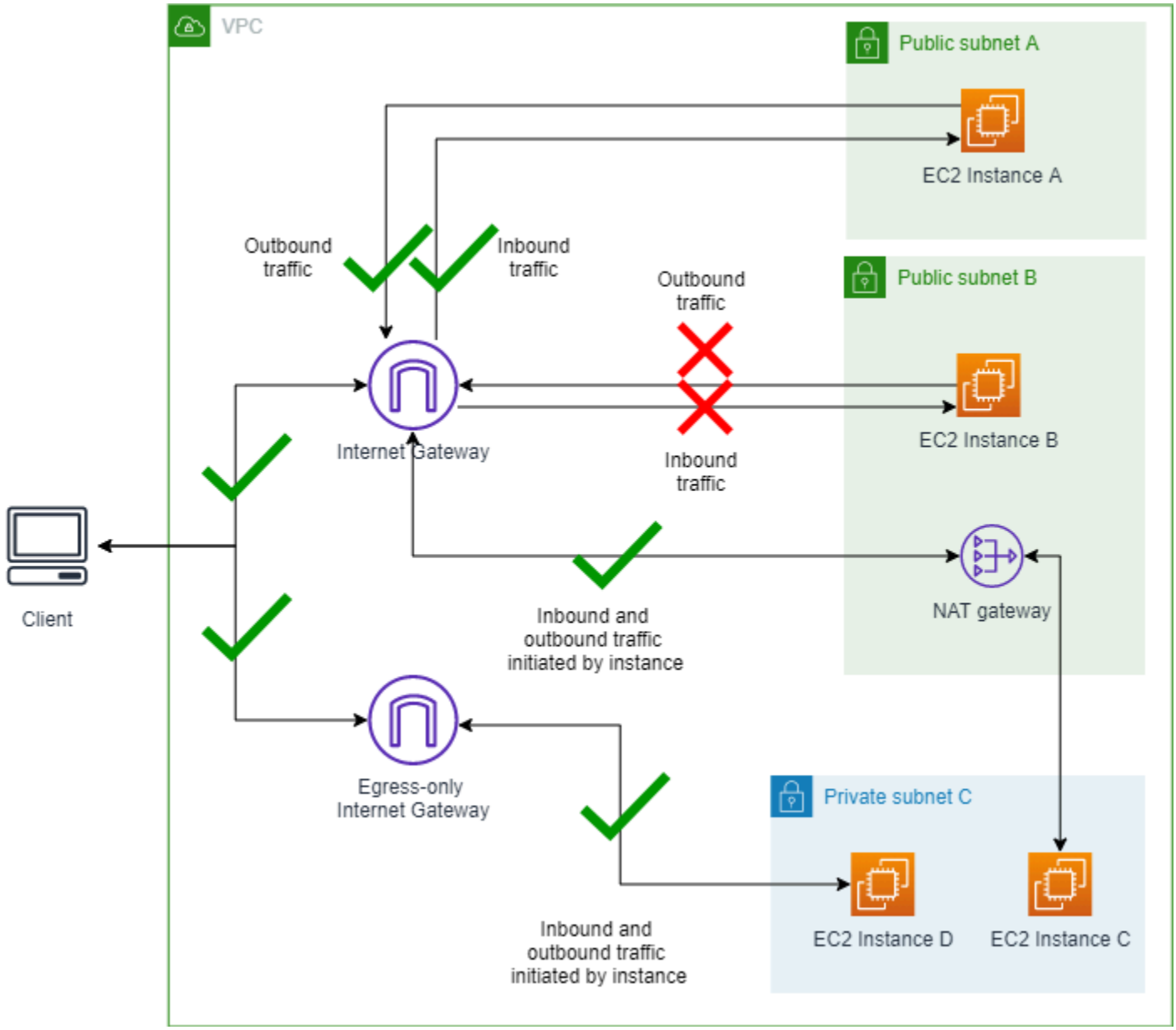
ping は成功し、トラフィックはブロックされていません。

シナリオ 4 - 除外を作成する

このセクションでは、除外を作成します。その後、VPC BPA は除外なしでサブネット上のトラフィックのみをブロックします。VPC BPA の除外は、単一の VPC またはサブネットに適用できるモードです。有効にした場合、対象の VPC またはサブネットがアカウントの VPC BPA モードから除外され、双方向または Egress-Only のアクセスが許可されます。アカウントで VPC BPA が有効になっていない場合でも VPC とサブネットのために VPC BPA の除外を作成して、VPC BPA がオンになっているときに除外に対するトラフィックの中断が発生しないようにできます。

この例では、サブネット A の除外を作成して、除外に対するトラフィックが VPC BPA によってどのように影響を受けるかを示します。

VPC BPA のイングレスのみのモードがオンになっており、[双方向] モードがオンになっているサブネット A の除外の図:



4.1 サブネット A の除外を作成する

除外を作成するには、このセクションを完了します。VPC BPA の除外は、単一の VPC またはサブネットに適用できるモードです。有効にした場合、対象の VPC またはサブネットがアカウントの VPC BPA モードから除外され、双方向または Egress-Only のアクセスが許可されます。アカウントで VPC BPA が有効になっていない場合でも VPC とサブネットのために VPC BPA の除外を作成して、VPC BPA がオンになっているときに除外に対するトラフィックの中断が発生しないようにできます。

AWS マネジメントコンソール

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 左のナビゲーションペインで、[設定] を選択します。
3. [ブロックパブリックアクセス] タブの [除外] で、[除外を作成] を選択します。
4. [VPC BPA パブリックサブネット A] を選択し、許可の方向として [双方向] が選択されているようにして、[除外を作成] を選択します。
5. [除外ステータス] が [アクティブ] に変わるまで待ちます。変更を確認するには、除外テーブルを更新する必要がある場合があります。

除外が作成されました。

AWS CLI

1. 除外の許可の方向を変更します:

```
aws ec2 --region us-east-2 create-vpc-block-public-access-exclusion --subnet-
id subnet-id --internet-gateway-exclusion-mode allow-bidirectional
```

2. 除外ステータスが更新されるまでに時間がかかる場合があります。除外のステータスを表示するには:

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-exclusions --
exclusion-ids exclusion-id
```

4.2 インスタンスに接続する

インスタンスに接続するには、このセクションを完了します。

AWS マネジメントコンソール

1. インスタンス A のパブリック IPv4 アドレスに対して Ping を実行します。トラフィックが許可されます。
2. インスタンス B のパブリック IPv4 アドレスに対して Ping を実行します。トラフィックがブロックされます。
3. シナリオ 1 で実行したように、EC2 Instance Connect を使用してインスタンス A に接続し、www.amazon.com に対して ping を実行します。インスタンス A からインターネット上のパブリックサイトに対して ping を実行できます。トラフィックは許可されます。

4. シナリオ 1 で実行したように、EC2 Instance Connect を使用してインスタンス B に接続し、そこから www.amazon.com に対して ping を実行します。インスタンス B からインターネット上のパブリックサイトに対して ping を実行することはできません。トラフィックがブロックされます。
5. シナリオ 1 で実行したように、EC2 Instance Connect を使用してインスタンス C とインスタンス D に接続し、そこから www.amazon.com に対して ping を実行します。インスタンス C または D からインターネット上のパブリックサイトに対して ping を実行できます。トラフィックは許可されます。

AWS CLI

1. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス A に対して Ping を実行します:

```
ping 18.225.8.244
```

出力:

```
Pinging 18.225.8.244 with 32 bytes of data:

Reply from 18.225.8.244: bytes=32 time=51ms TTL=110
Reply from 18.225.8.244: bytes=32 time=61ms TTL=110
```

ping は成功し、トラフィックはブロックされていません。

2. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #_   ~_  ####_        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
```

```
~~~         /
~~._.   _/
/ /
/m/'
Last login: Fri Sep 27 17:58:12 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=1 ttl=249 time=1.03 ms
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=2 ttl=249 time=1.72 ms
```

ping は成功し、トラフィックはブロックされていません。

3. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス B に対して Ping を実行します:

```
ping 3.18.106.198
```

出力:

```
Pinging 3.18.106.198 with 32 bytes of data:
Request timed out.
```

ping が失敗し、トラフィックがブロックされます。

4. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id i-08552a0774b5c8f72 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
```

```
_/ /
/m/'
Last login: Fri Sep 27 18:12:03 2024 from 3.16.146.5
[ec2-user@ip-10-0-2-98 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping が失敗し、トラフィックがブロックされます。

5. インスタンス C に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

Output

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
,     #   ~_  ####        Amazon Linux 2023
~~  _#####\  ~~     ###|
~~       #/ ___   https://aws.amazon.com/linux/amazon-linux-2023
~~       V~' '->
~~~         /
~~..   _/
_/ /
/m/'

Last login: Tue Sep 24 15:28:09 2024 from 10.0.2.86

[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com

PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.

64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=1 ttl=248 time=1.84 ms
64 bytes from server-3-160-24-126.cmh68.r.cloudfront.net (18.65.233.187):
 icmp_seq=2 ttl=248 time=1.40 ms
```

ping は成功し、トラフィックはブロックされていません。

6. インスタンス D に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

Output

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:00:52 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING
 www.amazon.com(g2600-141f-4000-059a-0000-0000-0000-3bd4.deploy.static.akamaitechnologie
 (2600:141f:4000:59a::3bd4)) 56 data bytes
64 bytes from
 g2600-141f-4000-059a-0000-0000-0000-3bd4.deploy.static.akamaitechnologies.com
 (2600:141f:4000:59a::3bd4): icmp_seq=1 ttl=48 time=15.9 ms
64 bytes from
 g2600-141f-4000-059a-0000-0000-0000-3bd4.deploy.static.akamaitechnologies.com
 (2600:141f:4000:59a::3bd4): icmp_seq=2 ttl=48 time=15.8 ms
```

ping は成功し、トラフィックはブロックされていません。

4.3 オプション: Reachability Analyzer との接続を検証する

シナリオ 2 で Reachability Analyzer において作成したのと同じネットワークパスを使用して、新しい分析を実行し、パブリックサブネット A についての除外が作成されたことでパスが到達可能になったことを確認できるようになりました。

Reachability Analyzer を利用できるリージョンについては、「Reachability Analyzer ガイド」の「Considerations」を参照してください。

AWS マネジメントコンソール

1. Network Insights コンソールで前に作成したネットワークパスから、[分析を再実行] を選択します。
2. 分析が完了するまで待ちます。これには数分間かかる場合があります。
3. パスが [到達可能] になっていることを確認します。

AWS CLI

1. 前に作成したネットワークパス ID を使用して、新しい分析を開始します:

```
aws ec2 --region us-east-2 start-network-insights-analysis --network-insights-path-id nip-id
```

2. 分析の結果を取得します:

```
aws ec2 --region us-east-2 describe-network-insights-analyses --network-insights-analysis-ids nia-id
```

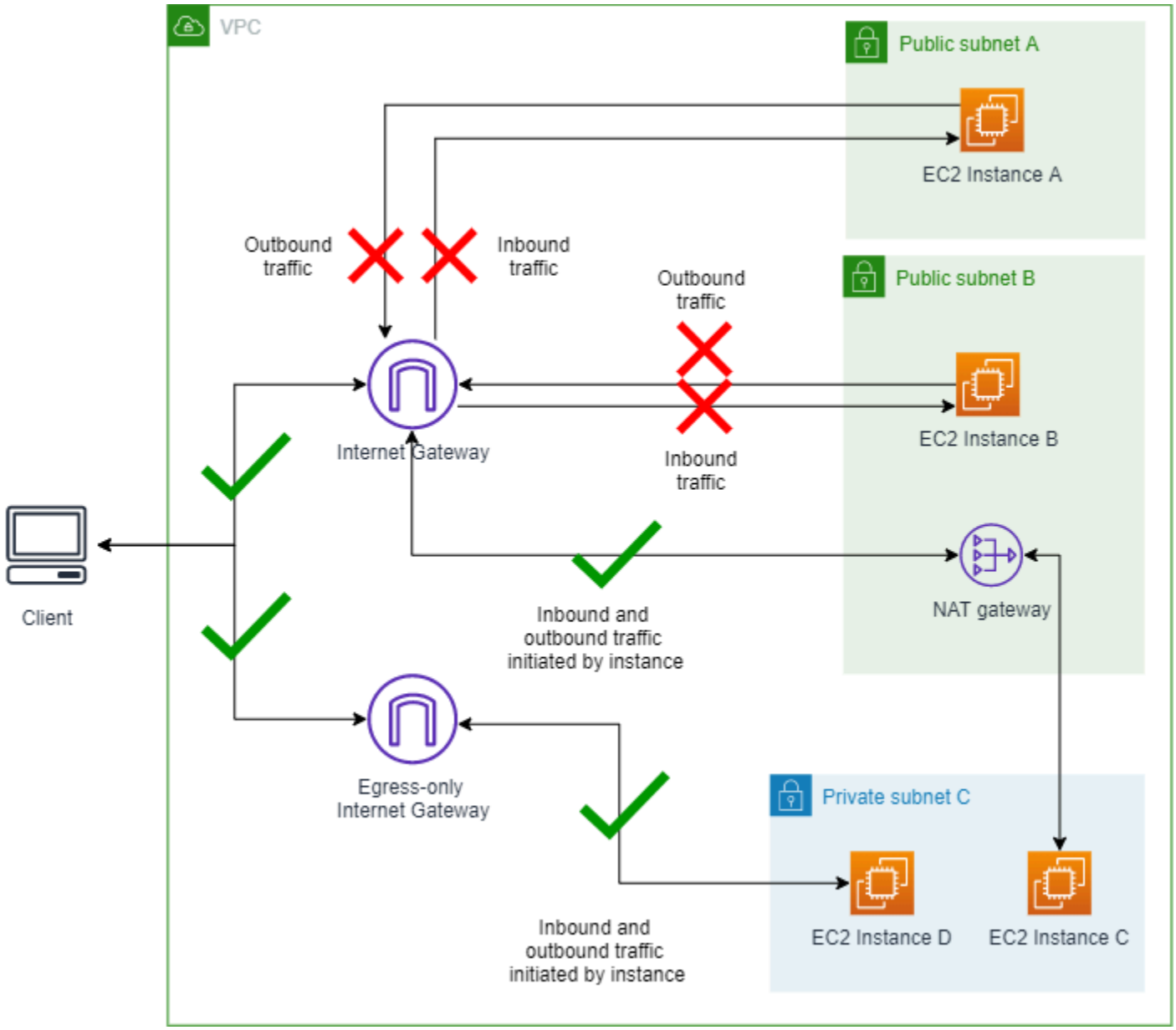
3. VPC_BLOCK_PUBLIC_ACCESS_ENABLED 説明コードが存在しないことを確認します。

シナリオ 5 - 除外モードを変更する

このセクションでは、除外における許可トラフィックの方向を変更して、VPC BPA がどのような影響を受けるのかを確認します。

> **Note**
>
> このシナリオでは、除外モードを Egress-Only に変更します。この操作を行うと、サブネット A の Egress-Only の除外ではアウトバウンドトラフィックが許可されないことに注意してください (Egress-Only という名前からすると、アウトバウンドトラフィックが許可されると予想されますがそうではありません)。ただし、アカウントレベルの BPA は Ingress-Only であるため、Egress-Only の除外は無視され、サブネット A のインターネットゲートウェイへのルーティングは VPC BPA によって制限され、アウトバウンドトラフィックがブロックされます。サブネット A でアウトバウンドトラフィックを有効にするには、VPC BPA を双方向モードに切り替える必要があります。

VPC BPA のイングレスのみのモードがオンになっており、エグレスのみのモードがオンになっているサブネット A の除外の図:



5.1 除外の許可の方向をエグレスのみに変更する

除外の許可の方向を変更するには、このセクションを完了します。

AWS マネジメントコンソール

1. シナリオ 4 で作成した除外を編集し、許可の方向を [エグレスのみ] に変更します。
2. [Save changes] (変更の保存) をクリックします。

3. [除外] ステータスが [アクティブ] に変わるまで待ちます。VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。変更を確認するには、除外テーブルを更新する必要がある場合があります。

AWS CLI

1. 除外の許可の方向を変更します:

```
aws ec2 --region us-east-2 modify-vpc-block-public-access-exclusion --exclusion-
id exclusion-id --internet-gateway-exclusion-mode allow-egress
```

   VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。

2. 除外ステータスが更新されるまでに時間がかかる場合があります。除外のステータスを表示するには:

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-exclusion
```

5.2 インスタンスに接続する

インスタンスに接続するには、このセクションを完了します。

AWS マネジメントコンソール

1. インスタンス A およびインスタンス B のパブリック IPv4 アドレスに対して Ping を実行します。トラフィックがブロックされます。
2. シナリオ 1 で実行したように、EC2 Instance Connect を使用してインスタンス A およびインスタンス B に接続し、www.amazon.com に対して ping を実行します。インスタンス A またはインスタンス B からインターネット上のパブリックサイトに対して ping を実行することはできません。トラフィックがブロックされます。
3. シナリオ 1 で実行したように、EC2 Instance Connect を使用してインスタンス C とインスタンス D に接続し、そこから www.amazon.com に対して ping を実行します。インスタンス C または D からインターネット上のパブリックサイトに対して ping を実行できます。トラフィックは許可されます。

AWS CLI

1. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス A に対して Ping を実行します:

```
ping 18.225.8.244
```

出力:

```
Pinging 18.225.8.244 with 32 bytes of data:
Request timed out.
```

ping が失敗し、トラフィックがブロックされます。

2. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:09:55 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping が失敗し、トラフィックがブロックされます。

3. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス B に対して Ping を実行します:

```
ping 3.18.106.198
```

出力:

```
Pinging 3.18.106.198 with 32 bytes of data:
Request timed out.
```

ping が失敗し、トラフィックがブロックされます。

4. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:09:55 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping が失敗し、トラフィックがブロックされます。

5. インスタンス C に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:00:31 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com
PING www.amazon.com(2600:9000:25f3:a600:7:49a5:5fd4:b121
 (2600:9000:25f3:a600:7:49a5:5fd4:b121)) 56 data bytes
64 bytes from 2600:9000:25f3:a600:7:49a5:5fd4:b121
 (2600:9000:25f3:a600:7:49a5:5fd4:b121): icmp_seq=1 ttl=58 time=1.51 ms
64 bytes from 2600:9000:25f3:a600:7:49a5:5fd4:b121
 (2600:9000:25f3:a600:7:49a5:5fd4:b121): icmp_seq=2 ttl=58 time=1.49 ms
```

ping は成功し、トラフィックはブロックされていません。

6. インスタンス D に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
```

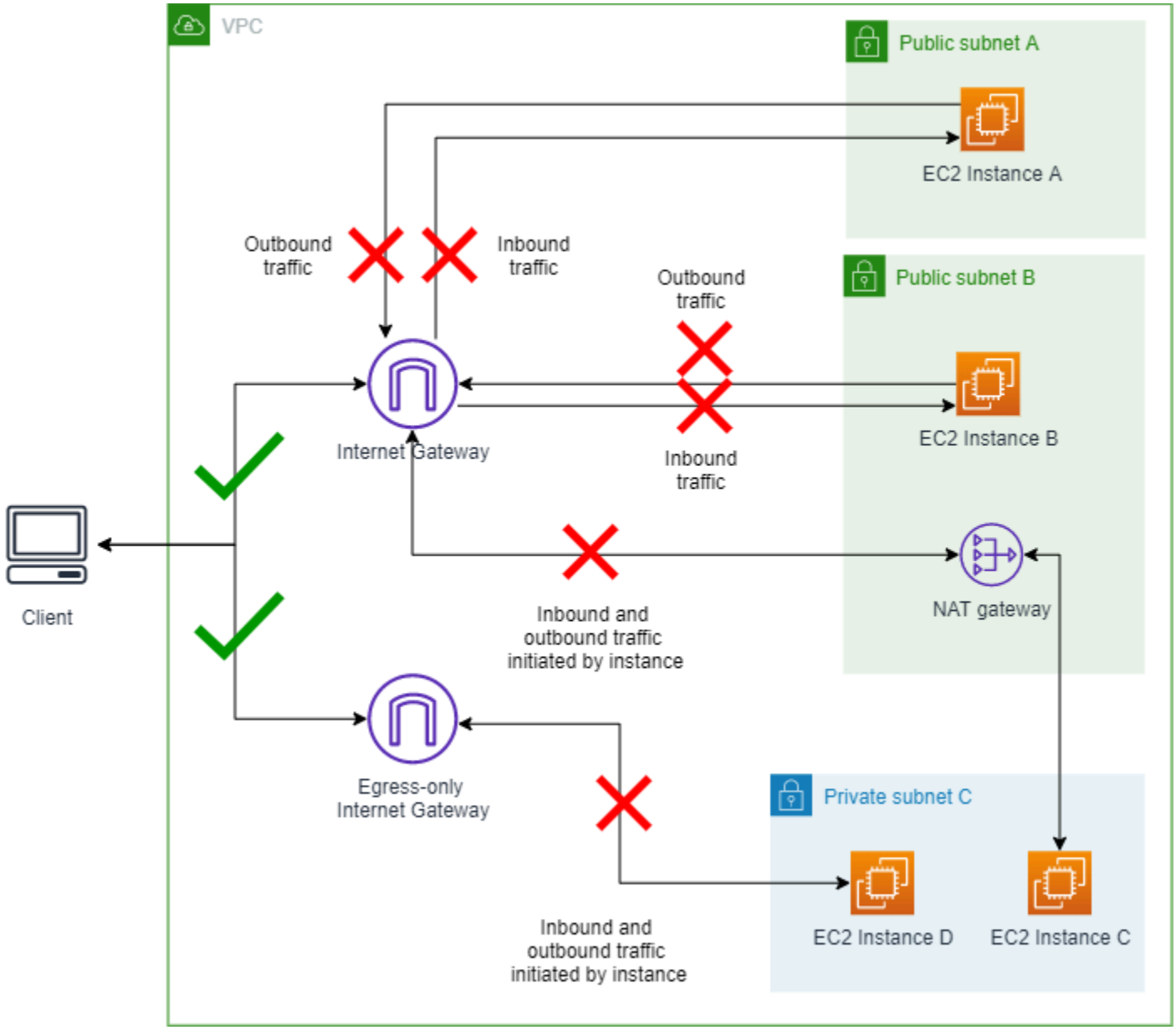
```
       _/m/'
Last login: Fri Sep 27 18:13:55 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING www.amazon.com(2606:2cc0::374 (2606:2cc0::374)) 56 data bytes
64 bytes from 2606:2cc0::374 (2606:2cc0::374): icmp_seq=1 ttl=58 time=1.21 ms
64 bytes from 2606:2cc0::374 (2606:2cc0::374): icmp_seq=2 ttl=58 time=1.51 ms
```

ping は成功し、トラフィックはブロックされていません。

シナリオ 6 - VPC BPA モードを変更する

このセクションでは、トラフィックにどのような影響が及ぶのかを確認するために、VPC BPA のブロックの方向を変更します。このシナリオでは、双方向モードで有効になっている VPC BPA は、シナリオ 1 と同様にすべてのトラフィックをブロックします。除外が NAT ゲートウェイまたはエグレスのみのインターネットゲートウェイにアクセスできる場合を除き、トラフィックはブロックされます。

VPC BPA の [双方向] モードがオンになっており、[エグレスのみ] のモードがオンになっているサブネット A の除外の図:



6.1 VPC BPA を双方向モードに変更する

VPC BPA のモードを変更するには、このセクションを完了します。

AWS マネジメントコンソール

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. 左のナビゲーションペインで、[設定] を選択します。
3. [パブリックアクセスの設定を編集] を選択します。
4. ブロックの方向を [双方向] に変更し、[変更を保存] を選択します。

5. [ステータス] が [オン] に変わるまで待ちます。VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。

AWS CLI

1. VPC BPA ブロックの方向を変更します:

```
aws ec2 --region us-east-2 modify-vpc-block-public-access-options --internet-gateway-block-mode block-bidirectional
```

VPC BPA の設定が有効になり、ステータスが更新されるまでに数分かかる場合があります。

2. VPC BPA のステータスを表示します。

```
aws ec2 --region us-east-2 describe-vpc-block-public-access-options
```

6.2 インスタンスに接続する

インスタンスに接続するには、このセクションを完了します。

AWS マネジメントコンソール

1. インスタンス A およびインスタンス B のパブリック IPv4 アドレスに対して Ping を実行します。トラフィックがブロックされます。
2. シナリオ 1 で実行したように、EC2 Instance Connect を使用してインスタンス A およびインスタンス B に接続し、www.amazon.com に対して ping を実行します。インスタンス A またはインスタンス B からインターネット上のパブリックサイトに対して ping を実行することはできません。トラフィックがブロックされます。
3. シナリオ 1 で実行したように、EC2 Instance Connect を使用してインスタンス C とインスタンス D に接続し、そこから www.amazon.com に対して ping を実行します。インスタンス C またはインスタンス D からインターネット上のパブリックサイトに対して ping を実行することはできません。トラフィックがブロックされます。

AWS CLI

1. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス A に対して Ping を実行します:

```
ping 18.225.8.244
```

出力:

```
Pinging 18.225.8.244 with 32 bytes of data:
Request timed out.
```

ping が失敗し、トラフィックがブロックされます。

2. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:17:44 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping が失敗し、トラフィックがブロックされます。

3. インバウンドトラフィックをチェックするために、パブリック IPv4 アドレスを使用してインスタンス A に対して Ping を実行します:

```
ping 3.18.106.198
```

出力:

```
Pinging 3.18.106.198 with 32 bytes of data:
Request timed out.
```

ping が失敗し、トラフィックがブロックされます。

4. プライベート IPv4 アドレスを使用して、アウトバウンドトラフィックに接続してチェックします。

```
aws ec2-instance-connect ssh --instance-id i-058db34f9a0997895 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:09:55 2024 from 3.16.146.5
[ec2-user@ip-10-0-1-85 ~]$ ping www.amazon.com
PING d3ag4hukkh62yn.cloudfront.net (18.65.233.187) 56(84) bytes of data.
```

ping が失敗し、トラフィックがブロックされます。

5. インスタンス C に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-04eca55f2a482b2c4 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
```

```
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:19:45 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-180 ~]$ ping www.amazon.com
PING www.amazon.com(2600:9000:25f3:6200:7:49a5:5fd4:b121
 (2600:9000:25f3:6200:7:49a5:5fd4:b121)) 56 data bytes
```

ping が失敗し、トラフィックがブロックされます。

6. インスタンス D に接続します。ping を実行するためのパブリック IP アドレスがないので、EC2 Instance Connect を使用して接続してから、インスタンスからパブリック IP に対して ping を実行してアウトバウンドトラフィックをチェックします:

```
aws ec2-instance-connect ssh --instance-id i-05f9e6a9cfac1dba0 --region us-
east-2 --connection-type eice
```

出力:

```
A newer release of "Amazon Linux" is available.  Version 2023.5.20240916:
Run "/usr/bin/dnf check-release-update" for full release and version update info
   ,     #_   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 27 18:20:58 2024 from 3.16.146.5
[ec2-user@ip-10-0-3-59 ~]$ ping www.amazon.com
PING www.amazon.com(2600:9000:25f3:b400:7:49a5:5fd4:b121
 (2600:9000:25f3:b400:7:49a5:5fd4:b121)) 56 data bytes
```

ping が失敗し、トラフィックがブロックされます。

クリーンアップ

このセクションでは、この高度な例のために作成したすべてのリソースを削除します。アカウントで作成されたリソースについて余分な追加料金が発生しないように、リソースをクリーンアップすることが重要です。

CloudFormation リソースを削除する

CloudFormation テンプレートを使用して作成したリソースを削除するには、このセクションを完了します。

AWS マネジメントコンソール

1. で CloudFormation コンソールを開きます。。https://console.aws.amazon.com/cloudformation/
2. VPC BPA スタックを選択します。
3. [削除] を選択します。
4. スタックの削除を開始したら、[イベント] タブで進行状況を表示し、スタックが削除されていることを確認します。スタックを完全に削除するには、そのスタックを強制削除する必要がある場合があります。

AWS CLI

1. CloudFormation スタックを削除します。スタックを完全に削除するには、そのスタックを強制削除する必要がある場合があります。

```
aws cloudformation delete-stack --stack-name VPC-BPA-stack --region us-east-2
```

2. 進行状況を表示し、スタックが削除されていることを確認します。

```
aws cloudformation describe-stack-events --stack-name VPC-BPA-stack --region us-east-2
```

CloudTrail を使用して除外の削除を追跡する

AWS CloudTrail を使用して除外の削除を追跡するには、このセクションを完了します。CloudTrail エントリは、除外を削除すると表示されます。

AWS マネジメントコンソール

https://console.aws.amazon.com/cloudtrailv2/ で AWS CloudTrail コンソールにおいて [リソースタイプ] > AWS::EC2::VPCBlockPublicAccessExclusion を検索すると、CloudTrail Event の履歴で、削除された除外を表示できます。

AWS CLI

lookup-events コマンドを使用して、除外の削除に関連するイベントを表示できます:

```
aws cloudtrail lookup-events --lookup-attributes
 AttributeKey=ResourceType,AttributeValue=AWS::EC2::VPCBlockPublicAccessExclusion
```

高度な例が完了しました。

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を提供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必ずしも適切または十分でない可能性があるため、処方箋ではなく、あくまで有用な検討事項とお考えください。

- VPC にサブネットを追加してアプリケーションをホストするときは、複数のアベイラビリティーゾーンにサブネットを作成します。アベイラビリティーゾーンは、AWS リージョンに冗長電源、ネットワーク、および接続を備えた 1 つ以上の個別のデータセンターです。複数のアベイラビリティーゾーンを使用すると、本番環境アプリケーションの可用性、耐障害性、およびスケーラビリティが向上します。
- セキュリティグループを使用して、サブネット内の EC2 インスタンスへのトラフィックを制御します。詳細については、「セキュリティグループ」を参照してください。
- ネットワーク ACL を使用して、サブネットレベルで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します。詳細については、「ネットワークアクセスコントロールリストを使用して、サブネットのトラフィックを制御する」を参照してください。
- AWS Identity and Access Management (IAM) ID フェデレーション、ユーザー、およびロールを使用して、VPC 内の AWS リソースへのアクセスを管理します。詳細については、「Amazon VPC の Identity and Access Management」を参照してください。
- VPC フローログを使用して、VPC、サブネット、またはネットワークインターフェイス間で送受信される IP トラフィックを監視します。詳細については、「VPC Flow Logs」を参照してください。

- Network Access Analyzer を使用して、VPC 内のリソースへの意図しないネットワークアクセスを特定します。詳細については、「Network Access Analyzer ガイド」を参照してください。
- AWS Network Firewall を使用して、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フィルタリングすることにより、VPC を監視および保護します。詳細については、「AWS Network Firewall ガイド」を参照してください。
- Amazon GuardDuty は、AWS 環境内のアカウント、コンテナ、ワークロード、データに対する潜在的な脅威を特定するために使用します。基本的な脅威検出には、Amazon EC2 インスタンスに関連付けられた VPC フローログのモニタリングが含まれます。詳細については、「Amazon GuardDuty ユーザーガイド」の「VPC Flow Logs」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への回答については、「Amazon VPC のよくある質問」の「セキュリティとフィルタリング」を参照してください。

他の AWS のサービス とともに Amazon VPC を使用する

Amazon Virtual Private Cloud (VPC) は、クラウドインフラストラクチャに安全でカスタマイズ可能なネットワーク環境を提供する基本の AWS サービスです。独自の VPC を作成して管理することができるだけでなく、VPC と他の AWS サービスを連携させて、特定のニーズに合わせた総合的なソリューションを構築することができます。

VPC は AWS PrivateLink を使用してさまざまな AWS サービスに接続することができます。それにより、VPC と、サポートされている AWS サービスまたはオンプレミスアプリケーション間をプライベートに接続し、ネットワークトラフィックを AWS ネットワーク内に留め、パブリックインターネットに公開されないようにすることができます。このことはとりわけ、厳格なセキュリティの境界とコンプライアンス要件を維持するのに有用です。

VPC のセキュリティをさらに強化したいときは AWS Network Firewall を使用します。このマネージドのファイアウォールサービスを使用すれば、ネットワークレベルのセキュリティポリシーを定義および適用し、VPC 内の North-South と East-West 両方のトラフィックをフィルタリングすることができます。Network Firewall を VPC と組み合わせることにより、防御戦略を強化して不正アクセスや悪意のある行為からクラウドリソースを保護することができます。

さらに、Route 53 Resolver DNS Firewall を使用して、自社の VPC 内の DNS トラフィックをフィルタリングすることができます。この機能を使用すると、VPC リソースによって解決可能なドメインを制御する、カスタムの DNS フィルタリングルールを作成して、セキュリティとコンプライアンスをさらに強化することができます。

VPC 内のリソース間または VPC に接続されたリソース間で到達可能性に関する問題が発生した場合は、Reachability Analyzer を使用します。Reachability Analyzer は、仮想接続テストを実行してホップバイホップパスの詳細情報を提供し、阻害しているコンポーネントがないか確認します。このトラブルシューティングツールは、ネットワーク接続の問題をすばやく特定して解決するのに役立ちます。

こうした補完的な AWS サービスを VPC に連携させることで、自社独自のビジネス要件やアーキテクチャ要件に対応した、強力で安全かつ回復力の高いクラウドソリューションを構築することができます。

内容

- AWS PrivateLink を使用して VPC をサービスに接続する
- AWS Network Firewall を使用してネットワークトラフィックをフィルタリングする

- Route 53 Resolver DNS Firewall を使用して DNS トラフィックをフィルタリングする
- Reachability Analyzer を使用した到達可能性に関する問題のトラブルシューティング

AWS PrivateLink を使用して VPC をサービスに接続する

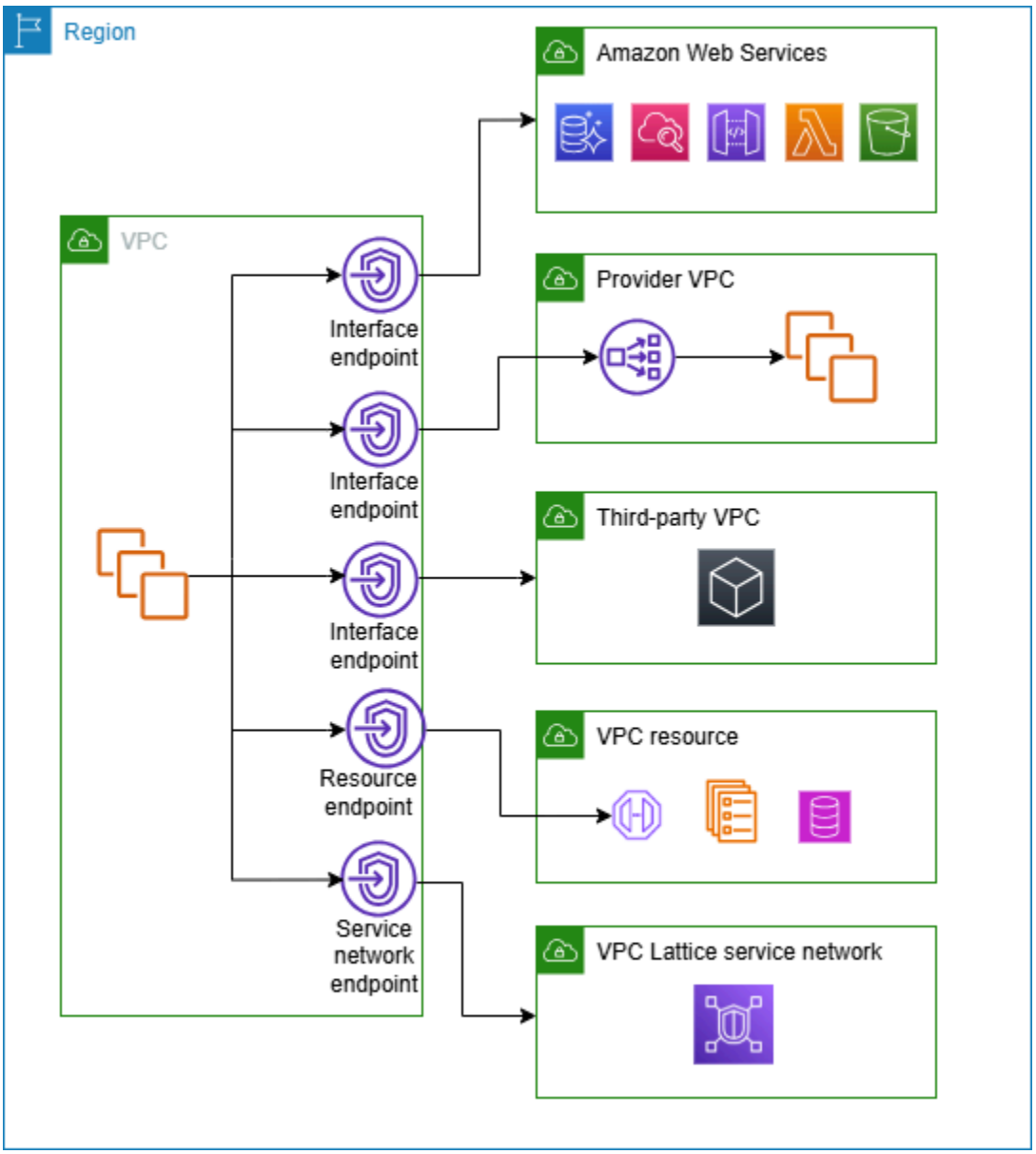
AWS PrivateLink は、仮想プライベートクラウド (VPC) とサポートされている AWS のサービス、他の AWS アカウント によってホストされているサービス、およびサポートされている AWS Marketplace のサービスとサポートされているリソース間のプライベート接続を確立します。サービスまたはリソースと通信するために、インターネットゲートウェイ、NAT デバイス、Direct Connect 接続、および AWS Site-to-Site VPN と接続する必要はありません。

AWS PrivateLink を使用するには、サービスまたはリソースにアクセスする必要があるサブネットに VPC エンドポイントを作成します。これによって Elastic Network Interface が指定したサブネットに作成され、これがサービスまたはリソースへのトラフィックのエントリポイントとなります。

また、AWS PrivateLink を使用した独自の VPC エンドポイントサービスを作成し、他の AWS の顧客にこのサービスへのアクセスを許可することもできます。PrivateLink を使用すると、プライベート API エンドポイントを作成して、自社独自のサービスを他の AWS の顧客に安全に公開することができます。これにより企業は、自社内の機能を収益化し、協働的なエコシステムを育み、サービスのアクセス方法および使用方法をコントロールすることができます。

AWS PrivateLink を使用する主な利点の一つは、従来のネットワーク構築 (インターネットゲートウェイ、NAT デバイス、VPN 接続など) を使用しなくても安全なプライベート接続を確立できることです。これにより、データトラフィックを AWS ネットワーク内に留めて、ネットワークアーキテクチャを簡素化しアタックサーフェスを狭め、全体的なセキュリティを向上させることができます。

以下の図は、AWS PrivateLink の一般的なユースケースを示したものです。VPC には、5 つの VPC エンドポイントを介してリソースにアクセスできるプライベートサブネットに複数の EC2 インスタンスがあります。3 つのインターフェイス VPC エンドポイント、1 つのリソース VPC エンドポイント、1 つのサービスネットワーク VPC エンドポイントがあります。



詳細については、 を参照してください。AWS PrivateLink

AWS Network Firewall を使用してネットワークトラフィックをフィルタリングする

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 を使用して DNS トラフィックをフィルタリングする

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。

DNS ファイアウォールリソース	説明
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。

Reachability Analyzer を使用した到達可能性に関する問題のトラブルシューティング

Reachability Analyzer は静的な設定分析ツールです。Reachability Analyzer を使用して、VPC 内の 2 つのリソース間のネットワーク到達可能性を分析およびデバッグできます。Reachability Analyzer は、これらのリソースに到達可能な場合は、リソース間にある仮想パスのホップバイホップの詳細を生成し、そうでない場合はブロッキングコンポーネントを識別します。

Reachability Analyzer を使用して、以下のリソース間の到達可能性を分析できます。

- インスタンス
- インターネットゲートウェイ
- ネットワークインターフェイス
- Transit Gateway
- Transit Gateway アタッチメント
- VPC エンドポイントサービス
- VPC エンドポイント
- VPC ピアリング接続
- VPN ゲートウェイ

詳細については、「Reachability Analyzer Guide」(到達可能性アナライザーガイド) を参照してください。

VPC の例

Amazon Virtual Private Cloud (VPC) は AWS エコシステム内の基本の構成要素であり、分離された仮想ネットワークを特定のニーズに合わせてプロビジョニングすることができます。独自の VPC を作成し管理することで、IP アドレス範囲、サブネット、ルーティングテーブル、接続オプションを定義する機能を含めネットワーク環境を完全に制御できます。

このセクションでは、仮想プライベートクラウド (VPC) の 3 つの設定例を取りあげます。それぞれ異なる要件に対応するように設計されています。

- テスト環境用の VPC: ここでは、開発環境またはテスト環境として使用できる VPC の作成方法を紹介します。
- ウェブサーバーとデータベースサーバー用の VPC: ここでは、本番環境での回復力のあるアーキテクチャに使用できる VPC の作成方法を紹介します。
- プライベートサブネットと NAT にサーバーを持つ VPC: こちらはより高度な設定で、EC2 インスタンスはすべてプライベートサブネット内にプロビジョニングされ、NAT ゲートウェイにより安全なアウトバウンドインターネットアクセスを促します。こちらは、必要なアウトバウンド通信を有効にしたまま、リソースへの直接的なインターネット接続を制限する必要がある場合の例です。

これらの VPC 設定の例を通じて、柔軟性の高さと、クラウドネットワーク環境を設計する際に使用できるカスタマイズのオプションについて、理解を深めていただきたいと考えています。選択する VPC 設定は、アプリケーションのアーキテクチャ、セキュリティ要件、全般的なビジネス目標に基づいている必要があります。細心の注意を払って VPC インフラストラクチャの計画を立てることで、クラウドベースのワークロードの成長と進化をサポートする、堅牢でスケーラブルかつ安全な仮想ネットワークを構築できます。

例

- 例: テスト環境の VPC
- 例: ウェブサーバーとデータベースサーバーの VPC
- 例: プライベートサブネットにサーバーがある VPC および NAT

関連する例

- VPC を相互に接続するには、「Amazon VPC ピアリングガイド」の「VPC ピア機能の設定」を参照してください。

- VPC を独自のネットワークに接続するには、AWS Site-to-Site VPN ユーザーガイドの「Site-to-Site VPN scenarios」を参照してください。
- VPC を相互に、および、独自のネットワークに接続するには、Amazon VPC Transit Gateway の「Example transit gateway scenarios」を参照してください。

追加リソース

- レジリエンシーのパターンとトレードオフを理解する (AWS アーキテクチャブログ)
- ネットワークトポロジーの計画 (AWS Well-Architected フレームワーク)
- Amazon Virtual Private Cloud の接続オプション (AWS ホワイトペーパー)

例: テスト環境の VPC

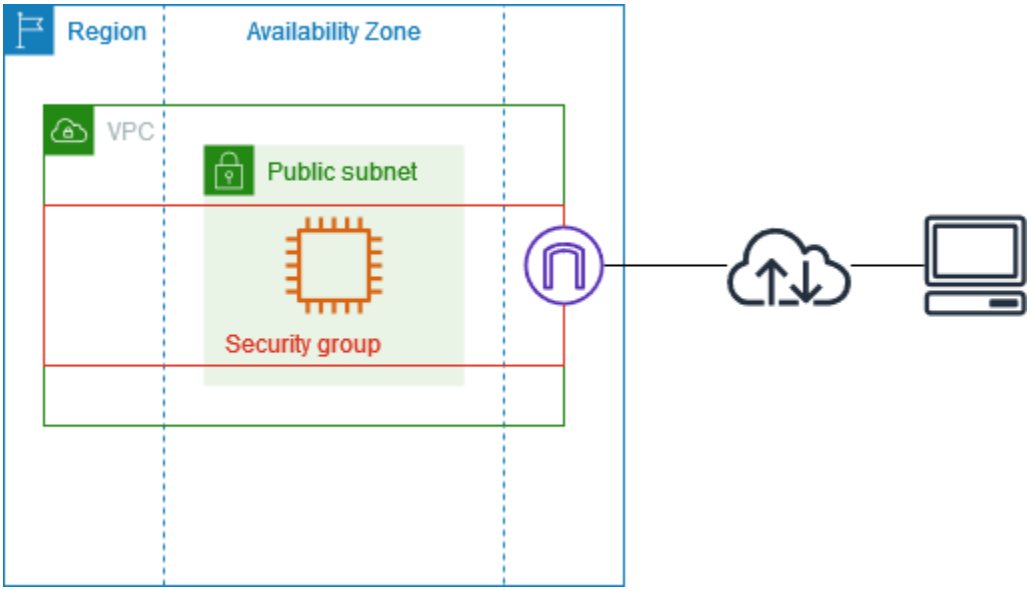
この例は、開発環境またはテスト環境として使用できる VPC を作成する方法について説明しています。この VPC は、本番環境での使用を目的とするものではないため、サーバーを複数のアベイラビリティーゾーンにデプロイする必要はありません。コストを抑えて複雑さを解消するには、サーバーを単一のアベイラビリティーゾーンにデプロイするのがよいでしょう。

内容

- 概要:
- 1. VPC を作成する
- 2. アプリケーションをデプロイします
- 3. 設定をテストする
- 4. クリーンアップ

概要:

次の図は、この例に含まれるリソースの概要を示しています。VPC には、1 つのアベイラビリティーゾーンとインターネットゲートウェイにパブリックサブネットがあります。サーバーはパブリックサブネットで動作する EC2 インスタンスです。インスタンスのセキュリティグループでは、自分のコンピュータからの SSH トラフィックに加えて、特に開発またはテストアクティビティに必要なその他のトラフィックも許可されます。



ルーティング

Amazon VPC コンソールを使用してこの VPC を作成すると、ローカルルートとインターネットゲートウェイへのルートを含む、パブリックサブネットのルートテーブルが作成されます。IPv4 と IPv6 の両方のルートを含むルートテーブルの例を次に示します。デュアルスタックサブネットの代わりに IPv4 専用サブネットを作成した場合、ルートテーブルには IPv4 ルートのみが含まれます。

ルーティング先	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*

セキュリティ

この設定例では、アプリケーションに必要な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する必要があります。例えば、コンピュータからの SSH トラフィックやネットワークからの HTTP トラフィックを許可するルールを、追加する必要があるとします。

以下は、IPv4 と IPv6 の両方のルールを含む、セキュリティグループのインバウンドルールの例です。デュアルスタックサブネットの代わりに IPv4 専用サブネットを作成する場合、必要となるのは IPv4 のルールのみです。

ソース	プロトコル	ポート範囲	説明
0.0.0.0/0	TCP	80	すべての IPv4 アドレスからのインバウンド HTTP アクセスを許可する。
::/0	TCP	80	すべての IPv6 アドレスからのインバウンド HTTP アクセスを許可する
0.0.0.0/0	TCP	443	すべての 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	すべての IPv6 アドレスからのインバウンド HTTPS アクセスを許可する
########## IPv4 ######	TCP	22	(オプション) ネットワーク内の IPv4 IP アドレスからのインバウンド SSH アクセスを許可する
####### IPv6 ##### #	TCP	22	(オプション) ネットワーク内の IPv6 IP アドレスからのインバウンド SSH アクセスを許可する
########## IPv4 ######	TCP	3389	(オプション) ネットワーク内の IPv4 IP アドレスからのインバウンド RDP アクセスを許可する
####### IPv6 ##### #	TCP	3389	(オプション) ネットワーク内の IPv6 IP アドレスからのインバウンド RDP アクセスを許可する

1. VPC を作成する

以下の手順を使用して、1 つのアベイラビリティーゾーンにパブリックサブネットを持つ VPC を作成します。この設定は、開発環境またはテスト環境に適しています。

VPC を作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ダッシュボードで、[VPC を作成] を選択します。
3. [Resources to create] (作成するリソース) で、[VPC and more] (VPC など) を選択します。
4. VPC を設定する
   a. [名前タグの自動生成] に、VPC の名前を入力します。
   b. [IPv4 CIDR ブロック] で、デフォルトの候補を維持するか、アプリケーションまたはネットワークが必要とする CIDR ブロックを入力します。詳細については、「the section called “VPC CIDR ブロック”」を参照してください。
   c. (オプション) アプリケーションが IPv6 アドレスを使用して通信する場合は、[IPv6 CIDR ブロック]、[Amazon が提供する IPv6 CIDR ブロック] を選択します。
5. サブネットを設定する
   a. [アベイラビリティーゾーンの数] で、[1] を選択します。デフォルトのアベイラビリティーゾーンをそのまま使用することも、[AZ のカスタマイズ] を展開してアベイラビリティーゾーンを選択することもできます。
   b. [Number of public subnets] (パブリックサブネットの数) には、[1] を選択します。
   c. [Number of private subnets] (プライベートサブネットの数) には、[0] を選択します。
   d. パブリックサブネットのデフォルトの CIDR ブロックをそのまま使用することも、[サブネット CIDR ブロックをカスタマイズする] を展開して CIDR ブロックを入力することもできます。詳細については、「the section called “サブネット CIDR ブロック”」を参照してください。
6. [NAT ゲートウェイ] は、デフォルト値の [なし] のままにします。
7. [VPC エンドポイント] には、[なし] を選択します。S3 のゲートウェイ VPC エンドポイントは、プライベートサブネットから Amazon S3 にアクセスする場合にのみ使用します。
8. [DNS オプション] で、両方のオプションを選択したままにします。これにより、インスタンスはパブリック IP アドレスに対応するパブリック DNS ホスト名を受け取ります。
9. [Create VPC (VPC の作成)] を選択します。

2. アプリケーションをデプロイします

EC2 インスタンスをデプロイするには、さまざまな方法があります。例えば、次のようになります。

- Amazon EC2 インスタンス起動ウィザード
- Amazon EC2 Auto Scaling
- AWS CloudFormation
- Amazon Elastic Container Service (Amazon ECS)

EC2 インスタンスをデプロイしたら、インスタンスに接続し、アプリケーションに必要なソフトウェアをインストールして、後で使用するためのイメージを作成します。詳細については、「Amazon EC2 ユーザーガイド」の「AMI を作成する」を参照してください。または、EC2 Image Builder を使用して、Amazon マシンイメージ (AMI) を作成、管理することもできます。

3. 設定をテストする

アプリケーションのデプロイが完了したら、それをテストできます。EC2 インスタンスに接続できない場合、またはアプリケーションが想定どおりのトラフィックを送受信できない場合は、Reachability Analyzer を使用してトラブルシューティングを行います。例えば、Reachability Analyzer は、ルートテーブルやセキュリティグループの設定上の問題を特定できます。詳細については、「Reachability Analyzer Guide」(到達可能性アナライザーガイド) を参照してください。

4. クリーンアップ

不要になった設定は、削除できます。VPC を削除する前に、インスタンスを終了する必要があります。詳細については、「the section called “VPC の削除”」を参照してください。

例: ウェブサーバーとデータベースサーバーの VPC

この例は、本番環境の 2 層アーキテクチャに使用できる VPC を作成する方法について説明しています。回復性を高めるには、サーバーを 2 つのアベイラビリティーゾーンにデプロイします。

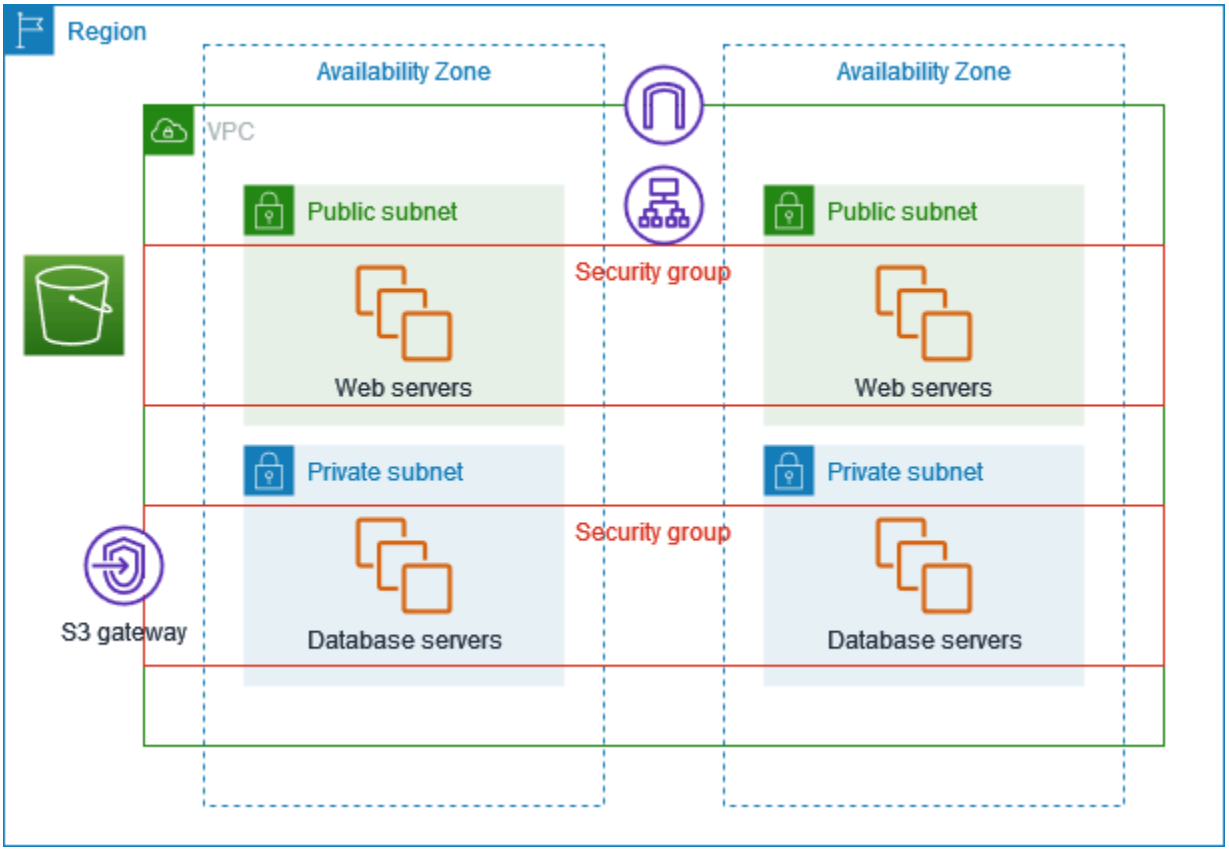
内容

- 概要:
- 1. VPC を作成する

- 2. アプリケーションをデプロイします
- 3. 設定をテストする
- 4. クリーンアップ

概要:

次の図は、この例に含まれるリソースの概要を示しています。VPC には、2 つのアベイラビリティーゾーン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。ウェブサーバーはパブリックサブネットで動作し、ロードバランサーを介してクライアントからのトラフィックを受信します。ウェブサーバーのセキュリティグループは、ロードバランサーからのトラフィックを許可します。データベースサーバーはプライベートサブネットで動作し、ウェブサーバーからのトラフィックを受信します。データベースサーバーのセキュリティグループは、ウェブサーバーからのトラフィックを許可します。データベースサーバーは、ゲートウェイ VPC エンドポイントを使用して Amazon S3 に接続できます。



ルーティング

Amazon VPC コンソールを使用してこの VPC を作成すると、ローカルルートとインターネットゲートウェイへのルートを含むパブリックサブネットのルートテーブル、およびローカルルートとゲート

ウェイ VPC エンドポイントへのルートを含む各プライベートサブネットのルートテーブルが作成されます。

以下は、IPv4 と IPv6 の両方のルートを含むパブリックサブネットのルートテーブルの例です。デュアルスタックサブネットの代わりに IPv4 専用サブネットを作成した場合、ルートテーブルには IPv4 ルートのみが含まれます。

ルーティング先	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*

以下は、IPv4 と IPv6 の両方のルートを含む、プライベートサブネットのルートテーブルの例です。IPv4 専用サブネットを作成した場合、ルートテーブルには IPv4 ルートのみが含まれます。最後のルートは、Amazon S3 を宛先とするトラフィックをゲートウェイ VPC エンドポイントに送信します。

ルーティング先	ターゲット
10.0.0.0/16	ローカル
2001:db8:1234:1a00::/56	ローカル
s3-prefix-list-id	*s3-gateway-id*

セキュリティ

この設定例では、ロードバランサー用のセキュリティグループ、ウェブサーバー用のセキュリティグループとデータベースサーバー用のセキュリティグループを作成します。

ロードバランサー

Application Load Balancer または Network Load Balancer のセキュリティグループは、ロードバランサーのリスナーポート上のクライアントからのインバウンドトラフィックを許可する必要がありま

す。インターネット上のどこからでもトラフィックを受け入れるには、ソースとして 0.0.0.0/0 を指定します。また、ロードバランサーセキュリティグループでは、ロードバランサーから、インスタンスリスナーポートおよびヘルスチェックポートでのターゲットインスタンスへのアウトバウンドトラフィックを許可する必要もあります。

ウェブサーバー

以下のセキュリティグループルールは、ロードバランサーから HTTP および HTTPS トラフィックを受信することをウェブサーバーに許可します。(オプション) ウェブサーバーがネットワークから SSH または RDP トラフィックを受信するように許可できます。ウェブサーバーから 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。

ソース	プロトコル	ポート範囲	説明
########## #### ID	TCP	80	ロードバランサーからのインバウンド HTTP アクセスを許可する
########## #### ID	TCP	443	ロードバランサーからのインバウンド HTTPS アクセスを許可する
########## IPv4 ######	TCP	22	(オプション) ネットワーク内の IPv4 IP アドレスからのインバウンド SSH アクセスを許可する
####### IPv6 ##### #	TCP	22	(オプション) ネットワーク内の IPv6 IP アドレスからのインバウンド SSH アクセスを許可する
########## IPv4 ######	TCP	3389	(オプション) ネットワーク内の IPv4 IP アドレスからのインバウンド RDP アクセスを許可する
####### IPv6 ##### #	TCP	3389	(オプション) ネットワーク内の IPv6 IP アドレスからのインバウンド RDP アクセスを許可する

目的地	プロトコル	ポート範囲	説明
Microsoft SQL Server ######### ################ ID	TCP	1433	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
MySQL ########### ############## ID	TCP	3306	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

データベースサーバー

次のセキュリティグループルールにより、データベースがウェブサーバーから、読み込みおよび書き込みリクエストを受信できます。

ソース	プロトコル	ポート範囲	コメント
################ # ID	TCP	1433	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
################ # ID	TCP	3306	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

目的地	プロトコル	ポート範囲	コメント
0.0.0.0/0	TCP	80	IPv4 を介して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	IPv4 を介して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

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。

1. VPC を作成する

次の手順を使用して、2 つのアベイラビリティーゾーンに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。

VPC を作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ダッシュボードで、[VPC を作成] を選択します。
3. [Resources to create] (作成するリソース) で、[VPC and more] (VPC など) を選択します。
4. VPC を設定する:
   a. [名前タグの自動生成] を選択したままにすると VPC リソース用の名前タグが作成され、オフにすると VPC リソース用の独自の名前タグが提供されます。
   b. [IPv4 CIDR ブロック] で、デフォルトの候補を維持するか、アプリケーションまたはネットワークが必要とする CIDR ブロックを入力します。詳細については、「the section called "VPC CIDR ブロック"」を参照してください。
   c. (オプション) アプリケーションが IPv6 アドレスを使用して通信する場合は、[IPv6 CIDR ブロック]、[Amazon が提供する IPv6 CIDR ブロック] を選択します。
   d. [テナンシー] を選択します 。このオプションは、VPC で起動する EC2 インスタンスを、他の AWS アカウント と共有しているハードウェアで実行するか、または自分専用のハードウェアで実行するかを定義します。VPC のテナンシーとして `Default` を選択すると、この VPC で起動した EC2 インスタンスは、インスタンスの起動時に指定したテナンシー属性を使用します。詳細については、「Amazon EC2 ユーザーガイド」の「定義済みのパラメータを使用したインスタンスの起動」を参照してください。VPC のテナンシーで `Dedicated` を選択すると、インスタンスは常に、ユーザー専用のハードウェアで実行される、専有インスタンスとして実行されます。
5. サブネットを設定する:
   a. [アベイラビリティーゾーンの数] で [2] を選択すると、2 つのアベイラビリティーゾーンでインスタンスを起動し、回復性を高めることができます。
   b. [Number of public subnets] (パブリックサブネットの数) で 2 を選択します。
   c. [Number of private subnets] (プライベートサブネットの数) は、2 を選択します。

d. サブネットのデフォルトの CIDR ブロックをそのまま使用したり、[サブネット CIDR ブロックをカスタマイズ] を開いて CIDR ブロックを入力したりすることができます。詳細については、「the section called “サブネット CIDR ブロック”」を参照してください。

6. [NAT ゲートウェイ] は、デフォルト値の [なし] のままにします。
7. [VPC エンドポイント] は、デフォルト値の [S3 ゲートウェイ] のままにします。S3 バケットにアクセスしない限り効果はありませんが、この VPC エンドポイントの有効化にコストはかかりません。
8. [DNS オプション] で、両方のオプションを選択したままにします。これにより、ウェブサーバーは、パブリック IP アドレスに対応するパブリック DNS ホスト名を受け取ります。
9. [Create VPC (VPC の作成)] を選択します。

2. アプリケーションをデプロイします

開発環境またはテスト環境でのウェブサーバーおよびデータベースサーバーをテスト済みで、アプリケーションを本番環境にデプロイするために使用するスクリプトまたはイメージを作成済みであることが理想的です。

ウェブサーバーには EC2 インスタンスを使用できます。EC2 インスタンスをデプロイするには、さまざまな方法があります。例えば、次のようになります。

- Amazon EC2 インスタンス起動ウィザード
- AWS CloudFormation
- Amazon Elastic Container Service (Amazon ECS)

可用性を高めるには、Amazon EC2 Auto Scaling を使用して複数のアベイラビリティーゾーンにサーバーをデプロイし、アプリケーションに必要な最小限のサーバー容量を維持します。

Elastic Load Balancing を使用すると、トラフィックをサーバー全体に均等に分散できます。Auto Scaling グループにロードバランサーをアタッチできます。

データベースサーバーには EC2 インスタンスを使用するか、目的別データベースタイプのいずれかを使用できます。詳細については、「AWS のデータベース: 選択方法」を参照してください。

3. 設定をテストする

アプリケーションのデプロイが完了したら、それをテストできます。アプリケーションが想定どおりのトラフィックを送受信できない場合は、Reachability Analyzer を使用してトラブルシューティング

を行います。例えば、Reachability Analyzer は、ルートテーブルやセキュリティグループの設定上の問題を特定できます。詳細については、「Reachability Analyzer Guide」(到達可能性アナライザーガイド) を参照してください。

4. クリーンアップ

不要になった設定は、削除できます。VPC を削除する前に、インスタンスを終了し、ロードバランサーを削除する必要があります。詳細については、「the section called “VPC の削除”」を参照してください。

例: プライベートサブネットにサーバーがある VPC および NAT

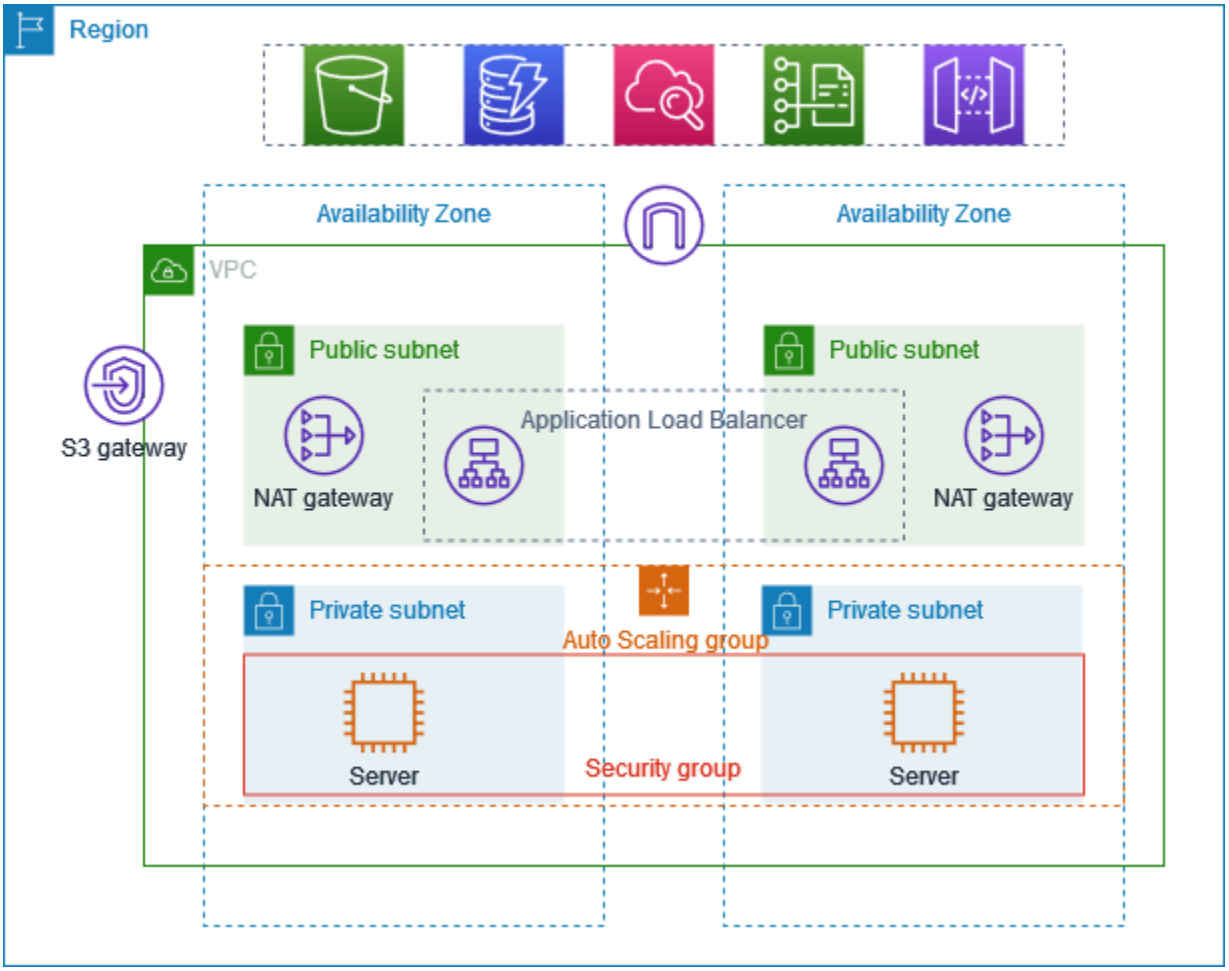
この例は、本番環境のサーバーに使用できる VPC を作成する方法について説明しています。回復性を高めるには、Auto Scaling グループと Application Load Balancer を使用してサーバーを 2 つのアベイラビリティーゾーンにデプロイします。セキュリティを強化するために、サーバーをプライベートサブネットにデプロイします。サーバーはロードバランサーを介してリクエストを受信します。サーバーは、NAT ゲートウェイを使用してインターネットに接続できます。回復性を高めるには、NAT ゲートウェイを両方のアベイラビリティーゾーンにデプロイします。

内容

- 概要
- 1. VPC を作成する
- 2. アプリケーションをデプロイします
- 3. 設定をテストする
- 4. クリーンアップ

概要

次の図は、この例に含まれるリソースの概要を示しています。VPC には、2 つのアベイラビリティーゾーン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。各パブリックサブネットには、NAT ゲートウェイとロードバランサーノードが含まれています。サーバーはプライベートサブネットで実行され、Auto Scaling グループを使用して起動および終了し、ロードバランサーからトラフィックを受信します。サーバーは、NAT ゲートウェイを使用してインターネットに接続できます。サーバーは、ゲートウェイ VPC エンドポイントを使用して Amazon S3 に接続できます。



ルーティング

Amazon VPC コンソールを使用してこの VPC を作成すると、ローカルルートとインターネットゲートウェイへのルートを含むパブリックサブネットのルートテーブルが作成されます。また、ローカルルートを含むプライベートサブネットのルートテーブルと、NAT ゲートウェイ、下り専用インターネットゲートウェイ、ゲートウェイ VPC エンドポイントへのルートが作成されます。

以下は、IPv4 と IPv6 の両方のルートを含むパブリックサブネットのルートテーブルの例です。デュアルスタックサブネットの代わりに IPv4 専用サブネットを作成した場合、ルートテーブルには IPv4 ルートのみが含まれます。

ルーティング先	ターゲット
10.0.0.0/16	ローカル
2001:db8:1234:1a00::/56	ローカル

ルーティング先	ターゲット
0.0.0.0/0	*igw-id*
::/0	*igw-id*

以下は、IPv4 と IPv6 の両方のルートを含む、いずれかのプライベートサブネットのルートテーブルの例です。IPv4 専用サブネットを作成した場合、ルートテーブルには IPv4 ルートのみが含まれます。最後のルートは、Amazon S3 を宛先とするトラフィックをゲートウェイ VPC エンドポイントに送信します。

ルーティング先	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	*eigw-id*
s3-prefix-list-id	*s3-gateway-id*

セキュリティ

以下は、サーバーに関連付けるセキュリティグループ用に作成できるルールの例です。セキュリティグループは、リスナーポートとプロトコルを経由するロードバランサーからのトラフィックを許可する必要があります。ヘルスチェックトラフィックも許可する必要があります。

ソース	プロトコル	ポート範囲	コメント
################ #### ID	*#########*	*#######*	リスナーポートのロードバランサーからのインバウンドトラフィックを許可する

ソース	プロトコル	ポート範囲	コメント
################ #### ID	########## ##	##########	ロードバランサーからのインバウンドヘルスチェックトラフィックを許可する

1. VPC を作成する

次の手順を使用して、2 つのアベイラビリティーゾーン、および各アベイラビリティーゾーンの NAT ゲートウェイで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。

VPC を作成するには

1. Amazon VPC コンソールの https://console.aws.amazon.com/vpc/ を開いてください。
2. ダッシュボードで、[VPC を作成] を選択します。
3. [Resources to create] (作成するリソース) で、[VPC and more] (VPC など) を選択します。
4. VPC を設定する
   a. [名前タグの自動生成] に、VPC の名前を入力します。
   b. [IPv4 CIDR ブロック] で、デフォルトの候補を維持するか、アプリケーションまたはネットワークが必要とする CIDR ブロックを入力します。
   c. アプリケーションが IPv6 アドレスを使用して通信する場合は、[IPv6 CIDR ブロック]、[Amazon が提供する IPv6 CIDR ブロック] を選択します。
5. サブネットを設定する
   a. [アベイラビリティーゾーンの数] で [2] を選択すると、複数のアベイラビリティーゾーンでインスタンスを起動し、回復性を改善できます。
   b. [Number of public subnets] (パブリックサブネットの数) で 2 を選択します。
   c. [Number of private subnets] (プライベートサブネットの数) は、2 を選択します。
   d. パブリックサブネットのデフォルトの CIDR ブロックをそのまま使用することも、[サブネット CIDR ブロックをカスタマイズする] を展開して CIDR ブロックを入力することもできます。詳細については、「the section called “サブネット CIDR ブロック”」を参照してください。
6. [NAT ゲートウェイ] で [AZ ごとに 1] を選択すると、回復性が高まります。

7. アプリケーションが IPv6 アドレスを使用して通信する場合、[Egress Only インターネットゲートウェイ] で [はい] を選択します。
8. インスタンスが S3 バケットにアクセスする必要がある場合は、[VPC エンドポイント] を、デフォルトの [S3 ゲートウェイ] のままにします。デフォルトのままにしないと、プライベートサブネットのインスタンスが Amazon S3 にアクセスできません。このオプションはコストがかからないため、今後、S3 バケットを使用する可能性がある場合は、デフォルトのままにしておくことができます。[なし] を選択した場合、後からいつでもゲートウェイ VPC エンドポイントを追加できます。
9. [DNS オプション] で、[DNS ホスト名を有効化] をオフにします。
10. [Create VPC (VPC の作成)] を選択します。

2. アプリケーションをデプロイします

開発環境またはテスト環境でのサーバーのテストを完了し、アプリケーションを本番環境にデプロイするために使用するスクリプトまたはイメージを作成済みであることが理想的です。

Amazon EC2 Auto Scaling を使用すると、複数のアベイラビリティーゾーンにサーバーをデプロイし、アプリケーションに必要な最小限のサーバー容量を維持できます。

Auto Scaling グループを使用してインスタンスを起動するには

1. Amazon EC2 Auto Scaling を使用して EC2 インスタンスを起動する際に必要になる、設定情報を指定するための起動テンプレートを作成します。詳細な手順については、「Amazon EC2 Auto Scaling ユーザーガイド」の「Create a launch template for your Auto Scaling group」を参照してください。
2. 最小サイズ、最大サイズ、必要なサイズを持つ EC2 インスタンスのコレクションである Auto Scaling グループを作成します。詳細な手順については、「Amazon EC2 Auto Scaling ユーザーガイド」の「Create an Auto Scaling group using a launch template」を参照してください。
3. Auto Scaling グループ内のインスタンス全体にトラフィックを均等に分散するロードバランサーを作成し、Auto Scaling グループにアタッチします。詳細については、「Elastic Load Balancing ユーザーガイド」および「Amazon EC2 Auto Scaling ユーザーガイド」の「Use Elastic Load Balancing」を参照してください。

3. 設定をテストする

アプリケーションのデプロイが完了したら、それをテストできます。アプリケーションが想定どおりのトラフィックを送受信できない場合は、Reachability Analyzer を使用してトラブルシューティングを行います。例えば、Reachability Analyzer は、ルートテーブルやセキュリティグループの設定上の問題を特定できます。詳細については、「Reachability Analyzer Guide」(到達可能性アナライザーガイド) を参照してください。

4. クリーンアップ

不要になった設定は、削除できます。VPC を削除する前に、Auto Scaling グループを削除してインスタンスを終了し、NAT ゲートウェイを削除してロードバランサーを削除する必要があります。詳細については、「the section called “VPC の削除”」を参照してください。

VPC チュートリアル

Amazon 仮想プライベートクラウド (VPC) は、AWS のネットワークインフラストラクチャの基盤です。AWS マネジメントコンソールはユーザーフレンドリーなインターフェイスを提供しますが、AWS コマンドラインインターフェイス (CLI) は、VPC リソースの作成および管理において、より優れた柔軟性と自動化機能を提供します。

このガイドでは、2 つの重要な VPC デプロイシナリオを示します。

- シンプルなウェブアプリケーションに最適なパブリックサブネットを使用した基本的な VPC セットアップ
- NAT ゲートウェイを使用するプライベートサブネットとパブリックサブネットの両方を使用した高度な VPC 設定。多層アプリケーションに適しています

これらのチュートリアルに従うことで、以下の方法を学ぶことができます。

- サブネット設定が異なる VPC を作成する
- インターネットゲートウェイと NAT ゲートウェイをセットアップする
- ルートテーブルとセキュリティグループを設定する
- AWS CLI コマンドを使用してネットワークインフラストラクチャを管理する
- AWS ネットワーキングのベストプラクティスを実装する

コマンドラインを使用して AWS ネットワークインフラストラクチャの構築を始めましょう。

チュートリアル

- AWS CLI を使用した Amazon VPC の開始方法
- AWS CLI を使用してプライベートサブネットおよび NAT ゲートウェイを持つ VPC を作成する

AWS CLI を使用した Amazon VPC の開始方法

このチュートリアルでは、AWS コマンドラインインターフェイス (AWS CLI) を使用して仮想プライベートクラウド (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とプライベートサブネットを使用して VPC をセットアップする方法、インターネット接続を設定する方法、EC2 インスタンスをデプロイして一般的なウェブアプリケーションアーキテクチャをデモンストレーションする方法について説明します。

前提条件

このチュートリアルを開始する前に、以下の要件を満たしていることを確認してください。

1. AWS CLI。インストールする必要がある場合は、「AWS CLI の最新バージョンのインストールまたは更新」に従ってください。
2. AWS CLI が適切な認証情報で設定されている。認証情報をまだ設定していない場合は、`aws configure` を実行します。
3. ネットワーク概念の基本的な理解。
4. AWS アカウントで VPC リソースを作成および管理するための Amazon VPC の Identity and Access Management。

コストに関する考慮事項

このチュートリアルでは、アカウントでコストが発生する可能性のある AWS リソースを作成します。主なコストは、NAT ゲートウェイ (1 時間あたり 0.045 USD とデータ処理料金) と EC2 インスタンス (t2.micro、それぞれ 1 時間あたり約 0.0116 USD) です。このチュートリアルを 1 時間以内に完了し、すべてのリソースをクリーンアップすると、合計コストは約 0.07 USD になります。開発環境でコストを最適化するには、NAT ゲートウェイの代わりに NAT インスタンスを使用することを検討してください。これにより、コストを大幅に削減できます。

次に進む前に AWS CLI が正しく設定されていることを確認します。

```
aws configure list
```

AWS アクセスキー、シークレットキー、デフォルトリージョンが表示されます。また、VPC リソースを作成するために必要なアクセス許可があることを確認します。

```
aws sts get-caller-identity
```

このコマンドは、AWS アカウント ID、ユーザー ID、ARN を表示し、認証情報が有効であることを確認します。

「VPC を作成する」

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。このセクションでは、CIDR ブロックが 10.0.0.0/16 の VPC を作成します。これにより、最大 65,536 個の IP アドレスが提供されます。

VPC を作成する

次のコマンドは、新しい VPC を作成し、名前タグを割り当てます。

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications
 'ResourceType=vpc,Tags=[{Key=Name,Value=MyVPC}]'
```

出力の VPC ID を書き留めます。これは後続のコマンドで必要になります。このチュートリアルでは、VPC ID の例として「vpc-0123456789abcdef0」を使用します。すべてのコマンドでこれを実際の VPC ID に置き換えます。

DNS サポートとホスト名を有効にする

デフォルトでは、DNS 解決と DNS ホスト名は新しい VPC では無効になっています。これらの機能を有効にして、VPC 内のインスタンスがドメイン名を解決できるようにします。

```
aws ec2 modify-vpc-attribute --vpc-id vpc-0123456789abcdef0 --enable-dns-support
aws ec2 modify-vpc-attribute --vpc-id vpc-0123456789abcdef0 --enable-dns-hostnames
```

これらのコマンドは、成功した場合に出力を生成しません。VPC で DNS サポートとホスト名解決が有効になりました。

サブネットを作成する

サブネットは、分離されたリソースのグループを配置できる VPC の IP アドレス範囲のセグメントです。このセクションでは、高可用性を実現するために、2 つのアベイラビリティーゾーンに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。

利用可能なアベイラビリティーゾーンを取得する

まず、リージョンで使用可能なアベイラビリティーゾーンを取得します。

```
aws ec2 describe-availability-zones
```

このチュートリアルでは、最初の 2 つのアベイラビリティーゾーンを使用します。出力からその名前 (例:「us-east-1a」および「us-east-1b」) を書き留めます。

パブリックサブネットを作成する

パブリックサブネットは、ウェブサーバーなど、インターネットからアクセスする必要があるリソースに使用されます。

```
aws ec2 create-subnet \
  --vpc-id vpc-0123456789abcdef0 \
  --cidr-block 10.0.0.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Public-Subnet-AZ1}]'
```

出力のサブネット ID を書き留めます。このチュートリアルでは、最初のパブリックサブネットの例として「subnet-0123456789abcdef0」を使用します。

```
aws ec2 create-subnet \
  --vpc-id vpc-0123456789abcdef0 \
  --cidr-block 10.0.1.0/24 \
  --availability-zone us-east-1b \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Public-Subnet-AZ2}]'
```

出力のサブネット ID を書き留めます。このチュートリアルでは、2 番目のパブリックサブネットの例として「subnet-0123456789abcdef1」を使用します。

プライベートサブネットの作成

プライベートサブネットは、データベースなど、インターネットから直接アクセスできないリソースに使用されます。

```
aws ec2 create-subnet \
  --vpc-id vpc-0123456789abcdef0 \
  --cidr-block 10.0.2.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Private-Subnet-AZ1}]'
```

出力のサブネット ID を書き留めます。このチュートリアルでは、最初のプライベートサブネットの例として「subnet-0123456789abcdef2」を使用します。

```
aws ec2 create-subnet \
  --vpc-id vpc-0123456789abcdef0 \
  --cidr-block 10.0.3.0/24 \
  --availability-zone us-east-1b \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=Private-Subnet-AZ2}]'
```

出力のサブネット ID を書き留めます。このチュートリアルでは、2 番目のプライベートサブネットの例として「subnet-0123456789abcdef3」を使用します。

これで、2 つのアベイラビリティーゾーンに分散された 2 つのパブリックサブネットと 2 つのプライベートサブネットの 4 つのサブネットができました。

ヒント: CIDR ブロックを計画する際は、既存のネットワークと重複しないようにしてください。本番環境では、セキュリティと管理のためにサブネットを適切なサイズに保ちながら、将来の成長に備えて十分な IP アドレスを割り当てます。

インターネット接続を設定する

VPC 内のリソースがインターネットと通信できるようにするには、インターネットゲートウェイを作成してアタッチする必要があります。このセクションでは、VPC のインターネット接続を設定します。

インターネットゲートウェイを作成する

インターネットゲートウェイは、VPC とインターネット間の通信を可能にします。

```
aws ec2 create-internet-gateway \
  --tag-specifications 'ResourceType=internet-gateway,Tags=[{Key=Name,Value=MyIGW}]'
```

出力からインターネットゲートウェイ ID を書き留めます。このチュートリアルでは、例として「igw-0123456789abcdef0」を使用します。

VPC にインターネットゲートウェイをアタッチする

インターネットゲートウェイを作成後、それを VPC にアタッチします。

```
aws ec2 attach-internet-gateway --internet-gateway-id igw-0123456789abcdef0 --vpc-id
 vpc-0123456789abcdef0
```

ルートテーブルを作成および設定する

ルートテーブルには、ネットワークトラフィックの転送先を決定するルール (ルート) が含まれています。まず、パブリックサブネットのルートテーブルを作成します。

```
aws ec2 create-route-table \
  --vpc-id vpc-0123456789abcdef0 \
  --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=Public-RT}]'
```

出力のルートテーブル ID を書き留めます。このチュートリアルでは、パブリックルートテーブルの例として「rtb-0123456789abcdef0」を使用します。

パブリックルートテーブルでインターネットゲートウェイにルートを追加します。

```
aws ec2 create-route --route-table-id rtb-0123456789abcdef0 --destination-cidr-block
 0.0.0.0/0 --gateway-id igw-0123456789abcdef0
```

パブリックサブネットをパブリックルートテーブルに関連付けます。

```
aws ec2 associate-route-table --route-table-id rtb-0123456789abcdef0 --subnet-id
 subnet-0123456789abcdef0
aws ec2 associate-route-table --route-table-id rtb-0123456789abcdef0 --subnet-id
 subnet-0123456789abcdef1
```

次に、プライベートサブネットのルートテーブルを作成します。

```
aws ec2 create-route-table \
  --vpc-id vpc-0123456789abcdef0 \
  --tag-specifications 'ResourceType=route-table,Tags=[{Key=Name,Value=Private-RT}]'
```

出力のルートテーブル ID を書き留めます。このチュートリアルでは、プライベートルートテーブルの例として「rtb-0123456789abcdef1」を使用します。

プライベートサブネットをプライベートルートテーブルに関連付けます。

```
aws ec2 associate-route-table --route-table-id rtb-0123456789abcdef1 --subnet-id
 subnet-0123456789abcdef2
aws ec2 associate-route-table --route-table-id rtb-0123456789abcdef1 --subnet-id
 subnet-0123456789abcdef3
```

NAT ゲートウェイを作成する

NAT ゲートウェイを使用すると、プライベートサブネットのインスタンスは、インターネットからのインバウンドトラフィックを防ぎながら、インターネットへのアウトバウンドトラフィックを開始できます。これは、更新をダウンロードしたり、外部サービスにアクセスしたりする必要があるインスタンスに不可欠です。

Elastic IP を割り当てる

まず、NAT ゲートウェイ用の Elastic IP アドレスを割り当てます。

```
aws ec2 allocate-address --domain vpc
```

出力の割り当て ID を書き留めます。このチュートリアルでは、例として「eipalloc-0123456789abcdef0」を使用します。

NAT ゲートウェイを作成する

割り当てられた Elastic IP を使用して、いずれかのパブリックサブネットに NAT ゲートウェイを作成します。

```
aws ec2 create-nat-gateway \
  --subnet-id subnet-0123456789abcdef0 \
  --allocation-id eipalloc-0123456789abcdef0 \
  --tag-specifications 'ResourceType=natgateway,Tags=[{Key=Name,Value=MyNATGateway}]'
```

出力の NAT ゲートウェイ ID を書き留めます。このチュートリアルでは、例として「nat-0123456789abcdef0」を使用します。

NAT ゲートウェイが使用可能になるまで待ってから次に進みます。

```
aws ec2 wait nat-gateway-available --nat-gateway-ids nat-0123456789abcdef0
```

NAT ゲートウェイにルートを追加する

プライベートルートテーブルの NAT ゲートウェイにルートを追加して、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にアクセスできるようにします。

```
aws ec2 create-route --route-table-id rtb-0123456789abcdef1 --destination-cidr-block
 0.0.0.0/0 --nat-gateway-id nat-0123456789abcdef0
```

注: 本番環境では、単一障害点を排除するために、プライベートサブネットがある各アベイラビリティーゾーンに NAT ゲートウェイを作成することを検討してください。

サブネット設定を構成する

パブリックサブネットを設定して、その中で起動されたインスタンスにパブリック IP アドレスが自動的に割り当てられるようにします。

```
aws ec2 modify-subnet-attribute --subnet-id subnet-0123456789abcdef0 --map-public-ip-
on-launch
aws ec2 modify-subnet-attribute --subnet-id subnet-0123456789abcdef1 --map-public-ip-
on-launch
```

これにより、パブリックサブネットで起動されたインスタンスがデフォルトでパブリック IP アドレスを受信し、インターネットからアクセスできるようになります。

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、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制御します。このセクションでは、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を作成します。

ウェブサーバーのセキュリティグループを作成する

```
aws ec2 create-security-group \
  --group-name WebServerSG \
  --description "Security group for web servers" \
  --vpc-id vpc-0123456789abcdef0
```

出力のセキュリティグループ ID を書き留めます。このチュートリアルでは、ウェブサーバーセキュリティグループの例として「sg-0123456789abcdef0」を使用します。

ウェブサーバーへの HTTP および HTTPS トラフィックを許可します。

```
aws ec2 authorize-security-group-ingress --group-id sg-0123456789abcdef0 --protocol tcp
 --port 80 --cidr 0.0.0.0/0
aws ec2 authorize-security-group-ingress --group-id sg-0123456789abcdef0 --protocol tcp
 --port 443 --cidr 0.0.0.0/0
```

注: 本番環境では、0.0.0.0/0 (任意の IP アドレス) からのトラフィックを許可するのではなく、インバウンドトラフィックを特定の IP 範囲に制限します。

データベースサーバーのセキュリティグループを作成する

```
aws ec2 create-security-group \
  --group-name DBServerSG \
  --description "Security group for database servers" \
  --vpc-id vpc-0123456789abcdef0
```

出力のセキュリティグループ ID を書き留めます。このチュートリアルでは、データベースサーバーセキュリティグループの例として「sg-0123456789abcdef1」を使用します。

ウェブサーバーからの MySQL/Aurora トラフィックのみを許可します。

```
aws ec2 authorize-security-group-ingress --group-id sg-0123456789abcdef1 --protocol tcp
 --port 3306 --source-group sg-0123456789abcdef0
```

この設定により、最小特権の原則に従って、ウェブサーバーセキュリティグループ内のインスタンスのみがポート 3306 のデータベースサーバーに接続できるようになります。

VPC 設定を検証する

必要なコンポーネントをすべて作成したら、VPC 設定を確認して、すべてが正しく設定されていることを確認します。

VPC を確認する

```
aws ec2 describe-vpcs --vpc-id vpc-0123456789abcdef0
```

サブネットを確認する

```
aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-0123456789abcdef0"
```

ルートテーブルを確認する

```
aws ec2 describe-route-tables --filters "Name=vpc-id,Values=vpc-0123456789abcdef0"
```

インターネットゲートウェイを確認する

```
aws ec2 describe-internet-gateways --filters "Name=attachment.vpc-
id,Values=vpc-0123456789abcdef0"
```

NAT ゲートウェイを確認する

```
aws ec2 describe-nat-gateways --filter "Name=vpc-id,Values=vpc-0123456789abcdef0"
```

セキュリティグループを確認する

```
aws ec2 describe-security-groups --filters "Name=vpc-id,Values=vpc-0123456789abcdef0"
```

これらのコマンドは、VPC の各コンポーネントに関する詳細情報を提供し、すべてが正しく設定されていることを確認できます。

EC2 インスタンスをデプロイする

VPC インフラストラクチャを作成したので、EC2 インスタンスをデプロイしてアーキテクチャの仕組みをデモンストレーションできます。パブリックサブネットでウェブサーバーを起動し、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します。

SSH アクセス用のキーペアを作成する

まず、キーペアを作成してインスタンスに安全に接続します。

```
aws ec2 create-key-pair --key-name vpc-tutorial-key --query 'KeyMaterial' --output text
 > vpc-tutorial-key.pem
chmod 400 vpc-tutorial-key.pem
```

このコマンドは、新しいキーペアを作成し、アクセス許可が制限されたファイルにプライベートキーを保存します。

最新の Amazon Linux 2 AMI を見つける

インスタンスに使用する最新の Amazon Linux 2 AMI を見つけます。

```
aws ec2 describe-images --owners amazon \
  --filters "Name=name,Values=amzn2-ami-hvm-*-x86_64-gp2" "Name=state,Values=available"
 \
  --query "sort_by(Images, &CreationDate)[-1].ImageId" --output text
```

出力の AMI ID を書き留めます。このチュートリアルでは、例として「ami-0123456789abcdef0」を使用します。

パブリックサブネットでウェブサーバーを起動する

次に、パブリックサブネットで EC2 インスタンスを起動してウェブサーバーとして機能させます。

```
aws ec2 run-instances \
  --image-id ami-0123456789abcdef0 \
  --count 1 \
  --instance-type t2.micro \
  --key-name vpc-tutorial-key \
  --security-group-ids sg-0123456789abcdef0 \
  --subnet-id subnet-0123456789abcdef0 \
```

```
  --associate-public-ip-address \
  --user-data '#!/bin/bash
               yum update -y
               yum install -y httpd
               systemctl start httpd
               systemctl enable httpd
               echo "<h1>Hello from $(hostname -f)</h1>" > /var/www/html/index.html' \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=WebServer}]'
```

出力のインスタンス ID を書き留めます。このチュートリアルでは、ウェブサーバーインスタンスの例として「i-0123456789abcdef0」を使用します。

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

次に、プライベートサブネットで EC2 インスタンスを起動して、データベースサーバーとして機能させます。

```
aws ec2 run-instances \
  --image-id ami-0123456789abcdef0 \
  --count 1 \
  --instance-type t2.micro \
  --key-name vpc-tutorial-key \
  --security-group-ids sg-0123456789abcdef1 \
  --subnet-id subnet-0123456789abcdef2 \
  --user-data '#!/bin/bash
               yum update -y
               yum install -y mariadb-server
               systemctl start mariadb
               systemctl enable mariadb' \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=Name,Value=DBServer}]'
```

出力のインスタンス ID を書き留めます。このチュートリアルでは、データベースサーバーインスタンスの例として「i-0123456789abcdef1」を使用します。

ウェブサーバーにアクセスする

ウェブサーバーインスタンスが実行されたら、パブリック IP アドレスを使用してそのインスタンスにアクセスできます。

```
aws ec2 describe-instances \
  --instance-ids i-0123456789abcdef0 \
  --query 'Reservations[0].Instances[0].PublicIpAddress' \
```

```
  --output text
```

このコマンドは、ウェブサーバーのパブリック IP アドレスを出力します。このチュートリアルでは、例として「203.0.113.10」を使用します。

ウェブブラウザでこの URL を開くことができるようになりました: http://203.0.113.10

SSH を使用したインスタンスへの接続

ウェブサーバーに接続するには:

```
ssh -i vpc-tutorial-key.pem ec2-user@203.0.113.10
```

データベースサーバーに接続するには、まずウェブサーバーに SSH 接続し、次にデータベースサーバーに SSH 接続する必要があります。

```
# Get the private IP of the database server
aws ec2 describe-instances \
  --instance-ids i-0123456789abcdef1 \
  --query 'Reservations[0].Instances[0].PrivateIpAddress' \
  --output text
```

このコマンドは、データベースサーバーのプライベート IP アドレスを出力します。このチュートリアルでは、例として「10.0.2.10」を使用します。

```
# First SSH to web server, then to database server
ssh -i vpc-tutorial-key.pem -A ec2-user@203.0.113.10
ssh ec2-user@10.0.2.10
```

これは、作成したネットワークアーキテクチャを示しています。ウェブサーバーにはパブリックにアクセスでき、データベースサーバーには VPC 内からのみアクセスできます。

トラブルシューティング

VPC の作成時に発生する可能性がある一般的な問題とその解決方法は以下のとおりです。

CIDR ブロックの重複

CIDR ブロックの重複に関するエラーが表示された場合は、VPC とサブネットの CIDR ブロックがアカウント内の既存の VPC またはサブネットと重複していないことを確認してください。

アクセス許可エラー

アクセス許可エラーが発生した場合は、IAM ユーザーまたはロールに VPC リソースを作成および管理するために必要なアクセス許可があることを確認してください。AmazonVPCFullAccess ポリシーをアタッチするか、必要なアクセス許可を持つカスタムポリシーを作成する必要があります。

リソースの制限

AWS アカウントには、作成できる VPC、サブネット、その他のリソースの数に対するデフォルトの制限があります。これらの制限に達した場合は、AWS サポートセンターを通じて引き上げをリクエストできます。

クリーンアップ中の依存関係の障害

リソースをクリーンアップする際、間違った順序でリソースを削除しようとすると、依存関係エラーが発生する可能性があります。最も依存しているリソースから始めて、常に作成と逆の順序でリソースを削除します。

リソースをクリーンアップする

VPC の使用が終了したら、料金が発生しないようにリソースをクリーンアップできます。依存関係を正しく処理するには、作成と逆の順序でリソースを削除します。

EC2 インスタンスを終了する

```
aws ec2 terminate-instances --instance-ids i-0123456789abcdef0 i-0123456789abcdef1
aws ec2 wait instance-terminated --instance-ids i-0123456789abcdef0 i-0123456789abcdef1
```

キーペアを削除する

```
aws ec2 delete-key-pair --key-name vpc-tutorial-key
rm vpc-tutorial-key.pem
```

NAT ゲートウェイを削除する

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-0123456789abcdef0
aws ec2 wait nat-gateway-deleted --nat-gateway-ids nat-0123456789abcdef0
```

Elastic IP を解放する

```
aws ec2 release-address --allocation-id eipalloc-0123456789abcdef0
```

セキュリティグループを削除する

```
aws ec2 delete-security-group --group-id sg-0123456789abcdef1
aws ec2 delete-security-group --group-id sg-0123456789abcdef0
```

ルートテーブルを削除する

まず、ルートテーブルの関連付け ID を見つけます。

```
aws ec2 describe-route-tables --route-table-id rtb-0123456789abcdef0
aws ec2 describe-route-tables --route-table-id rtb-0123456789abcdef1
```

次に、サブネットからルートテーブルの関連付けを解除します (関連付け ID を出力の関連付け ID に置き換えます)。

```
aws ec2 disassociate-route-table --association-id rtbassoc-0123456789abcdef0
aws ec2 disassociate-route-table --association-id rtbassoc-0123456789abcdef1
aws ec2 disassociate-route-table --association-id rtbassoc-0123456789abcdef2
aws ec2 disassociate-route-table --association-id rtbassoc-0123456789abcdef3
```

次に、ルートテーブルを削除します。

```
aws ec2 delete-route-table --route-table-id rtb-0123456789abcdef1
aws ec2 delete-route-table --route-table-id rtb-0123456789abcdef0
```

インターネットゲートウェイをデタッチおよび削除する

```
aws ec2 detach-internet-gateway --internet-gateway-id igw-0123456789abcdef0 --vpc-id
 vpc-0123456789abcdef0
aws ec2 delete-internet-gateway --internet-gateway-id igw-0123456789abcdef0
```

サブネットを削除する

```
aws ec2 delete-subnet --subnet-id subnet-0123456789abcdef0
aws ec2 delete-subnet --subnet-id subnet-0123456789abcdef1
aws ec2 delete-subnet --subnet-id subnet-0123456789abcdef2
aws ec2 delete-subnet --subnet-id subnet-0123456789abcdef3
```

VPC を削除する

```
aws ec2 delete-vpc --vpc-id vpc-0123456789abcdef0
```

本番環境への移行

このチュートリアルは、AWS CLI を使用して VPC を作成する方法を学習できるように設計されています。本番環境では、次のセキュリティとアーキテクチャのベストプラクティスを検討してください。

1. セキュリティグループルール: 0.0.0.0/0 からのトラフィックを許可するのではなく、インバウンドトラフィックを特定の IP 範囲に制限します。
2. 高可用性: プライベートサブネットがある各アベイラビリティーゾーンに NAT ゲートウェイをデプロイして、単一障害点を排除します。
3. ネットワーク ACL: セキュリティグループを超えたセキュリティの追加レイヤーとしてネットワーク ACL を実装します。
4. VPC フローログ: VPC フローログを有効にして、ネットワークトラフィックパターンをモニタリングおよび分析します。
5. リソースタグ付け: リソース管理を改善するための包括的なタグ付け戦略を実装します。

本番環境に対応したアーキテクチャの構築の詳細については、「AWS Well-Architected フレームワーク」と「AWS セキュリティ、アイデンティティ、コンプライアンスに関するベストプラクティス」を参照してください。

次のステップ

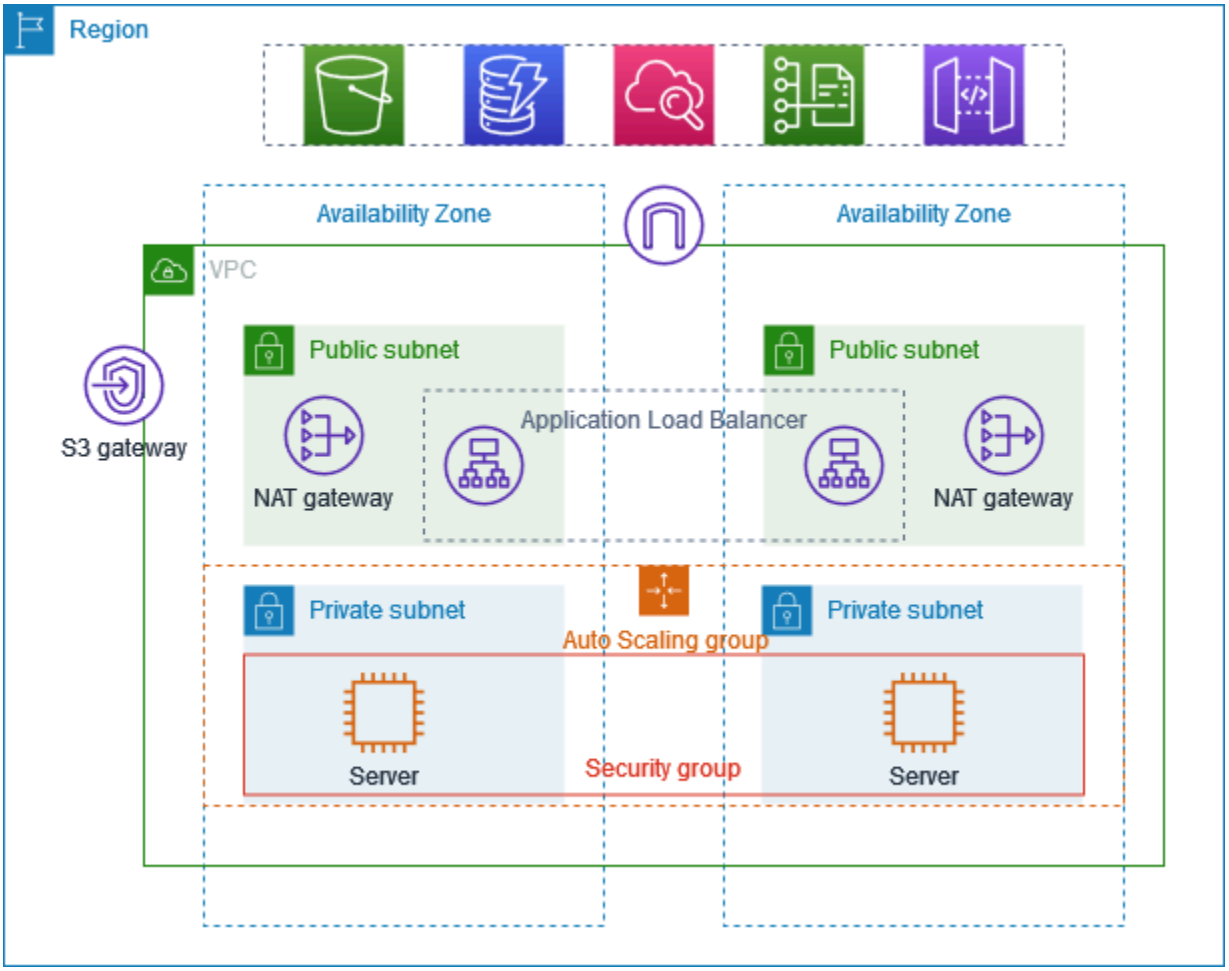
パブリックサブネットとプライベートサブネットを持つ VPC を作成したので、次のことができます。

1. パブリックサブネットまたはプライベートサブネットで EC2 インスタンスを起動する。
2. ロードバランサーをデプロイして、複数のインスタンスにトラフィックを分散する。
3. 高可用性とスケーラビリティのために Auto Scaling グループを設定する。
4. プライベートサブネットで RDS データベースを設定する。
5. VPC ピアリングを実装して、他の VPC に接続する。
6. VPN 接続を設定して、VPC をオンプレミスネットワークに接続する。

AWS CLI を使用してプライベートサブネットおよび NAT ゲートウェイを持つ VPC を作成する

このチュートリアルでは、AWS CLI を使用して本番環境のサーバーに使用できる VPC を作成する方法を説明します。回復性を高めるには、Auto Scaling グループと Application Load Balancer を使用して、2 つのアベイラビリティーゾーンにサーバーをデプロイします。セキュリティを強化するために、サーバーをプライベートサブネットにデプロイします。サーバーはロードバランサーを介してリクエストを受信し、NAT ゲートウェイを使用してインターネットに接続できます。回復性を高めるには、NAT ゲートウェイを各アベイラビリティーゾーンにデプロイします。

次の図は、このチュートリアルに含まれるリソースの概要を示しています。VPC には、2 つのアベイラビリティーゾーン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。各パブリックサブネットには、NAT ゲートウェイとロードバランサーノードが含まれています。サーバーはプライベートサブネットで実行され、Auto Scaling グループを使用して起動および終了し、ロードバランサーからトラフィックを受信します。サーバーは、NAT ゲートウェイを使用してインターネットに接続できます。サーバーは、ゲートウェイ VPC エンドポイントを使用して Amazon S3 に接続できます。



前提条件

このチュートリアルを開始する前に、以下が必要です。

- VPC リソース、EC2 インスタンス、ロードバランサー、Auto Scaling グループを作成するアクセス許可を持ってインストールおよび設定されている AWS CLI。AWS CLI のインストールについては、「AWS CLI の最新バージョンを使用してインストールまたは更新を行う」を参照してください。
- サブネット、ルートテーブル、インターネットゲートウェイなど、VPC の概念に関する基本的な知識。
- jq コマンドライン JSON プロセッサがインストールされている。これは、AWS CLI コマンドの出力を解析するために使用されます。jq のインストールの詳細については、「Download jq」を参照してください。
- 作成するリソースに十分なサービスクォータ。これには以下が含まれます。
- 少なくとも 2 つの使用可能な Elastic IP アドレス

- 少なくとも 2 つの NAT ゲートウェイ
- 少なくとも 1 つの VPC
- 少なくとも 4 つのサブネット
- 少なくとも 1 つの Application Load Balancer

推定コスト: このチュートリアルで作成されたリソースには、AWS アカウントで料金が発生します。NAT ゲートウェイ: 1 時間あたり約 $0.045、およびデータ処理料金 Elastic IP アドレス: 実行中のインスタンスに関連付けられている場合は無料、関連付けられていない場合は 1 時間あたり約 $0.005 EC2 インスタンス: インスタンスタイプによって異なる (このチュートリアルでは t3.micro を使用) Application Load Balancer: 1 時間あたり約 $0.0225、およびデータ処理料金

VPC およびサブネットを作成する

まず、CIDR ブロックが 10.0.0.0/16 の VPC を作成します。これにより、最大 65,536 個のプライベート IP アドレスが提供されます。

```
# Create a VPC with CIDR block 10.0.0.0/16
aws ec2 create-vpc --cidr-block 10.0.0.0/16 --tag-specifications
 'ResourceType=vpc,Tags=[{Key=Name,Value=ProductionVPC}]'
```

このコマンドにより、以下のような出力が返されます。

```
{
    "Vpc": {
        "CidrBlock": "10.0.0.0/16",
        "DhcpOptionsId": "dopt-abcd1234",
        "State": "pending",
        "VpcId": "vpc-abcd1234",
        "OwnerId": "123456789012",
        "InstanceTenancy": "default",
        "Ipv6CidrBlockAssociationSet": [],
        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-abcd1234",
                "CidrBlock": "10.0.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
```

```
        "IsDefault": false,
        "Tags": [
            {
                "Key": "Name",
                "Value": "ProductionVPC"
            }
        ]
    }
}
```

出力の VPC ID を書き留めます (例: vpc-abcd1234)。この ID は後続のコマンドで使用します。

次に、リージョン内の 2 つのアベイラビリティーゾーンを特定して、回復力のあるアーキテクチャを作成します。

```
# Get available Availability Zones
aws ec2 describe-availability-zones --query 'AvailabilityZones[0:2].ZoneName' --output
 text
```

このコマンドにより、以下のような出力が返されます。

```
us-east-1a us-east-1b
```

次に、ロードバランサーと NAT ゲートウェイ用の 2 つのパブリックサブネットと、アプリケーションサーバー用の 2 つのプライベートサブネットの 4 つのサブネットを作成します。vpc-abcd1234 を実際の VPC ID に置き換え、us-east-1a および us-east-1b を実際のアベイラビリティーゾーンに置き換えます。

```
# Create public subnet in first AZ
aws ec2 create-subnet \
  --vpc-id vpc-abcd1234 \
  --cidr-block 10.0.0.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=PublicSubnet1}]'

# Create private subnet in first AZ
aws ec2 create-subnet \
  --vpc-id vpc-abcd1234 \
  --cidr-block 10.0.1.0/24 \
  --availability-zone us-east-1a \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=PrivateSubnet1}]'
```

```
# Create public subnet in second AZ
aws ec2 create-subnet \
  --vpc-id vpc-abcd1234 \
  --cidr-block 10.0.2.0/24 \
  --availability-zone us-east-1b \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=PublicSubnet2}]'

# Create private subnet in second AZ
aws ec2 create-subnet \
  --vpc-id vpc-abcd1234 \
  --cidr-block 10.0.3.0/24 \
  --availability-zone us-east-1b \
  --tag-specifications 'ResourceType=subnet,Tags=[{Key=Name,Value=PrivateSubnet2}]'
```

各コマンドは、サブネット ID を含む出力を返します。後続のコマンドで使用するために、これらの ID を書き留めます。

- パブリックサブネット 1: subnet-abcd1234
- プライベートサブネット 1: subnet-abcd5678
- パブリックサブネット 2: subnet-efgh1234
- プライベートサブネット 2: subnet-efgh5678

インターネット接続を作成および設定する

このセクションでは、VPC とインターネット間の通信を許可するインターネットゲートウェイを作成し、VPC にアタッチします。

```
# Create an Internet Gateway
aws ec2 create-internet-gateway --tag-specifications 'ResourceType=internet-
gateway,Tags=[{Key=Name,Value=ProductionIGW}]'
```

コマンドは、インターネットゲートウェイ ID を含む出力を返します。この ID (例: igw-abcd1234) を書き留めます。

VPC にインターネットゲートウェイをアタッチします。igw-abcd1234 を実際のインターネットゲートウェイ ID に置き換え、vpc-abcd1234 を実際の VPC ID に置き換えます。

```
# Attach the Internet Gateway to the VPC
```

```
aws ec2 attach-internet-gateway --internet-gateway-id igw-abcd1234 --vpc-id vpc-
abcd1234
```

次に、パブリックサブネットとプライベートサブネットのルートテーブルを作成します。vpc-abcd1234 を実際の VPC ID に置き換えます。

```
# Create a route table for public subnets
aws ec2 create-route-table --vpc-id vpc-abcd1234 --tag-specifications
 'ResourceType=route-table,Tags=[{Key=Name,Value=PublicRouteTable}]'

# Create route table for private subnet in first AZ
aws ec2 create-route-table --vpc-id vpc-abcd1234 --tag-specifications
 'ResourceType=route-table,Tags=[{Key=Name,Value=PrivateRouteTable1}]'

# Create route table for private subnet in second AZ
aws ec2 create-route-table --vpc-id vpc-abcd1234 --tag-specifications
 'ResourceType=route-table,Tags=[{Key=Name,Value=PrivateRouteTable2}]'
```

各コマンドは、ルートテーブル ID を含む出力を返します。以下の ID を書き留めます。

- パブリックルートテーブル: rtb-abcd1234
- プライベートルートテーブル 1: rtb-efgh1234
- プライベートルートテーブル 2: rtb-ijkl1234

インターネットアクセスを有効にするために、パブリックルートテーブルでインターネットゲートウェイにルートを追加します。rtb-abcd1234 を実際のパブリックルートテーブル ID に置き換え、igw-abcd1234 を実際のインターネットゲートウェイ ID に置き換えます。

```
# Add a route to the Internet Gateway
aws ec2 create-route --route-table-id rtb-abcd1234 --destination-cidr-block 0.0.0.0/0
 --gateway-id igw-abcd1234
```

サブネットをそれぞれのルートテーブルに関連付けます。ルートテーブル ID とサブネット ID を実際の ID に置き換えます。

```
# Associate public subnets with the public route table
aws ec2 associate-route-table --route-table-id rtb-abcd1234 --subnet-id subnet-abcd1234
aws ec2 associate-route-table --route-table-id rtb-abcd1234 --subnet-id subnet-efgh1234

# Associate private subnets with their respective route tables
```

```
aws ec2 associate-route-table --route-table-id rtb-efgh1234 --subnet-id subnet-abcd5678
aws ec2 associate-route-table --route-table-id rtb-ijkl1234 --subnet-id subnet-efgh5678
```

NAT ゲートウェイを作成する

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がインターネットや他の AWS サービスに接続できるようにしますが、インターネットからはこれらのインスタンスとの接続を開始できないようにします。まず、NAT ゲートウェイに Elastic IP アドレスを割り当てます。

```
# Allocate Elastic IP for NAT Gateway in first AZ
aws ec2 allocate-address --domain vpc --tag-specifications 'ResourceType=elastic-
ip,Tags=[{Key=Name,Value=NAT1-EIP}]'

# Allocate Elastic IP for NAT Gateway in second AZ
aws ec2 allocate-address --domain vpc --tag-specifications 'ResourceType=elastic-
ip,Tags=[{Key=Name,Value=NAT2-EIP}]'
```

各コマンドは、割り当て ID を含む出力を返します。以下の ID を書き留めます。

- EIP 1 割り当て ID: `eipalloc-abcd1234`
- EIP 2 割り当て ID: `eipalloc-efgh1234`

各パブリックサブネットに NAT ゲートウェイを作成します。サブネット ID と割り当て ID を実際の ID に置き換えます。

```
# Create NAT Gateway in public subnet of first AZ
aws ec2 create-nat-gateway \
  --subnet-id subnet-abcd1234 \
  --allocation-id eipalloc-abcd1234 \
  --tag-specifications 'ResourceType=natgateway,Tags=[{Key=Name,Value=NAT-Gateway1}]'

# Create NAT Gateway in public subnet of second AZ
aws ec2 create-nat-gateway \
  --subnet-id subnet-efgh1234 \
  --allocation-id eipalloc-efgh1234 \
  --tag-specifications 'ResourceType=natgateway,Tags=[{Key=Name,Value=NAT-Gateway2}]'
```

各コマンドは、NAT ゲートウェイ ID を含む出力を返します。以下の ID を書き留めます。

- NAT ゲートウェイ 1: `nat-abcd1234`

- NAT ゲートウェイ 2: nat-efgh1234

NAT ゲートウェイのプロビジョニングには数分かかります。使用可能になるまで待ってから次に進みます。NAT ゲートウェイ ID を実際の ID に置き換えます。

```
# Wait for NAT Gateways to be available
aws ec2 wait nat-gateway-available --nat-gateway-ids nat-abcd1234
aws ec2 wait nat-gateway-available --nat-gateway-ids nat-efgh1234
```

プライベートルートテーブルの NAT ゲートウェイにルートを追加して、プライベートサブネットのインスタンスのインターネットアクセスを有効にします。ルートテーブル ID と NAT ゲートウェイ ID を実際の ID に置き換えます。

```
# Add route to NAT Gateway 1 in private route table 1
aws ec2 create-route \
  --route-table-id rtb-efgh1234 \
  --destination-cidr-block 0.0.0.0/0 \
  --nat-gateway-id nat-abcd1234

# Add route to NAT Gateway 2 in private route table 2
aws ec2 create-route \
  --route-table-id rtb-ijkl1234 \
  --destination-cidr-block 0.0.0.0/0 \
  --nat-gateway-id nat-efgh1234
```

Amazon S3 の VPC エンドポイントを作成する

Amazon S3 の VPC エンドポイントを使用すると、プライベートサブネットのインスタンスは NAT ゲートウェイを経由せずに S3 にアクセスできるため、データ転送コストを削減し、ネットワークパフォーマンスが向上します。vpc-abcd1234 を実際の VPC ID に置き換え、ルートテーブル ID を実際の ID に置き換えます。

```
# Get the prefix list ID for S3 in your region
S3_PREFIX_LIST_ID=$(aws ec2 describe-prefix-lists --filters "Name=prefix-
list-name,Values=com.amazonaws.$(aws configure get region).s3" --query
 'PrefixLists[0].PrefixListId' --output text)

# Create the VPC endpoint for S3
aws ec2 create-vpc-endpoint \
  --vpc-id vpc-abcd1234 \
```

```
  --service-name com.amazonaws.$(aws configure get region).s3 \
  --route-table-ids rtb-efgh1234 rtb-ijkl1234 \
  --tag-specifications 'ResourceType=vpc-endpoint,Tags=[{Key=Name,Value=S3-Endpoint}]'
```

コマンドは、VPC エンドポイント ID を含む出力を返します。この ID (例: vpce-abcd1234) を書き留めます。

セキュリティグループの設定

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制御します。任意の場所からのインバウンド HTTP トラフィックを許可するロードバランサーのセキュリティグループを作成します。vpc-abcd1234 を実際の VPC ID に置き換えます。

```
# Create security group for the load balancer
aws ec2 create-security-group \
  --group-name LoadBalancerSG \
  --description "Security group for the load balancer" \
  --vpc-id vpc-abcd1234 \
  --tag-specifications 'ResourceType=security-
group,Tags=[{Key=Name,Value=LoadBalancerSG}]'
```

コマンドは、セキュリティグループ ID を含む出力を返します。この ID (例: sg-abcd1234) を書き留めます。

ロードバランサーへのインバウンド HTTP トラフィックを許可します。sg-abcd1234 を実際のロードバランサーセキュリティグループ ID に置き換えます。

```
# Allow inbound HTTP traffic from anywhere
aws ec2 authorize-security-group-ingress \
  --group-id sg-abcd1234 \
  --protocol tcp \
  --port 80 \
  --cidr 0.0.0.0/0
```

ロードバランサーからのインバウンドトラフィックのみを許可するアプリケーションサーバーのセキュリティグループを作成します。vpc-abcd1234 を実際の VPC ID に置き換えます。

```
# Create security group for the application servers
aws ec2 create-security-group \
```

```
  --group-name AppServerSG \
  --description "Security group for the application servers" \
  --vpc-id vpc-abcd1234 \
  --tag-specifications 'ResourceType=security-
group,Tags=[{Key=Name,Value=AppServerSG}]'
```

コマンドは、セキュリティグループ ID を含む出力を返します。この ID (例: sg-efgh1234) を書き留めます。

ロードバランサーセキュリティグループからアプリケーションサーバーへのインバウンド HTTP トラフィックを許可します。sg-efgh1234 を実際のアプリケーションサーバーセキュリティグループ ID に置き換え、sg-abcd1234 を実際のロードバランサーセキュリティグループ ID に置き換えます。

```
# Allow inbound HTTP traffic from the load balancer security group
aws ec2 authorize-security-group-ingress \
  --group-id sg-efgh1234 \
  --protocol tcp \
  --port 80 \
  --source-group sg-abcd1234
```

EC2 インスタンスの起動テンプレートを作成する

起動テンプレートには、インスタンスを起動するための設定情報 (AMI ID、インスタンスタイプ、セキュリティグループなど) が含まれています。まず、インスタンスの起動時に実行されるユーザーデータスクリプトを作成します。

```
cat > user-data.sh << 'EOF'
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<h1>Hello from $(hostname -f) in $(curl -s http://169.254.169.254/latest/meta-
data/placement/availability-zone)</h1>" > /var/www/html/index.html
EOF
```

ユーザーデータスクリプトを base64 でエンコードします。

```
USER_DATA=$(base64 -w 0 user-data.sh)
```

最新の Amazon Linux 2 AMI ID を取得します。

```
# Get the latest Amazon Linux 2 AMI ID
aws ec2 describe-images --owners amazon --filters "Name=name,Values=amzn2-ami-hvm-
*-x86_64-gp2" "Name=state,Values=available" --query 'sort_by(Images, &CreationDate)
[-1].ImageId' --output text
```

AMI ID、インスタンスタイプ、セキュリティグループ、ユーザーデータを使用して起動テンプレートを作成します。sg-efgh1234 を実際のアプリケーションサーバーのセキュリティグループ ID に置き換え、$AMI_ID と $USER_DATA を前のコマンドで取得した値に置き換えます。

```
# Create a launch template
aws ec2 create-launch-template \
  --launch-template-name AppServerTemplate \
  --version-description "Initial version" \
  --tag-specifications 'ResourceType=launch-
template,Tags=[{Key=Name,Value=AppServerTemplate}]' \
  --launch-template-data '{
    "NetworkInterfaces": [{
      "DeviceIndex": 0,
      "Groups": ["sg-efgh1234"],
      "DeleteOnTermination": true
    }],
    "ImageId": "ami-abcd1234",
    "InstanceType": "t3.micro",
    "UserData":
 "IyEvYmluL2Jhc2gKeXVtIHVwZGF0ZSAteQp5dW0gaW5zdGFsbCAteSBodHRwZApzeXN0ZW1jdGwgc3RhcnQgaHR0cGQKc
+SGVsbG8gZnJvbSAkKGhvc3RuYW1lIC1mKSBpbiAkKGN1cmwgLXMgaHR0cDovLzE2OS4yNTQuMTY5LjI1NC9sYXRlc3QvbW
    "TagSpecifications": [{
      "ResourceType": "instance",
      "Tags": [{
        "Key": "Name",
        "Value": "AppServer"
      }]
    }]
  }'
```

ロードバランサーおよびターゲットグループを作成する

ターゲットグループは、指定したプロトコルとポートを使用して、EC2 インスタンスなどの登録済みターゲットにリクエストをルーティングします。アプリケーションサーバーのターゲットグループを作成します。vpc-abcd1234 を実際の VPC ID に置き換えます。

```
# Create a target group
aws elbv2 create-target-group \
  --name AppTargetGroup \
  --protocol HTTP \
  --port 80 \
  --vpc-id vpc-abcd1234 \
  --target-type instance \
  --health-check-protocol HTTP \
  --health-check-path / \
  --health-check-port traffic-port
```

コマンドは、ターゲットグループ ARN を含む出力を返します。この ARN (例: arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/AppTargetGroup/abcd1234) を書き留めます。

パブリックサブネットに Application Load Balancer を作成します。サブネット ID とセキュリティグループ ID を実際の ID に置き換えます。

```
# Create a load balancer
aws elbv2 create-load-balancer \
  --name AppLoadBalancer \
  --subnets subnet-abcd1234 subnet-efgh1234 \
  --security-groups sg-abcd1234 \
  --tags Key=Name,Value=AppLoadBalancer
```

コマンドは、ロードバランサー ARN を含む出力を返します。この ARN (例: arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/AppLoadBalancer/abcd1234) を書き留めます。

ロードバランサーがアクティブになるまで待ってから次に進みます。arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/AppLoadBalancer/abcd1234 を実際のロードバランサー ARN に置き換えます。

```
# Wait for load balancer to be active
aws elbv2 wait load-balancer-available \
  --load-balancer-arns arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/app/AppLoadBalancer/abcd1234
```

HTTP トラフィックをターゲットグループに転送するロードバランサーのリスナーを作成します。ロードバランサー ARN とターゲットグループ ARN を実際の ARN に置き換えます。

```
# Create a listener
aws elbv2 create-listener \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
app/AppLoadBalancer/abcd1234 \
  --protocol HTTP \
  --port 80 \
  --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
east-1:123456789012:targetgroup/AppTargetGroup/abcd1234
```

Auto Scaling グループを作成する

Auto Scaling グループには、自動スケーリングと管理の目的で論理グループとして扱われる EC2 インスタンスの集合が含まれます。起動テンプレートを使用し、プライベートサブネットにインスタンスを配置する Auto Scaling グループを作成します。サブネット ID とターゲットグループ ARN を実際の ID と ARN に置き換えます。

```
# Create an Auto Scaling group
aws autoscaling create-auto-scaling-group \
  --auto-scaling-group-name AppAutoScalingGroup \
  --launch-template LaunchTemplateName=AppServerTemplate,Version='$Latest' \
  --min-size 2 \
  --max-size 4 \
  --desired-capacity 2 \
  --vpc-zone-identifier "subnet-abcd5678,subnet-efgh5678" \
  --target-group-arns arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/
AppTargetGroup/abcd1234 \
  --health-check-type ELB \
  --health-check-grace-period 300 \
  --tags Key=Name,Value=AppServer,PropagateAtLaunch=true
```

設定をテストする

Auto Scaling グループがインスタンスを起動し、ヘルスチェックに合格したら、ロードバランサーをテストできます。ロードバランサーの DNS 名を取得します。ロードバランサー ARN を実際の ARN に置き換えます。

```
# Get the DNS name of the load balancer
aws elbv2 describe-load-balancers \
  --load-balancer-arns arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/app/AppLoadBalancer/abcd1234 \
  --query "LoadBalancers[0].DNSName" \
```

```
  --output text
```

curl を使用して、ロードバランサー名でアプリケーションをテストします。

```
curl http://LoadBalancerName
```

ページを複数回更新すると、異なるアベイラビリティーゾーンの異なるインスタンスからのレスポンスが表示されます。

リソースをクリーンアップする

このチュートリアルを終了したら、料金が発生しないようにすべてのリソースを削除する必要があります。すべての ID を実際のリソース ID に置き換えます。

```
# Delete the Auto Scaling group
aws autoscaling delete-auto-scaling-group \
  --auto-scaling-group-name AppAutoScalingGroup \
  --force-delete

# Wait for the Auto Scaling group to be deleted
sleep 60

# Delete the load balancer
aws elbv2 delete-load-balancer \
  --load-balancer-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/
app/AppLoadBalancer/abcd1234

# Wait for the load balancer to be deleted
sleep 30

# Delete the target group
aws elbv2 delete-target-group \
  --target-group-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:targetgroup/
AppTargetGroup/abcd1234

# Delete the launch template
aws ec2 delete-launch-template \
  --launch-template-name AppServerTemplate

# Delete the NAT Gateways
aws ec2 delete-nat-gateway --nat-gateway-id nat-abcd1234
aws ec2 delete-nat-gateway --nat-gateway-id nat-efgh1234
```

```
# Wait for the NAT Gateways to be deleted
sleep 90

# Release the Elastic IPs
aws ec2 release-address --allocation-id eipalloc-abcd1234
aws ec2 release-address --allocation-id eipalloc-efgh1234

# Delete the VPC endpoint
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-abcd1234

# Wait for security group dependencies to clear
sleep 30

# Delete the security groups
aws ec2 delete-security-group --group-id sg-efgh1234
aws ec2 delete-security-group --group-id sg-abcd1234

# Detach the Internet Gateway
aws ec2 detach-internet-gateway --internet-gateway-id igw-abcd1234 --vpc-id vpc-
abcd1234

# Delete the Internet Gateway
aws ec2 delete-internet-gateway --internet-gateway-id igw-abcd1234

# Delete the route tables
aws ec2 delete-route-table --route-table-id rtb-efgh1234
aws ec2 delete-route-table --route-table-id rtb-ijkl1234
aws ec2 delete-route-table --route-table-id rtb-abcd1234

# Delete the subnets
aws ec2 delete-subnet --subnet-id subnet-abcd1234
aws ec2 delete-subnet --subnet-id subnet-efgh1234
aws ec2 delete-subnet --subnet-id subnet-abcd5678
aws ec2 delete-subnet --subnet-id subnet-efgh5678

# Delete the VPC
aws ec2 delete-vpc --vpc-id vpc-abcd1234
```

次のステップ

プライベートサブネットと NAT ゲートウェイを持つ VPC を作成したので、次の関連トピックを確認しましょう。

- VPC のセキュリティのベストプラクティス
- VPC フローログを使用した IP トラフィックのログ記録
- Auto Scaling グループのスケーリングポリシー
- ロードバランサーターゲットグループのヘルスチェック

Amazon VPC クォータ

以下の表は、AWS アカウントに対して適用される 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。特記されていない場合、これらのクォータはリージョンごとに存在します。

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	引き上げ可能	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、リージョンあたり何百個もの VPC を使用できるようにします。
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	5	はい (最大 50)	1 つの VPC に割り当てることができる CIDR の数。

名前	デフォルト	引き上げ可能	コメント
各リージョンのアカウントごとの VPC ブロックパブリックアクセスの除外	50	はい。引き上げをリクエストするには、「AWS サポートユーザーガイド」の「サービスクォータ引き上げリクエスト」を参照してください。	アカウントで作成できる VPC BPA 除外の数。

DNS

各 EC2 インスタンスは、Route 53 Resolver (具体的には 10.0.0.2、169.254.169.253などの .2 アドレス) にネットワークインターフェイスあたり 1024 パケット/秒でパケットを送信できます。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

名前	デフォルト	引き上げ可能	コメント
リージョンあたりの 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 と共有 VPC に適用されます。
パブリック NAT ゲートウェイあたりの Elastic IP アドレスの数	2	はい	最大 8 までクォータ引き上げをリクエストできます。

ゲートウェイ

名前	デフォルト	引き上げ可能	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、リージョンあたりの 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、リージョンあたりの 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`、および `deleting` 状態のクォータのみにカウントされます。
NAT ゲートウェイあたりのプライベート IP アドレスのクォータ	8	はい	
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

カスタマーマネージドプレフィックスリストのデフォルトクォータは調整することができますが、Service Quotas コンソールを使用してクォータをリクエストすることはできません。サポート

ケースを開く必要があります。「AWS サポートユーザーガイド」の「サービスクォータ引き上げリクエスト」を参照してください。

名前	デフォルト	引き上げ可能	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	はい	
プレフィックスリストあたりのバージョン数	1,000	はい	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。
プレフィクスリストあたりの最大エントリ数	1,000	はい	カスタマーマネージドプレフィックスリストのサイズは、最大 1,000 個まで変更できます。詳細については、「プレフィックスリストのサイズ変更」を参照してください。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、最大エントリ数が 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	10,000	はい	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 10,000 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 10,000 個作成することができます。プレフィッ

名前	デフォルト	引き上げ可能	コメント
			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トへの他のアカウントの参照はお客様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	引き上げ可能	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	このクォータは、インバウンドルールの最大数とアウトバウンドルールの最大数の両方を決定します。このクォータはインバウンドルールを 40 まで、アウトバウンドルールを 40 まで (合計 80 ルールまで) 増やせますが、ネットワークのパフォーマンスに影響が出る可能性があります。

ネットワークインターフェイス

名前	デフォルト	引き上げ可能	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。

名前	デフォルト	引き上げ可能	コメント
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 と共有 VPC に適用されます。この制限はアベイラビリティーゾーン (AZ) ごとに適用されます。例えば、ネットワークインターフェイスが 3 つの AZ にある場合、各 AZ の制限は 5,000 で、リージョンの制限は 15,000 です。

ルートテーブル

名前	デフォルト	引き上げ可能	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。ルートテーブルのクォータの増加をリクエストする場合、サブネットのクォータの増加もリクエストできます。ルートテーブルには複数のサブネットを関連付けることができるものの、1 つのサブネットは 1 つのルートテーブルにしか関連付けることができません。
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)	500	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

名前	デフォルト	引き上げ可能	コメント
			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伝播されるルートの数	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。

ルートサーバー数

名前	デフォルト	引き上げ可能	コメント
VPC あたりのルートサーバー数	5	はい。引き上げをリクエストするには、「AWS サポートユーザーガイド」の「サービスクォータ引き上げリクエスト」を参照してください。	

名前	デフォルト	引き上げ可能	コメント
ルートサーバーあたりのルートサーバーエンドポイント数	10	はい。引き上げをリクエストするには、「AWS サポートユーザーガイド」の「サービスクォータ引き上げリクエスト」を参照してください。	

名前	デフォルト	引き上げ可能	コメント
ネットワークインターフェイスあたりのピアリングセッション数	20	はい。引き上げをリクエストするには、「AWS サポートユーザーガイド」の「サービスクォータ引き上げリクエスト」を参照してください。	
ルートサーバーとサブネットあたりのルートサーバーエンドポイント数	2	いいえ	冗長性を確保するためには、同じルートサーバーに対して同じサブネットに 2 つのエンドポイントしか持つことはできません。
ルートサーバーピアあたりのルートの数	100	いいえ	これは、ルートサーバーピアによって動的にアドバタイズできるルートの数です
ルートサーバーあたりのルートの数	100	いいえ	これは、ルートサーバーの Forwarding Information Base (FIB) にインストールできるルートの数です。

セキュリティグループ

名前	デフォルト	引き上げ可能	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 と共有 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	このクォータは、インバウンドルールとアウトバウンドルールに個別に適用されます。デフォルトクォータが 60 のアカウントの場合、セキュリティグループは 60 のインバウンドルール、 60 のアウトバウンドルールを使用できます。さらに、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。デフォルトのクォータが 60 のアカウントの場合、セキュリティグループは IPv4 トラフィックに 60 のインバウンドルール、IPv6 トラフィックに 60 のインバウンドルールを使用できます。詳細については、「the section called “セキュリティグループのサイズ”」を参照してください。 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

名前	デフォルト	引き上げ可能	コメント
			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。
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)	このクォータに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。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デフォルト値の 5 より下の値に減らすには、「AWS サポートユーザーガイド」の「サービスクォータ引き上げリクエスト」を参照してください。引き下げをリクエストするには、引き上げをリクエストする場合と同じプロセスを使用します。

VPC サブネット共有

すべての標準 VPC のクォータは共有 VPC サブネットに適用されます。

名前	デフォルト	引き上げ可能	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最大数。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。 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。

名前	デフォルト	引き上げ可能	コメント
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。

ネットワークアドレスの使用状況

ネットワークアドレスの使用状況 (NAU) は、マネージドプレフィックスリスト内の IP アドレス、ネットワークインターフェイス、CIDR で構成されます。NAU は、VPC 内のリソースに適用されるメトリクスで、VPC のサイズを計画および監視するのに役立ちます。詳細については、「ネットワークアドレスの使用状況」を参照してください。

NAU 数を構成するリソースには、それぞれ個別の Service Quotas があります。VPC に利用可能な NAU キャパシティがあっても、リソースが Service Quotas を超えていると VPC にリソースを起動することはできません。

名前	デフォルト	引き上げ可能	コメント
ネットワークアドレスの使用状況	64,000	はい (最大 256,000 まで)	VPC あたりの NAU ユニットの最大数。
ピアリングされたネットワークアドレスの使用状況	128,000	はい (最大 512,000 まで)	VPC と、リージョン内でそれにピアリングされているすべての VPC に対する NAU ユニットの最大数。異なるリージョンにまたがってピアリングされている VPC は、この最大数にカウントされません。

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 デベロッパーガイドの「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

詳細については次を参照してください:

- 「AWS Client VPN 管理者ガイド」の「AWS Client VPN クォータ」
- 「AWS Direct Connect ユーザーガイド」の「Direct Connect クォータ」
- Amazon VPC ピアリングガイドの「ピアリング接続のクォータ」
- AWS PrivateLink ガイドの「PrivateLink クォータ」
- 「AWS Site-to-Site VPN ユーザーガイド 」の「Site-to-Site VPN クォータ」
- Amazon VPC トラフィックミラーリングガイドの「トラフィックミラーリングのクォータ」
- Amazon VPC Transit Gateways ガイドの「トランジットゲートウェイのクォータ」

ドキュメント履歴

次の表には、「Amazon VPC ユーザーガイド」の各リリースで行われた重要な変更の説明が記載されています。

変更	説明	日付
VPC 暗号化コントロール	VPC 暗号化コントロールを使用して、VPC 内のネットワークトラフィックに対して転送時の暗号化を強制できるようになりました。この機能では、一元化された暗号化ポリシーの強制機能およびモニタリング機能を提供します。	2025 年 11 月 21 日
自動マルチ AZ 拡張用のリージョン NAT ゲートウェイ	ワークロードフットプリントに基づいて、アベイラビリティーゾーン間で自動的に拡張を行うリージョン NAT ゲートウェイを使用できるようになりました。リージョン NAT ゲートウェイにより、手動による介入なしで、セットアップの簡素化、セキュリティの強化、自動高可用性が実現します。	2025 年 11 月 19 日
パブリック IP アドレスへのインバウンド VPC トラフィックのルーティング	インターネットからのインバウンドトラフィックを VPC 内の特定のパブリック IP アドレスに転送する高度なルーティングルールを設定できるようになりました。この設定により、イングレスシナリオのためのトラフィックフロー	2025 年 8 月 13 日

	とルーティングの決定をより細かく制御できます。	
Amazon VPC Route Server を使用する VPC 内の動的ルーティング	Amazon VPC Route Server は、VPC とそのインターネットゲートウェイ内にデプロイされたワークロード間のトラフィックのルーティングを簡略化します。この機能を使用すると、VPC Route Server は任意の IPv4 または IPv6 ルートで VPC およびゲートウェイルートテーブルを動的に更新し、それらのワークロードのルーティングの耐障害性を実現します。これにより、VPC 内のトラフィックを自動的に再ルーティングできるため、VPC ルーティングの管理の容易性とサードパーティーワークロードとの相互運用性が向上します。	2025 年 3 月 31 日
AWS 管理ポリシーの更新	Amazon VPC で、AmazonVPCFullAccess および AmazonVPCReadOnlyAccess の管理ポリシーが更新されました。	2024 年 12 月 9 日
VPC BPA の宣言型ポリシーのサポート	AWS Organizations を使用して組織内のアカウントを管理している場合は、宣言型ポリシーを使用して、組織内のアカウントに VPC BPA を適用できます。	2024 年 12 月 1 日

VPC ブロックパブリックアクセス (BPA)	VPC ブロックパブリックアクセス (BPA) を使用すると、リージョンで所有している VPC とサブネットのリソースが、インターネットゲートウェイおよびエグレスのみのインターネットゲートウェイを通じてインターネットに到達したり、インターネットから到達されたりすることをブロックできます。	2024 年 11 月 19 日
共有セキュリティグループ	この機能を使用すると、1 つのセキュリティグループを、他の AWS Organizations アカウントと共有できます。	2024 年 10 月 30 日
セキュリティグループの VPC の関連付け	この機能を使用すると、1 つのセキュリティグループを、同じリージョン内の複数の VPC に関連付けることができます。	2024 年 10 月 30 日
NAT ゲートウェイ MTU のサポート	NAT ゲートウェイは、最大送信単位 (MTU) が 8,500 のトラフィックをサポートします。	2024 年 9 月 10 日
プライベート IPv6 アドレス指定	プライベート IPv6 アドレス指定に関する情報が追加されました。プライベート IPv6 アドレスは、Amazon VPC IP Address Manager でのみ使用できます。	2024 年 8 月 8 日

IPv6 優先リースタイム	IPv6 が割り当てられた実行中のインスタンスが DHCPv6 リースを更新する頻度を選択できるようになりました。	2024 年 2 月 20 日
ガイド構造の点検と改善	ガイドの構造を点検し、特定のシナリオに関する情報を見つける際の、カスタマーエクスペリエンスを改善しました。	2024 年 2 月 20 日
AWS 管理ポリシーの更新	Amazon VPC で、AmazonVPCFullAccess および AmazonVPCReadOnlyAccess の管理ポリシーが更新されました。	2024 年 2 月 8 日
AWS 管理ポリシーの更新	Amazon VPC で、AmazonVPCCrossAccountNetworkInterfaceOperations 管理ポリシーが更新されました。	2023 年 9 月 25 日
EC2-Classic の廃止	EC2-Classic では、他のお客様と共有される単一のフラットネットワーク内で EC2 インスタンスが実行されていました。EC2-Classic は Amazon VPC に置き換えられます。Amazon VPC では、インスタンスは、AWS アカウント から論理的に独立した仮想プライベートクラウド (VPC) で実行されます。	2023 年 7 月 31 日

NAT ゲートウェイへのセカンダリ IPv4 アドレスの追加	パブリック NAT ゲートウェイとプライベート NAT ゲートウェイにセカンダリプライベート IPv4 アドレスを追加できます。セカンダリ IPv4 アドレスは利用可能なポート数を増やすことから、ワークロードが NAT ゲートウェイを使用して確立できる同時接続数の上限も増加します。	2023 年 1 月 31 日
IAM ベストプラクティスとの連携	IAM ベストプラクティスに沿ってガイドを更新しました。詳細については、「IAM のセキュリティのベストプラクティス」を参照してください。	2023 年 1 月 4 日
NAT ゲートウェイのプライベート IP アドレスの選択	NAT ゲートウェイを作成するときに、NAT ゲートウェイに割り当てられているプライベート IP アドレスを選択できるようになりました。これまで、プライベート IP アドレスはサブネットの IP アドレス範囲から自動的に割り当てられていました。	2022 年 11 月 17 日
IPv6 デフォルトゲートウェイルーター設定	デフォルト VPC ルーターで使用するために、3 つの IPv6 アドレスが予約されるようになりました。	2022 年 11 月 11 日
Elastic IP アドレスの移転	Elastic IP アドレスを 1 つの AWS アカウントから別のアカウントに移転できるようになりました。	2022 年 10 月 31 日

ネットワークアドレス使用状況メトリクス	VPC のネットワークアドレス使用状況メトリクスを有効にすると、VPC のサイズの計画と監視に役立ちます。	2022 年 10 月 4 日
Amazon Data Firehose へのフローログの発行	フローログデータの送信先として Amazon Data Firehose 配信ストリームを指定できます。	2022 年 9 月 8 日
NAT ゲートウェイの帯域幅	NAT ゲートウェイは最大 100 Gbps (45 Gbps から増加) の帯域幅をサポートし、1 秒あたり最大 1,000 万パケット (最低 400 万パケット) を処理できるようになりました。	2022 年 6 月 15 日
複数の IPv6 CIDR ブロック	VPC には最大 5 つまで IPv6 CIDR ブロックを関連付けることができます。	2022 年 5 月 12 日
再編	Amazon Virtual Private Cloud ユーザーガイドを全般的に再編しました。	2022 年 1 月 2 日
NAT ゲートウェイ IPv6 から IPv4 へ	NAT ゲートウェイは、IPv6 から IPv4 へのネットワークアドレス変換をサポートします (一般的に NAT64 と言う)。	2021 年 11 月 24 日
VPC 内の IPv6 専用サブネット	IPv6 専用 EC2 インスタンスを起動できる IPv6 専用サブネットを作成できます。	2021 年 11 月 23 日

Amazon S3 への VPC フローログ配信オプション	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 Global View	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	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に対するリソース ID とタグ付けのサポート	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	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
作成時のタグ付け	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
フローログの機能強化	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
フローログに対するタグ付けサポート	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付け	NAT ゲートウェイの作成時にタグを追加できます。	2020 年 3 月 9 日
フローログの最大集計間隔	フローがキャプチャされ、フローログレコードに集計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 VPC コンソールから 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

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
フローログの機能強化	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
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
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
NAT ゲートウェイに対するタグ付けサポート	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
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 アドレス範囲に対する DNS 解決サポート	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
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

ClassicLink	ClassicLink は、EC2-Classic インスタンスをアカウント内の VPC にリンクするために使用できます。VPC 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るため、EC2-Classic インスタンスと VPC 内のインスタンス間のプライベート IP アドレスを使用した通信が可能になります。	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するカスタム DNS ドメイン名を使用して、VPC 内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 アドレス指定属性の変更	サブネットのパブリック IP アドレス指定属性を変更して、そのサブネットで起動されたインスタンスがパブリック IP アドレスを取得するかどうかを指定できます。	2014 年 6 月 21 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	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
ハードウェア専有インスタンス	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