Network Load Balancer

Elastic Load Balancing

Elastic Load Balancing: Network Load Balancer

Table of Contents

Network Load Balancer とは?

Elastic Load Balancing は、受信したトラフィックを複数のアベイラビリティーゾーンの複数のターゲット (EC2 インスタンス、コンテナ、IP アドレスなど) に自動的に分散させます。登録されているターゲットの状態をモニタリングし、正常なターゲットにのみトラフィックをルーティングします。Elastic Load Balancing は、受信トラフィックの時間的な変化に応じて、ロードバランサーをスケーリングします。また、大半のワークロードに合わせて自動的にスケールできます。

Elastic Load Balancing は、Application Load Balancer、Network Load Balancer、Gateway Load Balancer、Classic Load Balancer といったロードバランサーをサポートします。ニーズに最適なタイプのロードバランサーを選択できます。このガイドでは、Network Load Balancer について説明します。その他のロードバランサーの詳細については、Application Load Balancer のユーザーガイド、Gateway Load Balancers のユーザーガイド、および Classic Load Balancer のユーザーガイドを参照してください。

Network Load Balancer のコンポーネント

ロードバランサーは、クライアントにとって単一の通信先として機能します。ロードバランサーは、受信トラフィックを Amazon EC2 インスタンスなどの複数のターゲットに分散します。これにより、アプリケーションの可用性が向上します。ロードバランサーに 1 つ以上のリスナーを追加できます。

リスナーは、構成したプロトコルとポートを使用してクライアントからの接続リクエストをチェックし、リクエストをターゲットグループに転送します。

各ターゲットグループは、指定されたプロトコルとポート番号を使用して、1 つ以上の登録済みのターゲットにリクエストをルーティングします。Network Load Balancer ターゲットグループは、TCP、UDP、TCP_UDP、TLS、QUIC、TCP_QUIC プロトコルをサポートします。1 つのターゲットを複数のターゲットグループに登録できます。ターゲットグループ単位でヘルスチェックを設定できます。ヘルスチェックは、ロードバランサーのデフォルトアクションで指定されたターゲットグループに登録済みのすべてのターゲットで実行されます。

詳細については、次のユメントを参照してください。

- ロードバランサー
- リスナー
- ターゲットグループ

Network Load Balancer の概要

Network Load Balancer は、開放型システム間相互接続 (OSI) モデルの第 4 層で機能します。1 秒あたり数百万のリクエストを処理できます。ロードバランサーは、クライアントからリクエストを受信すると、デフォルトアクションのターゲットグループからターゲットを選択します。指定したプロトコルとポートを使用して、選択したターゲットへのリクエストの送信を試みます。

ロードバランサー用のアベイラビリティーゾーンを有効にすると、Elastic Load Balancing はアベイラビリティーゾーンにロードバランサーノードを作成します。デフォルトでは、各ロードバランサーノードは、アベイラビリティーゾーン内の登録済みターゲット間でのみトラフィックを分散します。クロスゾーン負荷分散を有効にすると、各ロードバランサーノードは、有効なすべてのアベイラビリティーゾーンの登録済みターゲットにトラフィックを分散します。詳細については、「Network Load Balancer のアベイラビリティーゾーンを更新する」を参照してください。

アプリケーションの耐障害性を向上させる目的で、複数のアベイラビリティーゾーンをロードバランサーに対して有効にすることができます。各ターゲットグループで、有効にした各アベイラビリティーゾーンに 1 つ以上のターゲットがあることを確認してください。たとえば、1 つ以上のターゲットグループで 1 つのアベイラビリティーゾーン内に正常なターゲットがない場合、DNS から該当するサブネットの IP アドレスを削除しますが、他のアベイラビリティーゾーンのロードバランサーノードは、引き続きトラフィックをルーティングできます。クライアントが有効期限 (TTL) を守らず、DNS から削除された後でリクエストを IP アドレスに送信すると、そのリクエストは失敗します。

TCP トラフィックの場合、ロードバランサーは、プロトコル、送信元 IP アドレス、送信元ポート、宛先 IP アドレス、宛先ポート、および TCP シーケンス番号に基づいて、フローハッシュアルゴリズムを使用してターゲットを選択します。クライアントからの TCP 接続のソースポートとシーケンス番号は異なり、別のターゲットにルーティングできます。各 TCP 接続は、接続中は単一のターゲットにルーティングされます。

UDP トラフィックの場合、ロードバランサーは、プロトコル、送信元 IP アドレス、送信元ポート、宛先 IP アドレス、および宛先ポートに基づいて、フローハッシュアルゴリズムを使用してターゲットを選択します。UDP フローは送信元と宛先が同じであるため、その存続期間を通じて一貫して単一のターゲットにルーティングされます。異なる UDP フローは異なる送信元 IP アドレスとポートを持つため、それらは異なるターゲットにルーティングできます。

QUIC トラフィックの場合、ロードバランサーは接続 ID (CID) で指定されたサーバー ID を使用してターゲットを選択します。サーバー ID がない初回の接続試行では、プロトコル、送信元 IP アドレス、送信元ポート、送信先 IP アドレス、送信先ポートに基づくフローハッシュアルゴリズムが使用

されます。この CID の接続 ID が確立されると、CID の存続期間中、同じターゲットにルーティングされます。

Elastic Load Balancing は、有効にした各アベイラビリティーゾーンにネットワークインターフェイスを作成します。アベイラビリティーゾーンの各ロードバランサーノードは、このネットワークインターフェイスを使用して静的 IP アドレスを取得します。インターネット向けのロードバランサーを作成する場合は、必要に応じて 1 つの Elastic IP アドレスをサブネットごとに関連付けることができます。

ターゲットグループを作成するときは、そのターゲットの種類を指定します。ターゲットの種類は、ターゲットの登録方法を決定します。例えば、インスタンス ID、IP アドレス、または Application Load Balancer を登録できます。ターゲットタイプは、クライアント IP アドレスを保持するかどうかにも影響します。詳細については、「the section called “クライアント IP の保存”」を参照してください。

アプリケーションへのリクエストの流れを中断することなく、ニーズの変化に応じてロードバランサーに対してターゲットの追加と削除を行うことができます。Elastic Load Balancing はアプリケーションへのトラフィックが時間の経過とともに変化するのに応じてロードバランサーをスケーリングします。Elastic Load Balancing では、大半のワークロードに合わせた自動的なスケーリングが可能です。

登録済みのインスタンスのヘルス状態をモニタリングするために使用されるヘルスチェックを設定することで、ロードバランサーは正常なターゲットにのみリクエストを送信できます。

詳細については、Elastic Load Balancing ユーザーガイドの How Elastic Load Balancing works を参照してください。

Classic Load Balancer からの移行のメリット

Classic Load Balancer の代わりに Network Load Balancer を使用すると、次の利点があります。

- 揮発性のワークロードを処理し、毎秒数百万のリクエストに対応できる能力。
- ロードバランサーの静的 IP アドレスのサポート。ロードバランサーで有効になっているサブネットごとに 1 つの Elastic IP アドレスを割り当てることもできます。
- ロードバランサーの VPC 外のターゲットを含め、IP アドレスによるターゲットの登録をサポート。
- 1 つの EC2 インスタンス上での複数のアプリケーションへのルーティングリクエストのサポート。複数のポートを使用して、各インスタンスまたは IP アドレスを同じターゲットグループに登録できます。

- コンテナ化されたアプリケーションのサポート。Amazon Elastic Container Service (Amazon ECS) は、タスクをスケジュールするときに未使用のポートを選択し、そのポートを使用するターゲットグループにタスクを登録できます。これにより、クラスターを効率的に使用することができます。
- 各サービスの個別のヘルスステータスのモニタリングのサポート。ヘルスチェックがターゲットグループレベルで定義され、多数の Amazon CloudWatch メトリクスがターゲットグループレベルで報告されます。ターゲットグループを Auto Scaling グループにアタッチすることで、各サービスをオンデマンドで動的にスケールすることができます。
- 高度な輻輳制御を使用した QUIC および TCP_QUIC プロトコルのサポート、ラウンドトリップ接続確立の削減、TLS の組み込み、ネットワーク間の接続移行。

各ロードバランサータイプでサポートされている機能の詳細については、Elastic Load Balancing の製品比較を参照してください。

開始方法

または を使用して Network Load Balancer を作成するには AWS マネジメントコンソール AWS CLI、 AWS CloudFormation「」を参照してくださいNetwork Load Balancer を作成する。

一般的なロードバランサー設定のデモについては、Elastic Load Balancing のデモを参照してください。

料金

詳細については、Elastic Load Balancing の料金表を参照してください。

Network Load Balancer

Network Load Balancer は、クライアントにとって単一の通信先として機能します。クライアントは Network Load Balancer にリクエストを送信し、Network Load Balancer は 1 つ以上のアベイラビリティーゾーンにあるターゲット (EC2 インスタンスなど) にそれらのリクエストを送信します。

Network Load Balancer を設定するには、ターゲットグループを作成し、ターゲットグループにターゲットを登録します。有効な各アベイラビリティーゾーンに少なくとも 1 つの登録済みターゲットがあるようにする場合、Network Load Balancer が最も効果的です。さらに、リスナーを作成してクライアントからの接続リクエストがないかチェックし、リクエストをクライアントからターゲットグループ内のターゲットにルーティングします。

Network Load Balancer は、VPC ピアリング、 AWS マネージド VPN Direct Connect、およびサードパーティー VPN ソリューションを介したクライアントからの接続をサポートします。

内容

- Network Load Balancer のゾーンシフト
- Network Load Balancer のキャパシティ予約

ロードバランサーの状態

Network Load Balancer の状態は次のいずれかです。

`provisioning`

Network Load Balancer はセットアップ中です。

`active`

Network Load Balancer は完全にセットアップされており、トラフィックをルーティングする準備ができています。

`failed`

Network Load Balancer をセットアップできませんでした。

IP アドレスタイプ

クライアントが Network Load Balancer で使用できる IP アドレスのタイプを設定できます。

Network Load Balancer は次の IP アドレスタイプをサポートしています。

`ipv4`

クライアントは IPv4 アドレス (192.0.2.1 など) を使用して接続する必要があります。

`dualstack`

クライアントは、IPv4 アドレス (192.0.2.1 など) と IPv6 アドレス (例えば、2001:0db8:85a3:0:0:8a2e:0370:7334) の両方を使用して Network Load Balancer に接続できます。

考慮事項

- Network Load Balancer は、ターゲットグループの IP アドレスのタイプに基づいてターゲットと通信します。

- UDP IPv6 リスナーの送信元 IP 保存をサポートするには、[IPv6 ソース NAT の プレフィックスを有効化] がオンになっていることを確認します。
- Network Load Balancer のデュアルスタックモードを有効にすると、Elastic Load Balancing が Network Load Balancer の AAAA DNS レコードを提供します。IPv4 アドレスを使用して Network Load Balancer と通信するクライアントは、A DNS レコードを解決します。IPv6 アドレスを使用して Network Load Balancer と通信するクライアントは、AAAA DNS レコードを解決します。
- インターネットゲートウェイを経由する内部デュアルスタック Network Load Balancer へのアクセスがブロックされ、意図しないインターネットアクセスを防止します。ただし、これにより他のインターネットアクセス (ピアリング、Transit Gateway AWS Direct Connect、 など) が妨げられることはありません Site-to-Site VPN。

詳細については、「Network Load Balancer の IP アドレスタイプを更新する」を参照してください。

接続のアイドルタイムアウト

クライアントが Network Load Balancer を通じて行う TCP リクエストごとに、その接続の状態が追跡されます。アイドルタイムアウトよりも長い時間、クライアントからもターゲットからもその接続経由でデータが送信されない場合、接続は追跡されなくなります。アイドルタイムアウト期間の経過後にクライアントまたはターゲットがデータを送信した場合、クライアントは接続が無効になったことを示す TCP RST パケットを受信します。

TCP フローのデフォルトのアイドルタイムアウト値は 350 秒ですが、60～6,000 秒の任意の値に更新できます。クライアントまたはターゲットは TCP キープアライブパケットを使用して、アイドルタイムアウトを再開できます。TLS 接続を維持するために送信されるキープアライブパケットには、データまたはペイロードを含めることはできません。

TLS リスナーの接続アイドルタイムアウトは 350 秒であり、変更できません。TLS リスナーがクライアントまたはターゲットのいずれかから TCP キープアライブパケットを受信すると、ロードバランサーは TCP キープアライブパケットを生成し、20 秒ごとにフロントエンド接続とバックエンド接続の両方に送信します。この動作を変更することはできません。

UDP はコネクションレスですが、ロードバランサーは送信元と宛先のIPアドレスとポートに基づいて UDP フロー状態を維持します。これにより、同じフローに属するパケットが一貫して同じターゲットに一貫して同じターゲットに送信されます。アイドルタイムアウト期間が経過した後、ロードバランサーは着信 UDP パケットを新しいフローとみなし、それを新しいターゲットにルーティング

します。Elastic Load Balancing は、UDP フローのアイドルタイムアウト値を 120 秒に設定します。これは変更できません。

EC2 インスタンスは、リターンパスを確立するために、30 秒以内に新しいリクエストに応答する必要があります。

詳細については、「アイドルタイムアウトを更新する」を参照してください。

ロードバランサーの属性

Network Load Balancer は、属性を編集することで設定できます。詳細については、「ロードバランサー属性を編集する」を参照してください。

Network Load Balancer のロードバランサー属性を以下に示します。

`access_logs.s3.enabled`

Amazon S3 に保存されたアクセスログが有効かどうかを示します。デフォルトは `false` です。

`access_logs.s3.bucket`

アクセスログの Amazon S3 バケットの名前。この属性は、アクセスログが有効になっている場合は必須です。詳細については、「バケットの要件」を参照してください。

`access_logs.s3.prefix`

Amazon S3 バケットの場所のプレフィックス。

`deletion_protection.enabled`

削除保護が有効化されているかどうかを示します。デフォルトは `false` です。

`ipv6.deny_all_igw_traffic`

Network Load Balancer へのインターネットゲートウェイ (IGW) アクセスをブロックし、インターネットゲートウェイを経由した内部 Network Load Balancer への意図しないアクセスを防止します。インターネット向け Network Load Balancer では `false`、内部 Network Load Balancer では `true` に設定されます。この属性は、IGW 以外のインターネットアクセス (ピアリング、Transit Gateway、 AWS Direct Connect、 など) を妨げません Site-to-Site VPN。

`load_balancing.cross_zone.enabled`

クロスゾーン負荷分散が有効かどうかを示します。デフォルトは `false` です。

`dns_record.client_routing_policy`

Network Load Balancer のアベイラビリティーゾーン間でトラフィックがどのように分散されるかを示します。指定できる値は、ゾーンアフィニティが 100% の `availability_zone_affinity`、ゾーンアフィニティが 85% の `partial_availability_zone_affinity`、ゾーンアフィニティが 0% の `any_availability_zone` です。

`secondary_ips.auto_assigned.per_subnet`

設定する[セカンダリ IP アドレス](#)の数。ターゲットを追加できない場合は、これを使用してポート割り当てエラーを解決します。有効な範囲は 0～7 です。デフォルトは 0 です。この値を設定した後で、減らすことはできません。

`zonal_shift.config.enabled`

[ゾーンシフト](#)が有効になっているかどうかを示します。デフォルトは `false` です。

クロスゾーンロードバランサー

デフォルトでは、各 Network Load Balancer ノードは、アベイラビリティーゾーン内の登録済みターゲット間でのみトラフィックを分散します。クロスゾーン負荷分散をオンにすると、各 Network Load Balancer ノードは、有効なすべてのアベイラビリティーゾーンの登録済みターゲットにトラフィックを分散します。ターゲットグループレベルでクロスゾーンロードバランサーを有効にすることもできます。詳細については、「Elastic Load Balancing ユーザーガイド」の「[the section called “クロスゾーンロードバランサー”](#)」および「[クロスゾーンロードバランシング](#)」を参照してください。

DNS 名

各 Network Load Balancer は、*name*-*id*.elb.*region*.amazonaws.com の構文でデフォルトのドメインネームシステム (DNS) 名を受け取ります。例えば、my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com です。

覚えやすい DNS 名を使用する場合は、カスタムドメイン名を作成し、Network Load Balancer の DNS 名に関連付けることができます。このカスタムドメイン名を使用してクライアントがリクエストを生成すると、DNS サーバーが Network Load Balancer の DNS 名に解決します。

最初に、認定ドメイン名レジストラにドメイン名を登録します。次に、ドメインレジストラなどの DNS サービスを使用して、Network Load Balancer にリクエストをルーティングするための DNS

レコードを作成します。詳細については、DNS サービスのドキュメントを参照してください。例えば、DNS サービスとして Amazon Route 53 を使用する場合は、Network Load Balancer をポイントするエイリアスレコードを作成します。詳細については、Amazon Route 53 デベロッパーガイドの[ELB ロードバランサーへのトラフィックのルーティング](#)を参照してください。

Network Load Balancer には、有効なアベイラビリティーゾーンごとに 1 つの IP アドレスがあります。これらは Network Load Balancer ノードの IP アドレスです。Network Load Balancer の DNS 名はこれらのアドレスに解決されます。例えば、Network Load Balancer のカスタムドメイン名が `example.networkloadbalancer.com` であるとします。以下の dig または nslookup コマンドを使用して、Network Load Balancer ノードの IP アドレスを調べます。

Linux または Mac

```
$ dig +short example.networkloadbalancer.com
```

Windows

```
C:\> nslookup example.networkloadbalancer.com
```

Network Load Balancer には、Network Load Balancer ノードの DNS レコードがあります。次の構文で DNS 名を使用して、Network Load Balancer ノードの IP アドレスを調べることができます: *az*.*name*-*id*.elb.*region*.amazonaws.com。

Linux または Mac

```
$ dig +short us-east-2b.my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com
```

Windows

```
C:\> nslookup us-east-2b.my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com
```

ロードバランサーのゾーンヘルス

Network Load Balancer には、Route 53 に有効な各アベイラビリティーゾーンのゾーン DNS レコードと IP アドレスがあります。Network Load Balancer が特定のアベイラビリティーゾーンのゾーンヘルスチェックに合格しなかった場合、その DNS レコードは Route 53 から削除されます。ロードバランサーのゾーンヘルスは Amazon CloudWatch メトリクス `ZonalHealthStatus` を使用してモ

ニタリングされるため、フェイルアウェイの原因となるイベントに関する詳細なインサイトが得られ、アプリケーションの可用性を最適化するための予防策を講じることができます。詳細については、「Network Load Balancer メトリクス」を参照してください。

Network Load Balancer は、さまざまな理由でゾーンヘルスチェックに合格せず、異常になる可能性があります。ゾーンヘルスチェックに合格しなかったことによって引き起こされる異常な Network Load Balancer の原因として一般的なものを、以下に示します。

以下の原因が考えられますので、確認してください。

- ロードバランサーに正常なターゲットがない
- 正常なターゲットの数が、設定された最小値未満である
- ゾーンシフトまたはゾーン自動シフトが進行中
- 問題が検出されたため、トラフィックが自動的に正常なゾーンに移行中

Network Load Balancer を作成する

Network Load Balancer はクライアントからリクエストを受け取り、EC2 インスタンスなどのターゲットグループのターゲット間でリクエストを割り当てます。詳細については、「the section called “Network Load Balancer の概要”」を参照してください。

タスク

- 前提条件
- ロードバランサーを作成する
- ロードバランサーをテストするには
- 次の手順

前提条件

- アプリケーションがサポートするアベイラビリティーゾーンと IP アドレスタイプを決定します。これらの各アベイラビリティーゾーンのサブネットを使用してロードバランサー VPC を設定します。アプリケーションが IPv4 と IPv6 の両方のトラフィックをサポートする場合は、サブネットに IPv4 と IPv6 の両方の CIDR があることを確認します。各アベイラビリティーゾーンに少なくとも 1 つのターゲットをデプロイします。
- ターゲットインスタンスのセキュリティグループが、クライアント IP アドレス (ターゲットがインスタンス ID で指定されている場合) またはロードバランサーノード (ターゲットが IP アドレス

で指定されている場合) からのトラフィックをリスナーポートで許可している必要があります。詳細については、「the section called “ターゲットセキュリティグループ”」を参照してください。

- ターゲットインスタンスのセキュリティグループで、ヘルスチェックプロトコルを使用してヘルスチェックポートでロードバランサーからのトラフィックを許可している必要があります。
- ロードバランサーに静的 IP アドレスを提供する場合は、各 Elastic IP アドレスが Amazon の IPv4 アドレスのプールからのものであり、ロードバランサーと同じネットワーク境界グループを持つことを確認してください。
- QUIC または TCP_QUIC リスナーを使用する予定の場合は、Network Load Balancer が `ipv4` アドレスタイプを使用し、セキュリティグループが関連付けられていないことを確認します。

ロードバランサーを作成する

Network Load Balancer の作成の一環として、ロードバランサー、少なくとも 1 つのリスナー、少なくとも 1 つのターゲットグループを作成します。有効な各アベイラビリティーゾーンに正常な登録済みターゲットが 1 つ以上ある場合、ロードバランサーはクライアントリクエストを処理する準備ができています。

Console

Network Load Balancer を作成するには

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、[ロードバランサー] を選択します。
3. [ロードバランサーを作成] を選択します。
4. [Network Load Balancer] で、[Create] (作成) を選択します。
5. 基本的な設定

    a. [ロードバランサー名] に、Network Load Balancer の名前を入力します。名前は、リージョンのロードバランサーのセット内で一意である必要があります。これは最大 32 文字で、英数字とハイフンのみを使用できます。先頭および末尾にハイフンまたは `internal-` を使用することはできません。

    b. [スキーム] で、[インターネット向け] または [内部] を選択します。インターネット向け Network Load Balancer は、クライアントからインターネット経由でリクエストをターゲットにルーティングします。内部 Network Load Balancer は、プライベート IP アドレスを使用してターゲットにリクエストをルーティングします。

c. [ロードバランサーの IP アドレスタイプ] については、クライアントが Network Load Balancer との通信に IPv4 アドレスを使用する場合は [IPv4] を、クライアントが Network Load Balancer との通信に IPv4 アドレスと IPv6 アドレスの両方を使用する場合は [デュアルスタック] を選択します。

6. ネットワークマッピング

   a. VPC の場合は、ロードバランサー用に準備した VPC を選択します。インターネット向けロードバランサーでは、インターネットゲートウェイを持つ VPC のみを選択できます。

   b. デュアルスタックロードバランサーでは、[IPv6 ソース NAT のプレフィックスを有効化] が [オン] (サブネットあたりのソース NAT プレフィックス) でない限り、UDP リスナーを追加することはできません。

   c. アベイラビリティーゾーンとサブネットの場合は、1 つ以上のアベイラビリティーゾーンを選択し、ゾーンごとに 1 つのサブネットを選択します。共有されたサブネットを選択できることに注意してください。

      複数のアベイラビリティーゾーンを選択し、選択した各ゾーンにターゲットが登録されていることを確認すると、アプリケーションの耐障害性が向上します。

   d. インターネット向けロードバランサーの場合、各アベイラビリティーゾーンに Elastic IP アドレスを選択できます。これにより、ロードバランサーに静的 IP アドレスが提供されます。

      内部ロードバランサーでは、各サブネットのアドレス範囲からプライベート IPv4 アドレスを入力するか、 に AWS 選択させることができます。

      デュアルスタックロードバランサーでは、各サブネットのアドレス範囲から IPv6 アドレスを入力するか、 に AWS 選択させることができます。

      ソース NAT が有効になっているロードバランサーの場合は、カスタム IPv6 プレフィックスを入力するか、 に AWS 選択させることができます。

7. セキュリティグループ

   ロードバランサー VPC のデフォルトのセキュリティグループを AWS が事前に選択します。必要に応じて、追加のセキュリティグループを選択できます。適切なセキュリティグループがない場合は、[新しいセキュリティグループを作成]を選択して今すぐ新しいセキュリティグループを作成します。詳細については、「Amazon VPC ユーザーガイド」の「セキュリティグループの作成」を参照してください。

> **Warning**
> この時点で Network Load Balancer にセキュリティグループを関連付けていない場合、後で関連付けすることはできません。

> **Warning**
> QUIC または TCP_QUIC リスナーを活用するには、Network Load Balancer にセキュリティグループがあってはなりません。

8. リスナーとルーティング

   a. デフォルトは、ポート 80 で TCP トラフィックを受け付けるリスナーです。必要に応じて、デフォルトのリスナー設定を保持する、または [プロトコル] または [ポート] を変更することができます。

   b. [Default action] (デフォルトアクション) では、トラフィックを転送するターゲットグループを選択します。

      別のターゲットグループを追加するには、[ターゲットグループを追加] を選択し、必要に応じて重みを更新します。

      ニーズに合ったターゲットグループがない場合は、[ターゲットグループを作成] を選択して今すぐ作成します。詳細については、「ターゲットグループの作成」を参照してください。

   c. (オプション) [リスナータグを追加] をクリックし、タグキーとタグ値を入力します。

   d. (オプション) [リスナーを追加] を選択して別のリスナー (TLS リスナーなど) を追加できます。

9. セキュアリスナー設定

   このセクションは、TLS リスナーを追加する場合にのみ表示されます。

   a. [セキュリティポリシー] で、要件を満たすセキュリティポリシーを選択します。詳細については、「セキュリティポリシー」を参照してください。

   b. デフォルトの SSL/TLS サーバー証明書の場合は、証明書ソースとして [ACM から] を選択します。 AWS Certificate Managerを使用してプロビジョニングまたはインポートした証明書を選択します。ACM で使用可能な証明書がないが、ロードバランサーで使

用する証明書がある場合は、[証明書をインポート] を選択し、必要な情報を入力します。それ以外の場合は、[証明書をリクエスト] を選択します。詳細については、「AWS Certificate Manager ユーザーガイド」の「AWS Certificate Manager 証明書」を参照してください。

c. (オプション) [ALPN ポリシー] で、ALPN を有効にするポリシーを選択します。詳細については、「the section called “ALPN ポリシー”」を参照してください。

10. ロードバランサータグ

    (オプション) [ロードバランサータグ] を展開します。(オプション) [新しいタグを追加] をクリックし、タグキーとタグ値を入力します。詳細については、「タグ」を参照してください。

11. [概要]

    設定を確認し、[ロードバランサーの作成] を選択します。作成時に、Network Load Balancer にいくつかのデフォルト属性が適用されます。Network Load Balancer の作成後に、それらを表示および編集できます。詳細については、「ロードバランサーの属性」を参照してください。

AWS CLI

Network Load Balancer を作成するには

create-load-balancer コマンドを使用します。

次の例では、2 つの有効なアベイラビリティーゾーンとセキュリティグループを持つインターネット向けロードバランサーを作成します。

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --type network \
    --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
    --security-groups sg-1111222233334444
```

内部 Network Load Balancer を作成するには

次の例のように、`--scheme` オプションを含めます。

```
aws elbv2 create-load-balancer \
```

```
    --name my-load-balancer \
    --type network \
    --scheme internal \
    --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
    --security-groups sg-1111222233334444
```

デュアルスタック Network Load Balancer を作成するには

次の例のように、`--ip-address-type` オプションを含めます。

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --type network \
    --ip-address-type dualstack \
    --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
    --security-groups sg-1111222233334444
```

リスナーを追加するには

create-listener コマンドを使用します。例については「リスナーの作成」を参照してください。

CloudFormation

Network Load Balancer を作成するには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースタイプを定義します。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      IpAddressType: dualstack
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      Tags:
        - Key: 'department'
```

```
            Value: '123'
```

リスナーを追加するには

AWS::ElasticLoadBalancingV2::Listener リソースタイプを定義します。例については「リスナーの作成」を参照してください。

ロードバランサーをテストするには

Network Load Balancer を作成したら、EC2 インスタンスが最初のヘルスチェックに合格したことを確認してから、Network Load Balancer が EC2 インスタンスにトラフィックを送信することをテストできます。Network Load Balancer を削除するには、「Network Load Balancer を削除する」を参照してください。

Network Load Balancer をテストするには

1. Network Load Balancer が作成されたら、[閉じる] を選択します。
2. 左側のナビゲーションペインで、[ターゲットグループ] を選択します。
3. 新しいターゲットグループを選択します。
4. [Targets] を選択して、インスタンスの準備ができていることを確認します。インスタンスのステータスが `initial` の場合、インスタンスがまだ登録の途中であるか、正常と見なされるのに必要なヘルスチェックの最小数に合格しなかったと考えられます。少なくとも 1 つのインスタンスのステータスが正常であれば、Network Load Balancer をテストできます。詳細については、「ターゲットヘルスステータス」を参照してください。
5. ナビゲーションペインで、[ロードバランサー] を選択します。
6. 新しい Network Load Balancer を選択します。
7. Network Load Balancer の DNS 名 (my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com など) をコピーします。インターネットに接続したウェブブラウザのアドレスフィールドに DNS 名を貼り付けます。すべて適切な場合は、ブラウザにサーバーのデフォルトページが表示されます。

次の手順

ロードバランサーを作成したら、次の操作を行います。

- ロードバランサーの属性を設定します。

- ターゲットグループ属性を設定します。
- [TLS リスナー] オプションの証明書リストに証明書を追加します。
- モニタリング機能を設定します。

Network Load Balancer のアベイラビリティーゾーンを更新する

アベイラビリティーゾーンは、Network Load Balancer に対していつでも有効または無効にできます。アベイラビリティーゾーンを有効にしたら、そのアベイラビリティーゾーンからサブネットを 1 つ指定する必要があります。アベイラビリティーゾーンを有効にしたら、ロードバランサーはこれらのアベイラビリティーゾーン内の登録済みターゲットにリクエストをルーティングするようになります。有効な各アベイラビリティーゾーンに少なくとも 1 つの登録済みターゲットがあるようにする場合、ロードバランサーが最も効果的です。複数のアベイラビリティーゾーンを有効にすると、アプリケーションの耐障害性の向上に役立ちます。

Elastic Load Balancing は、選択したアベイラビリティーゾーンに Network Load Balancer ノードを作成し、そのアベイラビリティーゾーンに選択したサブネットのネットワークインターフェイスを作成します。アベイラビリティーゾーンの各 Network Load Balancer ノードは、このネットワークインターフェイスを使用して IPv4 アドレスを取得します。これらのネットワークインターフェイスは表示できますが、変更することはできません。

考慮事項

- インターネット向け Network Load Balancer の場合、指定するサブネットには最低 8 個の利用可能な IP アドレスが必要です。内部 Network Load Balancer の場合、サブネットからプライベート IPv4 アドレス AWS を選択できる場合にのみ必要です。
- 制約のあるアベイラビリティーゾーンにあるサブネットを指定することはできません。ただし、制約されていないアベイラビリティーゾーンにあるサブネットを指定し、クロスゾーン負荷分散を使用して、制約されているアベイラビリティーゾーンのターゲットにトラフィックを分散することはできます。
- ローカルゾーンでサブネットを指定することはできません。
- Network Load Balancer にアクティブな Amazon VPC エンドポイントの関連付けがある場合、サブネットを削除することはできません。
- 以前に削除したサブネットを追加すると、別の ID で新しいネットワークインターフェイスが作成されます。
- 同じアベイラビリティーゾーン内のサブネットの変更は、独立したアクションである必要があります。まず既存のサブネットの削除を完了してから、新しいサブネットを追加できます。

- サブネットの削除が完了するまでに最大 3 分かかる場合があります。

インターネット向け Network Load Balancer を作成するとき、各アベイラビリティーゾーンに Elastic IP アドレスを指定することを選択できます。Elastic IP アドレスは Network Load Balancer に静的 IP アドレスを提供します。Elastic IP アドレスを指定しない場合、 AWS は各アベイラビリティーゾーンに 1 つの Elastic IP アドレスを割り当てます。

内部 Network Load Balancer を作成する場合、各サブネットから 1 つの プライベート IP アドレスを指定することを選択できます。プライベート IP アドレスは Network Load Balancer に静的 IP アドレスが提供されます。プライベート IP アドレスを指定しない場合、 はプライベート IP アドレスを AWS 割り当てます。

Network Load Balancer のアベイラビリティーゾーンを更新する前に、既存の接続、トラフィックフロー、または本番稼働用ワークロードに対する潜在的な影響を評価することをお勧めします。

> アベイラビリティーゾーンの更新は中断される可能性があります
>
> - サブネットが削除されると、関連付けられた Elastic Network Interface (ENI) が削除されます。これにより、アベイラビリティーゾーン内のすべてのアクティブな接続が終了します。
> - サブネットを削除すると、サブネットが関連付けられているアベイラビリティーゾーン内のすべてのターゲットが unused としてマークされます。これにより、これらのターゲットは使用可能なターゲットプールから削除され、ターゲットへのすべてのアクティブな接続は終了します。これには、クロスゾーン負荷分散を利用するときに他のアベイラビリティーゾーンから発信される接続が含まれます。
> - Network Load Balancer の完全修飾ドメイン名 (FQDN) の有効期間 (TTL) は 60 秒です。アクティブなターゲットを含むアベイラビリティーゾーンが削除されると、DNS 解決が再度発生し、トラフィックが残りのアベイラビリティーゾーンにシフトされるまで、既存のクライアント接続でタイムアウトが発生する可能性があります。

Console

アベイラビリティーゾーンを変更するには

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、[ロードバランサー] を選択します。

3. ロードバランサーを選択します。
4. [Network mapping] (ネットワークマッピング) タブで、[Edit subnets] (サブネットの編集) を選択します。
5. アベイラビリティーゾーンを有効にするには、そのチェックボックスを選択し、サブネットを 1 つ選択します。使用可能なサブネットが 1 つしかない場合は、それが選択されます。
6. 有効なアベイラビリティーゾーンのサブネットを変更するには、リストから他のサブネットのいずれかを選択します。
7. アベイラビリティゾーンを無効にするには、そのチェックボックスをオフにします。
8. [Save changes] (変更の保存) をクリックします。

AWS CLI

アベイラビリティーゾーンを変更するには

set-subnets コマンドを使用します。

```
aws elbv2 set-subnets \
    --load-balancer-arn load-balancer-arn \
    --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890
```

CloudFormation

アベイラビリティーゾーンを変更するには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースを更新します。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref new-subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
```

Network Load Balancer の IP アドレスタイプを更新する

Network Load Balancer は、クライアントが IPv4 アドレスのみを使用して Network Load Balancer と通信できるように設定する、または IPv4 アドレスと IPv6 アドレスの両方 (デュアルスタック) を使用してロードバランサーと通信できるように設定することができます。Network Load Balancer は、ターゲットグループの IP アドレスのタイプに基づいてターゲットと通信します。詳細については、「IP アドレスタイプ」を参照してください。

デュアルスタックの要件

- Network Load Balancer の作成時に IP アドレスタイプを設定し、いつでも更新できます。
- Network Load Balancer に指定する 仮想プライベートクラウド (VPC) とサブネットには、IPv6 CIDR ブロックが関連付けられている必要があります。詳細については、Amazon EC2 ユーザーガイドの IPv6 アドレスを参照してください。
- Network Load Balancer サブネットのルートテーブルは、IPv6 トラフィックをルーティングする必要があります。
- Network Load Balancer サブネットのネットワーク ACL は、IPv6 トラフィックを許可する必要があります。
- Network Load Balancer にアタッチされた QUIC または TCP_QUIC リスナーはありません。

Console

IP アドレスタイプを更新するには

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、[ロードバランサー] を選択します。
3. Network Load Balancer のチェックボックスをオンにします。
4. [Actions]、[Edit IP address type] を選択します。
5. [IP アドレスタイプ] で、[IPv4] を選択して IPv4 アドレスのみをサポートするか、[デュアルスタック] を選択して IPv4 と IPv6 アドレスの両方をサポートします。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

IP アドレスタイプを更新するには

set-ip-address-type コマンドを使用します。

```
aws elbv2 set-ip-address-type \
    --load-balancer-arn load-balancer-arn \
    --ip-address-type dualstack
```

CloudFormation

IP アドレスタイプを更新するには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースを更新します。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      IpAddressType: dualstack
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
```

Network Load Balancer の属性を編集する

Network Load Balancer を作成したら、その属性を編集できます。

ロードバランサーの属性

- 削除保護
- クロスゾーンロードバランサー
- アベイラビリティゾーン DNS アフィニティー
- セカンダリ IP アドレス

削除保護

Network Load Balancer が誤って削除されるのを防ぐために、削除保護を有効にできます。デフォルトでは、Network Load Balancer で削除保護が無効になっています。

Network Load Balancer の削除保護を有効にした場合、Network Load Balancer を削除する前に無効にする必要があります。

Console

削除保護を有効または無効にするには

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、[ロードバランサー] を選択します。
3. Network Load Balancer の名前を選択して、その詳細ページを開きます。
4. [属性] タブで、[編集] を選択します。
5. [保護] で、[削除保護] を有効または無効にします。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

削除保護を有効または無効にするには

`deletion_protection.enabled` 属性を指定して modify-load-balancer-attributes コマンドを使用します。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=deletion_protection.enabled,Value=true"
```

CloudFormation

削除保護を有効または無効にするには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースを更新して、`deletion_protection.enabled` 属性を含めます。

```
Resources:
```

```
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "deletion_protection.enabled"
          Value: "true"
```

クロスゾーンロードバランサー

Network Load Balancer では、クロスゾーンロードバランサーは、ロードバランサーレベルでのデフォルトでオフになっていますが、いつでもオンにすることができます。ターゲットグループの場合、デフォルトではロードバランサー設定を使用しますが、ターゲットグループレベルでクロスゾーンロードバランサーを明示的にオンまたはオフにすることでデフォルトを上書きできます。詳細については、「the section called “クロスゾーンロードバランサー”」を参照してください。

Console

ロードバランサーのクロスゾーン負荷分散を有効または無効にするには

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の [Load Balancing] で、[Load Balancers] を選択します。
3. ロードバランサーの名前を選択して、その詳細ページを開きます。
4. [属性] タブで、[編集] を選択します。
5. [Edit load balancer attributes] (ロードバランサー属性の編集) ページで、[Cross-zone load balancing] (クロスゾーンロードバランサー) をオンまたはオフにします。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

ロードバランサーのクロスゾーン負荷分散を有効または無効にするには

load_balancing.cross_zone.enabled 属性を指定して modify-load-balancer-attributes コマンドを使用します。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=load_balancing.cross_zone.enabled,Value=true"
```

CloudFormation

ロードバランサーのクロスゾーン負荷分散を有効または無効にするには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースを更新して、load_balancing.cross_zone.enabled 属性を含めます。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "load_balancing.cross_zone.enabled"
          Value: "true"
```

アベイラビリティゾーン DNS アフィニティー

デフォルトのクライアントルーティングポリシーを使用すると、Network Load Balancer DNS 名に送信されたリクエストには、正常な Network Load Balancer の IP アドレスがすべて届きます。これにより、Network Load Balancer のアベイラビリティーゾーン全体にクライアント接続が分散されます。アベイラビリティーゾーンのアフィニティールーティングポリシーでは、クライアント DNS クエリは自身のアベイラビリティーゾーン内の Network Load Balancer の IP アドレスを優先します。これにより、クライアントがターゲットに接続する際にアベイラビリティーゾーンの境界を越える必要がなくなるため、レイテンシーと回復性の両方が向上します。

アベイラビリティーゾーンのアフィニティールーティングポリシーは、Route 53 Resolver を使用してネットワークロードバランサーの DNS 名を解決するクライアントにのみ適用されます。詳細については、「Amazon Route 53 デベロッパーガイド」の「Amazon Route 53 Resolver とは」を参照してください。

Route 53 リゾルバーを使用してネットワークロードバランサーで使用できるクライアントルーティングポリシー:

- アベイラビリティーゾーンのアフィニティ – 100% のゾーンアフィニティ

  クライアントの DNS クエリでは、自身のアベイラビリティーゾーンの Network Load Balancer の IP アドレスが優先されます。自身のゾーンに正常な Network Load Balancer の IP アドレスがない場合、クエリは他のゾーンで解決される可能性があります。
- 部分的アベイラビリティーゾーンのアフィニティ – 85% のゾーンアフィニティ

  クライアントの DNS クエリの 85% は自身のアベイラビリティーゾーンにある Network Load Balancer の IP アドレスを優先し、残りのクエリは正常な任意のゾーンで解決されます。自身のゾーンに正常な IP アドレスがない場合、クエリは他の正常なゾーンで解決される可能性があります。どのゾーンにも正常な IP アドレスがない場合、クエリは任意のゾーンで解決されます。
- 任意のアベイラビリティーゾーン (デフォルト) – 0% のゾーンアフィニティ

  クライアント DNS クエリは、すべての Network Load Balancer アベイラビリティーゾーンの正常な Network Load Balancer の IP アドレスで解決されます。

アベイラビリティーゾーンのアフィニティはクライアントから Network Load Balancer にリクエストをルーティングするのに役立ち、クロスゾーン負荷分散は Network Load Balancer からターゲットにリクエストをルーティングするのに役立ちます。アベイラビリティーゾーンのアフィニティを使用するときは、クロスゾーン負荷分散をオフにして、クライアントからターゲットへの Network Load Balancer トラフィックが同じアベイラビリティーゾーン内に保持されるようにします。この設定では、クライアントトラフィックは同じ Network Load Balancer アベイラビリティーゾーンに送信されるため、各アベイラビリティーゾーンで個別にスケーリングするようにアプリケーションを設定することをお勧めします。これは、アベイラビリティーゾーンあたりのクライアント数、またはアベイラビリティーゾーンあたりのトラフィックが同じでない場合の重要な考慮事項です。詳細については、「ターゲットグループに対するクロスゾーン負荷分散」を参照してください。

アベイラビリティーゾーンに異常があると見なされた場合や、ゾーンシフトが開始された場合は、フェールオープンが有効でない限り、ゾーン IP アドレスは異常と見なされ、クライアントには返されません。DNS レコードがオープンに失敗しても、アベイラビリティーゾーンのアフィニティは維

持されます。これにより、アベイラビリティーゾーンの独立性が保たれ、ゾーン間で発生する可能性のある障害を防ぐことができます。

アベイラビリティーゾーンのアフィニティを使用すると、アベイラビリティーゾーン間でバランスが崩れることが予想されます。各アベイラビリティーゾーンのワークロードをサポートするために、ターゲットがゾーンレベルでスケーリングされていることを確認することをお勧めします。これらの不均衡が著しい場合は、アベイラビリティーゾーンのアフィニティをオフにすることをお勧めします。これにより、60 秒以内、つまり DNS TTL の範囲内で、すべての Network Load Balancer のアベイラビリティーゾーン間でクライアント接続を均等に分散できます。

アベイラビリティゾーンアフィニティを使用する前に、以下の点を考慮してください。

- アベイラビリティーゾーンのアフィニティにより、Route 53 Resolver を使用しているすべての Network Load Balancer クライアントに変化が生じます。
  - クライアントは、ゾーンローカル DNS 解決とマルチゾーン DNS 解決を区別できません。アベイラビリティーゾーンのアフィニティが判断します。
  - アベイラビリティーゾーンのアフィニティの影響を受けるタイミングや、どの IP アドレスがどのアベイラビリティーゾーンにあるかを知る信頼できる方法がクライアントには提供されません。
- Network Load Balancer と Route 53 Resolver でアベイラビリティーゾーンのアフィニティを使用する場合は、クライアントが独自のアベイラビリティーゾーンで Route 53 Resolver インバウンドエンドポイントを使用することをお勧めします。
- DNS ヘルスチェックにより完全に異常であると判断され、DNS から削除されるまで、クライアントはゾーンローカル IP アドレスに割り当てられたままになります。
- クロスゾーン負荷分散がオンになっているアベイラビリティーゾーンのアフィニティを使用すると、アベイラビリティーゾーン間のクライアント接続の分散が不均衡になる可能性があります。各アベイラビリティーゾーンで個別にスケールするようにアプリケーションスタックを設定し、アプリケーションスタックがゾーンクライアントのトラフィックをサポートできるようにすることをお勧めします。
- クロスゾーン負荷分散がオンになっている場合、Network Load Balancer はクロスゾーンの影響を受けます。
- Network Load Balancer の各アベイラビリティーゾーンの負荷は、クライアントのリクエストのゾーンロケーションに比例します。どのアベイラビリティーゾーンで、いくつのクライアントを実行するかを設定しない場合は、各アベイラビリティーゾーンを事後的に個別にスケーリングする必要があります。

モニタリング

ゾーン Network Load Balancer メトリクスを使用して、アベイラビリティーゾーン間の接続の分散を追跡することをお勧めします。メトリクスを使用して、ゾーンごとの新規接続数およびアクティブ接続数を表示できます。

次の点を追跡することをおすすめします。

- **ActiveFlowCount** – クライアントからターゲットへの同時フロー (または接続) の合計数。
- **NewFlowCount** – 期間内にクライアントからターゲットに確立された新しいフロー (または接続) の合計数。
- **HealthyHostCount** – 正常と見なされるターゲットの数。
- **UnHealthyHostCount** – 異常とみなされるターゲットの数。

詳細については、Network Load Balancer の CloudWatch メトリクスを参照してください。

アベイラビリティーゾーンのアフィニティを有効にする

Console

アベイラビリティーゾーンのアフィニティを有効にするには

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、[ロードバランサー] を選択します。
3. Network Load Balancer の名前を選択して、その詳細ページを開きます。
4. [属性] タブで、[編集] を選択します。
5. [アベイラビリティーゾーンのルーティング設定] の [クライアントルーティングポリシー (DNS レコード)] で、[アベイラビリティーゾーンのアフィニティ] または [Partial Availability Zone affinity] (部分的アベイラビリティーゾーンのアフィニティ) を選択します。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

アベイラビリティーゾーンのアフィニティを有効にするには

`dns_record.client_routing_policy` 属性を指定して modify-load-balancer-attributes コマンドを使用します。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes
 "Key=dns_record.client_routing_policy,Value=partial_availability_zone_affinity"
```

CloudFormation

アベイラビリティーゾーンのアフィニティを有効にするには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースを更新して、dns_record.client_routing_policy 属性を含めます。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "dns_record.client_routing_policy"
          Value: "partial_availability_zone_affinity"
```

セカンダリ IP アドレス

ポート割り当てエラーが発生し、ターゲットグループにターゲットを追加して解決できない場合は、ロードバランサーネットワークインターフェイスにセカンダリ IP アドレスを追加できます。ロードバランサーが有効になっているゾーンごとに、ロードバランサーサブネットから IPv4 アドレスを選択し、対応するネットワークインターフェイスに割り当てます。これらのセカンダリ IP アドレスは、ターゲットとの接続を確立するために使用されます。また、ヘルスチェックトラフィックにも使用されます。ポート割り当てエラーが解決されない場合にのみ、開始するセカンダリ IP アドレスを 1 つ追加し、PortAllocationErrors メトリクスをモニタリングして、別のセカンダリ IP アドレスを追加することをお勧めします。

> ⚠ Warning
>
> セカンダリ IP アドレスを追加した後に、削除することはできません。セカンダリ IP アドレスを解放する唯一の方法は、ロードバランサーを削除することです。セカンダリ IP アドレスを追加する前に、ロードバランサーサブネットに十分な使用可能な IPv4 アドレスがあることを確認します。

Console

セカンダリ IP アドレスを追加するには

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、[ロードバランサー] を選択します。
3. Network Load Balancer の名前を選択して、その詳細ページを開きます。
4. [属性] タブで、[編集] を選択します。
5. [特殊ケース属性] を展開し、[サブネットごとに自動で割り当てられたセカンダリ IP アドレス] 属性のロックを解除して、セカンダリ IP アドレスの数を選択します。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

セカンダリ IP アドレスを追加するには

`secondary_ips.auto_assigned.per_subnet` 属性を指定して modify-load-balancer-attributes コマンドを使用します。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=secondary_ips.auto_assigned.per_subnet,Value=1"
```

describe-network-interfaces コマンドを使用して、ロードバランサーネットワークインターフェイスの IPv4 アドレスを取得できます。`--filters` パラメータは Network Load Balancer のネットワークインターフェイスに結果をスコープし、`--query` パラメータが指定された名前のロードバランサーに結果をスコープして、指定されたフィールドのみを表示します。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 elbv2 describe-network-interfaces \
    --filters "Name=interface-type,Values=network_load_balancer" \
    --query "NetworkInterfaces[?contains(Description,'my-nlb')].
{ID:NetworkInterfaceId,AZ:AvailabilityZone,Addresses:PrivateIpAddresses[*]}"
```

CloudFormation

セカンダリ IP アドレスを追加するには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースを更新して、`secondary_ips.auto_assigned.per_subnet` 属性を含めます。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "secondary_ips.auto_assigned.per_subnet"
          Value: "1"
```

Network Load Balancer のセキュリティグループを更新する

セキュリティグループを Network Load Balancer に関連付けて、Network Load Balancer へのインバウンド/アウトバウンドのトラフィックを制御できます。インバウンドトラフィックを許可するポート、プロトコル、ソース、およびアウトバウンドトラフィックを許可するポート、プロトコル、および送信先を指定します。Network Load Balancer にセキュリティグループを割り当てないと、すべてのクライアントトラフィックが Network Load Balancer リスナーに到達し、すべてのトラフィックが Network Load Balancer を離れる可能性があります。

ターゲットに関連付けられたセキュリティグループに、Network Load Balancer に関連付けられたセキュリティグループを参照するルールを追加できます。これにより、クライアントは Network Load

Balancer を介してターゲットへトラフィックを送信できるようになりますが、直接ターゲットへ送信することはできません。ターゲットに関連付けられたセキュリティ グループで Network Load Balancer に関連付けられたセキュリティグループが参照されることで、Network Load Balancer に対して[クライアント IP の保存](#)を有効にしている場合でも、ターゲットは Network Load Balancer からのトラフィックを確実に受信できます。

インバウンドセキュリティグループルールによってブロックされたトラフィックに対しては料金が発生しません。

内容

考慮事項

- Network Load Balancer を作成するときに、セキュリティグループを Network Load Balancer に関連付けることができます。セキュリティグループを関連付けずに Network Load Balancer を作成した場合、後でセキュリティグループを Network Load Balancer に関連付けることはできません。Network Load Balancer を作成するときに、セキュリティグループを Network Load Balancer に関連付けることをお勧めします。
- セキュリティグループを関連付けて Network Load Balancer を作成した後は、Network Load Balancer に関連付けられたセキュリティグループはいつでも変更できます。
- ヘルスチェックにはアウトバウンドルールが適用されますが、インバウンドルールは適用されません。アウトバウンドルールがヘルスチェックトラフィックをブロックしないようにする必要があります。そうしないと、Network Load Balancer はターゲットに異常があると見なします。
- PrivateLink トラフィックがインバウンドルールの対象となるかどうかを制御できます。PrivateLink トラフィックのインバウンドルールを有効にすると、トラフィックの送信元はエンドポイントのインターフェイスではなく、クライアントのプライベート IP アドレスになります。

例: クライアントトラフィックのフィルタリング

以下に示すように、Network Load Balancer に関連付けられているセキュリティグループのインバウンドルールでは、指定されたアドレス範囲からのトラフィックのみが許可されます。これが内部の Network Load Balancer の場合には、VPC CIDR 範囲をソースとして指定して、特定の VPC からのトラフィックのみを許可できます。これが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必要があるインターネット向け Network Load Balancer の場合は、ソースとして 0.0.0.0/0 を指定できます。

インバウンド

プロトコル	ソース	ポート範囲	コメント
protocol	*###### IP ## ####*	*#######*	リスナーポート上の CIDR からのインバウンドトラフィックを許可します
ICMP	0.0.0.0/0	すべて	インバウンド ICMP トラフィックが MTU またはパス MTU ディスカバリー † をサポートできるようにします †

† 詳細については、「Amazon EC2 ユーザーガイド」の「パス MTU 検出」を参照してください。

アウトバウンド

プロトコル	目的地	ポート範囲	コメント
すべて	どこでも	すべて	すべてのアウトバウンドトラフィックを許可します

例: Network Load Balancer からのトラフィックのみを受け入れる

Network Load Balancer に sg-111112222233333 というセキュリティグループがあるとします。ターゲット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で次のルールを使用して、Network Load Balancer からのトラフィックのみを受け付けるようにします。ターゲットがターゲットポートとヘルスチェックポートの両方で Network Load Balancer からのトラフィックを確実に受信できる

ようにする必要があります。詳細については、「the section called “ターゲットセキュリティグループ”」を参照してください。

インバウンド

プロトコル	ソース	ポート範囲	コメント
protocol	sg-111112 222233333	*########*	ターゲットポートの Network Load Balancer からのインバウンドトラフィックを許可します
protocol	sg-111112 222233333	*#######*	ヘルスチェックポートで Network Load Balancer からの受信トラフィックを許可します

アウトバウンド

プロトコル	目的地	ポート範囲	コメント
すべて	どこでも	いずれか	すべてのアウトバウンドトラフィックを許可します

関連付けられたセキュリティグループの更新

Network Load Balancer の作成時に少なくとも 1 つのセキュリティグループを Network Load Balancer に関連付けていた場合は、その Network Load Balancer のセキュリティグループをいつでも更新できます。

Console

セキュリティグループを更新するには

1. Amazon EC2 コンソールの 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の [ロードバランシング] で [ロードバランサー] を選択します。
3. Network Load Balancer を選択します。
4. [セキュリティ] タブで、[編集] を選択します。

5. セキュリティグループを Network Load Balancer に関連付けるには、そのセキュリティグループを選択します。セキュリティグループを Network Load Balancer から削除するには、そのセキュリティグループを選択解除します。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

セキュリティグループを更新するには

set-security-groups コマンドを使用します。

```
aws elbv2 set-security-groups \
    --load-balancer-arn load-balancer-arn \
    --security-groups sg-1234567890abcdef0 sg-0abcdef0123456789
```

CloudFormation

セキュリティグループを更新するには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースを更新します。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
        - !Ref myNewSecurityGroup
```

セキュリティ設定の更新

デフォルトでは、Network Load Balancer に送信されるすべてのトラフィックにインバウンドセキュリティグループのルールが適用されます。ただし、重複する IP アドレスから発生する可能性のある Network Load Balancer に送信されるトラフィックには AWS PrivateLink、これらのルールを適用し

たくない場合があります。この場合、Network Load Balancer に送信されるトラフィックにインバウンドルールを適用しないように Network Load Balancer を設定できます AWS PrivateLink。

Console

セキュリティ設定を更新するには

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の [ロードバランシング] で [ロードバランサー] を選択します。
3. Network Load Balancer を選択します。
4. [セキュリティ] タブで、[編集] を選択します。
5. [セキュリティ設定] で、[PrivateLink トラフィックにインバウンドルールを適用する] をオフにします。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

セキュリティ設定を更新するには

set-security-groups コマンドを使用します。

```
aws elbv2 set-security-groups \
    --load-balancer-arn load-balancer-arn \
    --enforce-security-group-inbound-rules-on-private-link-traffic off
```

CloudFormation

セキュリティ設定を更新するには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースを更新します。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      EnforceSecurityGroupInboundRulesOnPrivateLinkTraffic: off
      Subnets:
```

```
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
```

Network Load Balancer のセキュリティグループを監視する

`SecurityGroupBlockedFlowCount_Inbound` および `SecurityGroupBlockedFlowCount_Outbound` CloudWatch メトリクスを使用して、Network Load Balancer のセキュリティ グループによってブロックされているフローの数を監視します。ブロックされたトラフィックは他のメトリックには反映されません。詳細については、「the section called “CloudWatch メトリクス”」を参照してください。

VPC フローログを使用して、Network Load Balancer のセキュリティグループによって承認または拒否されたトラフィックを監視します。詳細については、Amazon VPC ユーザーガイドの VPC フローログを参照してください。

Network Load Balancer にタグを付ける

タグを使用すると、さまざまな方法で Network Load Balancer を分類できます。例えば、目的、所有者、環境などに基づいてリソースを分類できます。

各 Network Load Balancer に対して複数のタグを追加できます。すでに Network Load Balancer に関連付けられているキーを持つタグを追加すると、そのキーの値が更新されます。

タグが不要になったら、Network Load Balancer からタグを削除できます。

制限事項

- リソースあたりのタグの最大数 – 50
- キーの最大長 – 127 文字 (Unicode)
- 値の最大長 – 255 文字 (Unicode)
- タグのキーと値は大文字と小文字が区別されます。使用できる文字は、UTF-8 で表現できる文字、スペース、および数字と、特殊文字 (+、-、=、.、_、:、/、@) です。ただし、先頭または末尾にはスペースを使用しないでください。
- タグ名または値に `aws:` プレフィックスを使用しないでください。このプレフィックスは AWS 使用のために予約されています。このプレフィックスが含まれるタグの名前または値は編集または削

除できません。このプレフィックスを持つタグは、リソースあたりのタグ数の制限時には計算されません。

Console

ロードバランサーのタグを更新するには

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、[ロードバランサー] を選択します。
3. Network Load Balancer のチェックボックスをオンにします。
4. [Tags (タグ)] タブで、[Manage tags (タグ管理)] を選択します。
5. タグを追加するには、[Add tag] (タグの追加) を選択し、タグのキーとタグの値を入力します。使用できる文字は、文字、スペース、数字 (UTF-8)、および特殊文字 (+-=. _:/@) です。ただし、先頭または末尾にはスペースを使用しないでください。タグ値は大文字と小文字が区別されます。
6. タグを更新するには、[キー] と [値] に新しい値を入力します。
7. タグを削除するには、タグの横にある [削除] を選択します。
8. [Save changes] (変更の保存) をクリックします。

AWS CLI

タグを追加するには

add-tags コマンドを使用します。次の例では、2 つのタグを追加します。

```
aws elbv2 add-tags \
    --resource-arns load-balancer-arn \
    --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

タグを削除するには

remove-tags コマンドを使用します。次の例では、指定したキーを使用してタグを削除します。

```
aws elbv2 remove-tags \
    --resource-arns load-balancer-arn \
    --tag-keys project department
```

CloudFormation

タグを追加するには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースのリソースタイプを定義して、Tags プロパティを含めます。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      Tags:
        - Key: 'project'
          Value: 'lima'
        - Key: 'department'
          Value: 'digital-media'
```

Network Load Balancer を削除する

Network Load Balancer が利用可能になると、Network Load Balancer の実行時間に応じて 1 時間ごと、または 1 時間未満の時間について課金されます。不要になった Network Load Balancer は削除できます。Network Load Balancer が削除されると、Network Load Balancer の課金も停止されます。

削除保護が有効になった場合、Network Load Balancer を削除することはできません。詳細については、「削除保護」を参照してください。

別のサービスで使用中の Network Load Balancer は削除できません。たとえば、Network Load Balancer が VPC エンドポイントサービスに関連付けられている場合、関連付けられた Network Load Balancer を削除するには、まずエンドポイントサービス設定を削除する必要があります。

Network Load Balancer を削除すると、そのリスナーも削除されます。Network Load Balancer を削除しても、登録済みターゲットには影響を与えません。たとえば、EC2 インスタンスは実行を続

け、ターゲットグループに登録されたままです。ターゲットグループを削除するには、「Network Load Balancer のターゲットグループを削除する」を参照してください。

Console

Network Load Balancer を削除するには

1. Network Load Balancer をポイントするドメインの DNS レコードが存在する場合は、新しい場所にポイントして DNS の変更が有効になってから、Network Load Balancer を削除します。例えば、次のようになります。

   - 有効期限 (TTL) が 300 秒の CNAME レコードの場合は、少なくとも 300 秒待ってから次のステップに進みます。
   - Route 53 エイリアス (A) レコードの場合は、少なくとも 60 秒間待機します。
   - Route 53 を使用している場合、レコードに対する変更が世界中のすべての Route 53 ネームサーバーに反映されるまで 60 秒かかります。更新の対象となるレコードの TTL 値には、この時間を加算します。
2. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
3. ナビゲーションペインで、[ロードバランサー] を選択します。
4. Network Load Balancer のチェックボックスをオンにします。
5. [アクション] 、[ロードバランサーを削除] の順に選択します。
6. 確認を求められたら、「**confirm**」を入力し、[削除] を選択します。

AWS CLI

Network Load Balancer を削除するには

delete-load-balancer コマンドを使用します。

```
aws elbv2 delete-load-balancer \
    --load-balancer-arn load-balancer-arn
```

Network Load Balancer リソースマップを表示する

Network Load Balancer リソースマップは、関連するリスナー、ターゲットグループ、ターゲットなど、Network Load Balancer アーキテクチャのインタラクティブに表示したものです。リソースマッ

プでは、すべてのリソース間の関係とルーティングパスも強調表示され、Network Load Balancer 設定が視覚的に表示されます。

ロードバランサーのリソースマップを表示するには

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、[ロードバランサー] を選択します。
3. Network Load Balancer を選択します。
4. [リソースマップ] タブを選択します。

リソースマップの要素

マップビュー

Network Load Balancer リソースマップには、[概要] と [異常なターゲットマップ] という 2 つのビューがあります。[概要] はデフォルトで選択されており、Network Load Balancer のすべてのリソースが表示されます。[異常なターゲットマップ] ビューを選択すると、異常なターゲットとそのターゲットに関連付けられたリソースのみが表示されます。

[異常なターゲットマップ] は、ヘルスチェックに失敗したターゲットのトラブルシューティングに使用できます。詳細については、「リソースマップを使用して異常なターゲットをトラブルシューティングする」を参照してください。

リソース列

Network Load Balancer リソースマップには、3 つのリソース列が含まれており、それぞれが各リソースタイプに対応しています。リソースグループは、[リスナー]、[ターゲットグループ]、[ターゲット] です。

リソースタイル

列内の各リソースには固有のタイルがあり、その特定のリソースの詳細が表示されます。

- リソースタイルにカーソルを合わせると、そのリソースと他のリソースとの関係が強調表示されます。
- リソースタイルを選択すると、そのリソースと他のリソースとの関係が強調表示され、そのリソースに関する追加の詳細が表示されます。
  - [ターゲットグループのヘルスサマリー]: 各ヘルスステータスの登録済みターゲットの数。

- [ターゲットのヘルスステータス]: ターゲットの現在のヘルスステータスと説明。

> Note
>
> [リソース詳細を表示] をオフにして、リソースマップ内の追加の詳細を非表示にすることができます。

- 各リソースタイルには、選択するとそのリソースの詳細ページが開くリンクが含まれています。
  - リスナー - リスナーの protocol:port を選択します。例: `TCP:80`
  - ターゲットグループ - ターゲットグループ名を選択します。例: `my-target-group`
  - ターゲット - ターゲット ID を選択します。例: `i-1234567890abcdef0`

リソースマップをエクスポートする

[エクスポート] を選択すると、Network Load Balancer のリソースマップの現在のビューを PDF としてエクスポートできます。

Network Load Balancer の CloudWatch ログ

Amazon CloudWatch Logs は、Network Load Balancer アクセスログを販売ログとしてサポートし、オブザーバビリティを向上させ、ネットワークトラフィックパターンのデバッグを簡素化します。Network Load Balancer アクセスログを CloudWatch で直接分析して、クライアント接続、トラフィック分散、接続ステータスに関するインサイトを取得し、ネットワークの問題をより迅速に特定してトラブルシューティングすることができます。

Apache Parquet 形式をサポートする Amazon CloudWatch Logs、Amazon Data Firehose、Amazon Simple Storage Service (Amazon S3) への Network Load Balancer アクセスログの配信を設定できます。

> Important
>
> アクセスログが作成されるのは、ロードバランサーに TLS リスナーがあり、TLS リクエストに関する情報のみが含まれる場合のみです。アクセスログは、ベストエフォートベースでリクエストを記録します。アクセスログは、すべてのリクエストを完全に報告するためのものではなく、リクエストの本質を把握するものとして使用することをお勧めします。

> ⚠ Important
>
> 従来の「レガシー」アクセスログは、Network Load Balancer で引き続き使用できます。レガシーアクセスログの設定を管理するには、ロードバランサーの [属性] タブにアクセスします。「レガシー」アクセスログの詳細については、「Network Load Balancer のアクセスログ」を参照してください。

この CloudWatch Logs 統合により、CloudWatch Logs Insights クエリを使用して詳細なアクセスパターンを追跡したり、モニタリング用のメトリクスフィルターを作成したり、Live Tail を使用してトラフィックパターンをリアルタイムで確認したりできます。

Network Load Balancer の CloudWatch Logs アクセスログは、コンソールのロードバランサーの [統合] タブから有効にできます。ログ記録を有効にするには、特定のアクセス許可を持つユーザーとしてログインする必要があります。さらに、ログの送信を有効にする AWS には、 にアクセス許可を付与する必要があります。

各ログ記録先に必要なアクセス許可については、「 AWS サービスからのログ記録を有効にする」を参照してください。

詳細については、「Amazon CloudWatch Logs とは」を参照してください。

詳細については、Amazon CloudWatch 料金表をご覧ください。

Network Load Balancer のゾーンシフト

ゾーンシフトは Amazon Application Recovery Controller (ARC) の機能です。ゾーンシフトを使用すると、1 回のアクションで Network Load Balancer のリソースを障害のあるアベイラビリティーゾーンから移動できます。このようにして、 AWS リージョンの他の正常なアベイラビリティーゾーンから操作を継続できます。

ゾーンシフトを開始すると、Network Load Balancer は影響を受けるアベイラビリティーゾーンのターゲットへのトラフィックのルーティングを停止します。影響を受けるアベイラビリティーゾーン内のターゲットへの既存の接続は、ゾーンシフトによって終了されません。これらの接続が正常に完了するまでに数分かかる場合があります。

内容

- Network Load Balancer のゾーンシフトを有効にする
- Network Load Balancer のゾーンシフトを開始する
- Network Load Balancer のゾーンシフトを更新する
- Network Load Balancer のゾーンシフトをキャンセルする

ゾーンシフトを開始する前に

- ゾーンシフトはデフォルトで無効になっており、各 Network Load Balancer で有効にする必要があります。詳細については、「Network Load Balancer のゾーンシフトを有効にする」を参照してください。
- 特定の Network Load Balancer のゾーンシフトは 1 つのアベイラビリティーゾーンに対してのみ開始できます。複数のアベイラビリティーゾーンに対してゾーンシフトを開始することはできません。
- AWS は、複数のインフラストラクチャの問題がサービスに影響を与える場合、DNS からゾーン Network Load Balancer IP アドレスをプロアクティブに削除します。ゾーンシフトを開始する前に、現在のアベイラビリティーゾーンの容量を必ず確認してください。Network Load Balancer でゾーンシフトを使用すると、ゾーンシフトの影響を受けるアベイラビリティーゾーンもターゲット容量を失います。
- クロスゾーン負荷分散が有効になっている Network Load Balancer のゾーンシフト中に、ゾーンロードバランサーの IP アドレスは DNS から削除されます。障害のあるアベイラビリティーゾーン内のターゲットへの既存の接続は、それらが自然に閉じられるまで保持されますが、障害のあるアベイラビリティーゾーン内のターゲットへの新しい接続はルーティングされなくなります。

詳細については、「Amazon Application Recovery Controller (ARC) デベロッパーガイド」の「Route 53 ARC ゾーンシフトのベストプラクティス」を参照してください。

ゾーンシフトの管理オーバーライド

Network Load Balancer に属するターゲットには、`TargetHealth` 状態とは独立した新しい `AdministrativeOverride` ステータスが含まれます。

Network Load Balancer のゾーンシフトが開始されると、シフトされるゾーン内のすべてのターゲットが管理上オーバーライドされたと見なされます。Network Load Balancer は、管理上オーバーライドされたターゲットへの新しいトラフィックのルーティングを停止します。既存の接続は、有機的に閉じられるまでそのまま残ります。

可能な AdministrativeOverride 状態は次のとおりです。

不明

内部エラーのため、状態を伝播できません

no_override

ターゲットで現在アクティブなオーバーライドはない

zonal_shift_active

ゾーンシフトがターゲットアベイラビリティーゾーンでアクティブです

zonal_shift_delegated_to_dns

このターゲットのゾーンシフト状態は、 では利用できませんDescribeTargetHealthが、AWS ARC - Zonal Shift API またはコンソールから直接表示できます。

Network Load Balancer のゾーンシフトを有効にする

ゾーンシフトはデフォルトで無効になっており、各 Network Load Balancer で有効にする必要があります。これにより、必要な特定の Network Load Balancer のみを使用してゾーンシフトを開始できます。詳細については、「the section called “ゾーンシフト”」を参照してください。

前提条件

ロードバランサーのクロスゾーン負荷分散を有効にする場合、ゾーンシフトを有効にする前に、ロードバランサーにアタッチされたすべてのターゲットグループが次の要件を満たしている必要があります。

- ターゲットグループプロトコルは TCP または TLS である必要があります。
- ターゲットグループタイプを alb にすることはできません。
- [異常のあるターゲットの接続終了] は無効にする必要があります。
- load_balancing.cross_zone.enabled ターゲットグループ属性は true または use_load_balancer_configuration (デフォルト) である必要があります。

Console

ゾーンシフトを有効にするには

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の [ロードバランシング] で [ロードバランサー] を選択します。
3. Network Load Balancer を選択します。
4. [属性] タブで、[編集] を選択します。
5. [アベイラビリティーゾーンルーティング設定] の [ARC ゾーンシフト統合] で [有効化] を選択します。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

ゾーンシフトを有効にするには

`zonal_shift.config.enabled` 属性を指定して modify-load-balancer-attributes コマンドを使用します。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes "Key=zonal_shift.config.enabled,Value=true"
```

CloudFormation

ゾーンシフトを有効にするには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースを更新して、`zonal_shift.config.enabled` 属性を含めます。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        -Key: "zonal_shift.config.enabled"
         Value: "true"
```

Network Load Balancer のゾーンシフトを開始する

ARC のゾーンシフトにより、サポートされているリソースのトラフィックをアベイラビリティーゾーンから一時的に移動できるため、アプリケーションは AWS リージョン内の他のアベイラビリティーゾーンで正常に動作し続けます。

前提条件

開始する前に、ロードバランサーのゾーンシフトが有効になっていることを確認します。

Console

この手順では、Amazon EC2 コンソールでゾーンシフトを開始する方法について説明します。ARC コンソールを使用してゾーンシフトを開始する手順については、「Amazon Application Recovery Controller (ARC) デベロッパーガイド」の「Starting a zonal shift」を参照してください。

ゾーンシフトを開始するには

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の [ロードバランシング] で [ロードバランサー] を選択します。
3. Network Load Balancer を選択します。
4. [統合] タブで [Amazon Application Recovery Controller (ARC)] を展開し、[ゾーンシフトを開始] を選択します。
5. トラフィックを移動させたいアベイラビリティーゾーンを選択します。
6. ゾーンシフトの有効期限を選択または入力します。ゾーンシフトは、最初は 1 分から最大 3 日 (72 時間) まで設定できます。

   すべてのゾーンシフトは一時的なものです。有効期限を設定する必要がありますが、アクティブなシフトを後で更新して有効期限を設定できます。
7. コメントを入力します。後でゾーンシフトを更新してコメントを編集できます。
8. このチェックボックスを選択して、ゾーンシフトを開始するとトラフィックがアベイラビリティーゾーンからシフトされてアプリケーションの容量が減少することを確認します。
9. [確認] を選択します。

AWS CLI

ゾーンシフトを開始するには

Amazon Application Recovery Controller (ARC) start-zonal-shift コマンドを使用します。

```
aws arc-zonal-shift start-zonal-shift \
    --resource-identifier load-balancer-arn \
    --away-from use2-az2 \
    --expires-in 2h \
    --comment "zonal shift due to scheduled maintenance"
```

Network Load Balancer のゾーンシフトを更新する

ゾーンシフトは、更新して新しい有効期限を設定できます。また、コメントを編集したり置き換えたりもできます。

Console

この手順では、Amazon EC2 コンソールを使用してゾーンシフトを更新する方法について説明します。Amazon Application Recovery Controller コンソールを使用してゾーンシフトを更新する手順については、「Amazon Application Recovery Controller (ARC) Developer Guide」の「Update a zonal shift」を参照してください。

ゾーンシフトを更新するには

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の [ロードバランシング] で [ロードバランサー] を選択します。
3. アクティブなゾーンシフトを持つ Application Load Balancer を選択します。
4. [統合] タブで [Amazon Application Recovery Controller (ARC)] を展開し、[ゾーンシフトを更新] を選択します。

   これにより ARC コンソールが開き、更新プロセスが続行されます。
5. (オプション) [ゾーンシフトの有効期限を設定] で、有効期限を選択または入力します。
6. (オプション) [コメント] では、必要に応じて既存のコメントを編集するか、新しいコメントを入力します。
7. [更新] を選択します。

AWS CLI

ゾーンシフトを更新するには

Amazon Application Recovery Controller (ARC) update-zonal-shift コマンドを使用します。

```
aws arc-zonal-shift update-zonal-shift \
    --zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf57EXAMPLE \
    --expires-in 1h \
    --comment "extending zonal shift for scheduled maintenance"
```

Network Load Balancer のゾーンシフトをキャンセルする

ゾーンシフトは、有効期限が切れる前であればいつでもキャンセルできます。開始したゾーンシフト、またはゾーンオートシフトの練習実行のリソースに対して AWS 開始したゾーンシフトをキャンセルできます。

Console

この手順では、Amazon EC2 コンソールを使用してゾーンシフトをキャンセルする方法について説明します。Amazon Application Recovery Controller コンソールを使用してゾーンシフトをキャンセルする手順については、「Amazon Application Recovery Controller (ARC) デベロッパーガイド」の「Canceling a zonal shift」を参照してください。

ゾーンシフトをキャンセルするには

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の [ロードバランシング] で [ロードバランサー] を選択します。
3. アクティブなゾーンシフトを持つ Network Load Balancer を選択します。
4. [統合] タブの [Amazon Application Recovery Controller (ARC)] で、[ゾーンシフトをキャンセル] を選択します。

   これにより ARC コンソールが開き、キャンセルプロセスが続行されます。
5. [Cancel zonal shift] (ゾーンシフトをキャンセル) を選択します。
6. 確認を求められたら、[確認] を選択します。

AWS CLI

ゾーンシフトをキャンセルするには

Amazon Application Recovery Controller (ARC) cancel-zonal-shift コマンドを使用します。

```
aws arc-zonal-shift cancel-zonal-shift \
    --zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf57EXAMPLE
```

Network Load Balancer のキャパシティ予約

ロードバランサーキャパシティユニット (LCU) 予約では、ロードバランサーの静的最小キャパシティを予約できます。Network Load Balancer は、検出されたワークロードをサポートし、容量のニーズを満たすように自動的にスケーリングします。最小容量を設定すると、ロードバランサーは受信したトラフィックに基づいてスケールアップまたはスケールダウンを続けますが、設定された最小容量を下回ることも防止します。

次の状況では LCU 予約の使用を検討してください。

- 突然の異常な高トラフィックが発生し、イベント中にロードバランサーが突然のトラフィックの急増をサポートできるようにしたいイベントが近づいている。
- ワークロードの性質上、短期間、予期しないスパイクトラフィックが発生している。
- ロードバランサーを設定して、特定の開始時刻にサービスをオンボードまたは移行し、自動スケーリングが有効になるまで待機するのではなく、大容量から開始する必要がある。
- ロードバランサー間でワークロードを移行していて、ソースのスケールに合わせて送信先を設定する場合。

必要なキャパシティの見積もり

ロードバランサー用に予約する容量を決定するときは、負荷テストを実行するか、予想される今後のトラフィックを表すワークロードの履歴データを確認することをお勧めします。Elastic Load Balancing コンソールを使用すると、レビューされたトラフィックに基づいて、予約する必要があるキャパシティを見積もることができます。

または、CloudWatch メトリクス ProcessedBytes を参照して、適切なキャパシティレベルを決定することもできます。ロードバランサーのキャパシティは LCU で予約され、各 LCU は 2.2Mbps に等しくなります。最大 (ProcessedBytes) メトリクスを使用してロードバランサーの 1 分あたりのスループットトラフィックの最大数を表示し、そのスループットを 2.2Mbps の変換レートを使用して LCU に変換すると 1 LCU になります。

参照する履歴ワークロードデータがなく、負荷テストを実行できない場合は、LCU 予約計算ツールを使用して必要なキャパシティを見積もることができます。LCU 予約計算ツールは、 AWS 観測された過去のワークロードに基づいてデータを使用し、特定のワークロードを表していない場合があり

ます。詳細については、「ロードバランサーキャパシティユニット予約計算ツール」を参照してください。

サポート対象のリージョン

この機能は次のリージョンでご利用いただけます。

- 米国東部 (バージニア北部)
- 米国東部 (オハイオ)
- 米国西部 (オレゴン)
- アジアパシフィック (香港)
- アジアパシフィック (シンガポール)
- アジアパシフィック (シドニー)
- アジアパシフィック (東京)
- 欧州 (フランクフルト)
- 欧州 (アイルランド)
- 欧州 (ストックホルム)

LCU 予約の最小値と最大値

予約リクエストの合計は、アベイラビリティーゾーンあたり 2,750 LCU 以上である必要があります。最大値は、アカウントのクォータによって決まります。詳細については、「the section called “ロードバランサーキャパシティユニット”」を参照してください。

Network Load Balancer のロードバランサーキャパシティユニットの予約をリクエストする

LCU 予約を使用する前に、次を確認してください。

- LCU 予約は、TLS リスナーを使用する Network Load Balancer ではサポートされていません。
- LCU 予約は、Network Load Balancer のスループットキャパシティの予約のみをサポートします。LCU 予約をリクエストするときは、1 LCU の変換レートを使用して容量のニーズを Mbps から LCU に変換します。
- キャパシティはリージョンレベルで予約され、アベイラビリティーゾーン間で均等に分散されます。LCU 予約を有効にする前に、各アベイラビリティーゾーンに十分に均等に分散されたターゲットがあることを確認します。

- LCU 予約リクエストは先着順で受理され、その時点でゾーンで使用可能なキャパシティによって異なります。ほとんどのリクエストは通常 1 時間以内に処理されますが、最大数時間かかる場合があります。
- 既存の予約を更新するには、前のリクエストをプロビジョニングするか、失敗する必要があります。リザーブドキャパシティは必要な回数だけ増やすことができますが、リザーブドキャパシティは 1 日に 2 回しか減らせません。
- リザーブドキャパシティまたはプロビジョニングされたキャパシティは、終了またはキャンセルされるまで引き続き料金が発生します。

Console

LCU 予約をリクエストするには

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、[ロードバランサー] を選択します。
3. ロードバランサー名を選択します。
4. [キャパシティ] タブで、[LCU 予約を編集] を選択します。
5. [履歴参照ベースの見積り] を選択します。
6. 推奨の予約済み LCU レベルを表示するには、参照期間を選択します。
7. 過去のリファレンスワークロードがない場合は、[手動見積り] を選択し、予約する LCU の数を入力できます。
8. [保存] を選択します。

AWS CLI

LCU 予約をリクエストするには

modify-capacity-reservation コマンドを使用します。

```
aws elbv2 modify-capacity-reservation \
    --load-balancer-arn load-balancer-arn \
    --minimum-load-balancer-capacity CapacityUnits=3000
```

CloudFormation

LCU 予約をリクエストするには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースを更新します。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      MinimumLoadBalancerCapacity:
        CapacityUnits: 3000
```

Network Load Balancer のロードバランサーキャパシティユニット予約の更新またはキャンセル

ロードバランサーのトラフィックパターンが変更された場合は、ロードバランサーの LCU 予約を更新またはキャンセルできます。

Console

LCU 予約を更新またはキャンセルするには

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、[ロードバランサー] を選択します。
3. ロードバランサー名を選択します。
4. [キャパシティ] タブで、次のいずれかを実行します。
    a. LCU 予約を更新するには、[LCU 予約を編集] を選択します。
    b. LCU 予約をキャンセルするには、[キャパシティをキャンセル] を選択します。

AWS CLI

LCU 予約をキャンセルするには

modify-capacity-reservation コマンドを使用します。

```
aws elbv2 modify-capacity-reservation \
    --load-balancer-arn load-balancer-arn \
    --reset-capacity-reservation
```

Network Load Balancer のロードバランサーキャパシティユニット予約のモニタリング

予約ステータス

ステータスに表示される可能性のあるエラー値は、次のとおりです。

- `pending` – プロビジョニング中の予約を示します。
- `provisioned` – リザーブドキャパシティが使用可能であることを示します。
- `failed` – その時点でリクエストを完了できないことを示します。
- `rebalancing` – アベイラビリティーゾーンが追加または削除され、ロードバランサーがキャパシティを再調整していることを示します。

LCU 使用率

リザーブド LCU 使用率を確認するには、1 分あたりの `ProcessedBytes` メトリクスを 1 時間あたりの `Sum(ReservedLCUs)` と比較します。1 分あたりのバイト数を 1 時間あたりの LCU に変換するには、(1 分あたりのバイト数)*8/60/ (10^6)/2.2 を使用します。

Console

LCU 予約のステータスを表示するには

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、[ロードバランサー] を選択します。
3. ロードバランサー名を選択します。
4. [キャパシティ] タブで、[予約ステータス] と [リザーブド LCU] 値を表示できます。

AWS CLI

LCU 予約のステータスをモニタリングするには

describe-capacity-reservation コマンドを使用します。

```
aws elbv2 describe-capacity-reservation \
    --load-balancer-arn load-balancer-arn
```

Network Load Balancer のリスナー

リスナーとは、設定したプロトコルとポートを使用して接続リクエストをチェックするプロセスです。Network Load Balancer の使用を開始する前に、1 つ以上のリスナーを追加する必要があります。ロードバランサーにリスナーがない場合、クライアントからのトラフィックを受信できません。リスナーに対して定義したルールにより、EC2 インスタンスなど、登録するターゲットにロードバランサーがリクエストをルーティングする方法が決まります。

内容

リスナーの設定

リスナーは次のポートとプロトコルをサポートします。

- プロトコル: TCP、TLS、UDP、TCP_UDP、QUIC、TCP_QUIC
- ポート: 1 ～ 65535

アプリケーションがビジネスロジックに集中できるように、TLS リスナーを使用して、暗号化および復号の作業をロードバランサーに任せることができます。リスナープロトコルが TLS の場合は、リスナーに SSL サーバー証明書を少なくとも 1 つデプロイする必要があります。詳細については、「サーバー証明書」を参照してください。

ターゲットがロードバランサーではなく TLS トラフィックを復号化する必要がある場合は、TLS リスナーを作成する代わりに、ポート 443 に TCP リスナーを作成できます。TCP リスナーを使用すると、ロードバランサーは暗号化されたトラフィックを復号化せずにターゲットに渡します。

QUIC リスナーを使用して QUIC トラフィックを許可できます。Network Load Balancer は、RFC9000 に従ってパススルーロードバランサーとして機能します。QUIC リスナーと QUIC 対応バックエンドを使用して、モバイルデバイスのシームレスな接続移行を実現します。

同じポートで TCP と UDP の両方をサポートするには、TCP_UDP リスナーを作成します。TCP_UDP リスナーのターゲットグループは、TCP_UDP プロトコルを使用する必要があります。

同じポートで TCP と QUIC の両方をサポートするには、TCP_QUIC リスナーを作成します。TCP_QUIC リスナーのターゲットグループは、TCP_QUIC プロトコルを使用する必要があります。

デュアルスタックロードバランサーの UDP リスナーにはIPv6 ターゲットグループが必要です。

WebSockets は、TCP、TLS、TCP_UDP、TCP_QUIC リスナーでのみサポートされています。

QUIC トラフィックはバージョンネゴシエーションをサポートしていません。サポートされている QUIC バージョンは QUIC v1 のみです。

設定済みのリスナーに送信されるすべてのネットワークトラフィックが、意図されたトラフィックとして分類されます。設定済みのリスナーに一致しないネットワークトラフィックが、意図しないトラフィックとして分類されます。Type 3 以外の ICMP リクエストも、意図しないトラフィックとみなされます。Network Load Balancer は、意図しないトラフィックをターゲットに転送せずにドロップします。新しい接続またはアクティブな TCP 接続の一部ではない設定済みリスナーのリスナーポートに送信される TCP データパケットは、TCP リセット (RST) で拒否されます。

詳細については、Elastic Load Balancing ユーザーガイドのルーティングのリクエストを参照してください。

デフォルトアクション

リスナーを作成するときは、ルーティングリクエストのデフォルトアクションを指定します。デフォルトのアクションは、指定したターゲットグループにリクエストを転送します。

複数のターゲットグループにトラフィックを分散する

デフォルトアクションで複数のターゲットグループを指定した場合、リクエストは関連する重みに基づいて、これらのターゲットグループに分散されます。ターゲットグループごとに 0～999 の重みを指定する必要があります。重みが 0 のターゲットグループはトラフィックを受信しません。ターゲットグループを追加またはターゲットグループの重みを更新すると、新しい接続は新しいターゲットグループの重みに基づいてルーティングされます。既存の接続は影響を受けず、通常どおり終了するまで続行されます。

例えば、ターゲットグループを 2 つ指定し、それぞれ重みが 10 の場合、各ターゲットグループはリクエストの半分を受信します。2 つのターゲットグループ (1 つは重みが 10 で、もう 1 つは重みが 20) を指定すると、重みが 20 のターゲットグループは他のターゲットグループの 2 倍の数のリクエストを受信します。

一般的なユースケースは、あるターゲットグループから別のターゲットグループへのトラフィックの移行です。つまり、0 になるまで元のターゲットグループの重みを減らしながら、新しいターゲットグループの重みを徐々に増やします。ターゲットグループの重みを 0 に更新すると、しばらくしてから新しい接続は受信されなくなり、既存の接続は閉じられます。

スティッキーセッションと加重ターゲットグループ

リスナーの転送アクションでは、ターゲットグループの維持を有効化するかどうかを指定できます。有効化すると、ターゲットグループの維持により、同じ送信元 IP アドレスからの後続の接続で、以前選択したターゲットグループが優先されます。

考慮事項

- TLS リスナーの場合、TCP ターゲットグループと TLS ターゲットグループの両方をリスナールールに追加することはできません。すべてのターゲットグループは同じプロトコルを使用する必要があります。
- TLS リスナーの場合、ターゲットグループの維持はサポートされていません。
- デュアルスタックロードバランサーの場合、IPv4 ターゲットグループと IPv6 ターゲットグループの両方を同じデフォルトアクションに追加することはできません。デフォルトアクションのすべてのターゲットグループは、同じ IP アドレスタイプを使用する必要があります。
- リスナーの場合、転送アクションに複数のターゲットグループが含まれ、そのいずれかで維持が有効になっていれば、転送アクションでもターゲットグループの維持が有効化されている必要があります。

リスナー属性

Network Load Balancer のリスナー属性を以下に示します。

`tcp.idle_timeout.seconds`

TCP アイドルタイムアウト値 (秒単位)。有効な範囲は 60～6,000 秒です。デフォルト値は 350 秒です。

詳細については、「アイドルタイムアウトを更新する」を参照してください。

セキュアリスナー

TLS リスナーを使用するには、ロードバランサーにサーバー証明書を少なくとも 1 つデプロイする必要があります。ロードバランサーはサーバー証明書を使用してフロントエンド接続を終了してから、ターゲットにリクエストを送信する前に、クライアントからのリクエストを復号します。ロードバランサーが復号化せずに、暗号化されたトラフィックをターゲットに渡す必要がある場合は、TLS リスナーを作成するのではなく、ポート 443 で TCP リスナーを作成します。ロードバランサーは、リクエストを復号化せずにそのままの状態でターゲットに渡します。

Elastic Load Balancing は、セキュリティポリシーと呼ばれる TLS ネゴシエーション設定を使用して、クライアントとロードバランサー間の TLS 接続をネゴシエートします。セキュリティポリシーはプロトコルと暗号の組み合わせです。プロトコルは、クライアントとサーバーの間の安全な接続を確立し、クライアントとロードバランサーの間で受け渡しされるすべてのデータのプライバシーを保証します。暗号とは、暗号化キーを使用してコード化されたメッセージを作成する暗号化アルゴリズムです。プロトコルは、複数の暗号を使用し、インターネットを介してデータを暗号化します。接続ネゴシエーションのプロセスで、クライアントとロードバランサーでは、それぞれサポートされる暗号とプロトコルのリストが優先される順に表示されます。サーバーのリストで最初にクライアントの暗号と一致した暗号が安全な接続用に選択されます。

Network Load Balancer は、相互 TLS 認証 (mTLS) をサポートしていません。mTLS をサポートするには、TLS リスナーの代わりに TCP リスナーを作成します。ロードバランサーはリクエストをそのまま渡すため、ターゲットに mTL を実装できます。

Network Load Balancer は、TLS 1.3 の PSK と TLS 1.2 以前のセッションチケットを使用した TLS の再開をサポートします。セッション ID を使用した再開、または SNI を使用してリスナーで複数の証明書が設定されている場合はサポートされていません。0-RTT データ機能と early_data 拡張機能は実装されていません。

関連するデモについては、Network Load Balancer での TLS サポートおよび Network Load Balancer での SNI サポートを参照してください。

ALPN ポリシー

Application-Layer Protocol Negotiation (ALPN) は、初期 TLS ハンドシェイク hello メッセージで送信される TLS 拡張機能です。ALPN を使用すると、アプリケーションレイヤーは HTTP/1 や HTTP/2 などのセキュアな接続上で使用するプロトコルをネゴシエートできます。

クライアントが ALPN 接続を開始すると、ロードバランサーはクライアントの ALPN 設定リストを ALPN ポリシーと比較します。クライアントが ALPN ポリシーからのプロトコルをサポートしている場合、ロードバランサーは ALPN ポリシーの設定リストに基づいて接続を確立します。それ以外の場合、ロードバランサーは ALPN を使用しません。

サポートされている ALPN ポリシー

サポートされている ALPN ポリシーは次のとおりです。

`HTTP1Only`

HTTP/1.* のみをネゴシエートします。ALPN 設定リストは http/1.1、http/1.0 です。

`HTTP2Only`

HTTP/2 のみをネゴシエートします。ALPN 設定リストは h2 です。

`HTTP2Optional`

HTTP/2 よりも HTTP/1.* を優先します (これは HTTP/2 テストに役立ちます)。ALPN 設定リストは http/1.1、http/1.0、h2 です。

`HTTP2Preferred`

HTTP/1.* よりも HTTP/2 を優先します。ALPN 設定リストは、h2、http/1.1、http/1.0 です。

`None`

ALPN をネゴシエートしないでください。これがデフォルト値です。

ALPN 接続を有効にする

TLS リスナーを作成または変更するときに、ALPN 接続を有効にできます。詳細については、「リスナーの追加」および「ALPN ポリシーを更新するには」を参照してください。

Network Load Balancer のリスナーを作成する

リスナーとは接続リクエストをチェックするプロセスです。ロードバランサーを作成するときにリスナーを定義し、いつでもロードバランサーにリスナーを追加できます。

前提条件

- デフォルトアクションのターゲットグループを指定する必要があります。詳細については、「Network Load Balancer のターゲットグループを作成する」を参照してください。
- TLS リスナーの SSL 証明書を指定する必要があります。ターゲットにリクエストをルーティングする前に、ロードバランサーはこの証明書を使用して接続を終了し、クライアントからのリクエストを復号します。詳細については、「Network Load Balancer のサーバー証明書」を参照してください。
- `dualstack` ロードバランサーの UDP リスナーで IPv4 ターゲットグループを使用することはできません。
- QUIC および TCP_QUIC リスナーは、関連するセキュリティグループを使用する `dualstack` ロードバランサーまたはロードバランサーでは許可されません。
- QUIC および TCP_QUIC リスナーは、関連するセキュリティグループを使用するロードバランサーでは許可されません。
- Network Load Balancer では、常に 1 つの QUIC または TCP_QUIC リスナーのみが許可されます。
- QUIC および TCP_QUIC リスナーは、UDP または TCP_UDP リスナーを使用する Network Load Balancer では許可されません。

リスナーの追加

クライアントからロードバランサーへの接続用のプロトコルとポート、およびデフォルトのリスナールールのターゲットグループでリスナーを設定します。詳細については、「リスナーの設定」を参照してください。

Console

リスナーを追加するには

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、[ロードバランサー] を選択します。

3. ロードバランサーの名前を選択して、その詳細ページを開きます。
4. [Listeners] (リスナー) タブで、[Add listener] (リスナーの追加) を選択します。
5. [プロトコル] で、TCP、UDP、TCP_UDP、TLS、QUIC、または TCP_QUIC を選択します。デフォルトポートのままにすることも、別のポートを入力することもできます。
6. [Default action] (デフォルトアクション) では、トラフィックを転送するターゲットグループを選択します。

   別のターゲットグループを追加するには、[ターゲットグループを追加] を選択し、必要に応じて重みを更新します。

   ニーズに合ったターゲットグループがない場合は、[ターゲットグループを作成] を選択して今すぐ作成します。詳細については、「ターゲットグループの作成」を参照してください。
7. [TLS リスナー] [Security policy (セキュリティポリシー)] で、デフォルトのセキュリティポリシーを保持することをお勧めします。
8. [TLS リスナー] デフォルトの SSL/TLS サーバー証明書の場合は、デフォルトの証明書を選択します。証明書は、次のいずれかのソースから選択できます。
   - を使用して証明書を作成またはインポートした場合は AWS Certificate Manager、ACM から を選択し、証明書 (ACM から) から証明書を選択します。
   - IAM を使用して証明書をインポートした場合は、[IAM から] を選択し、[証明書 (IAM から)] から証明書を選択します。
   - 証明書がある場合は、[証明書をインポート] を選択します。[ACM にインポート] または [IAM にインポート] を選択します。[証明書のプライベートキー] では、PEM エンコードされたプライベートキーファイルの内容をコピーして貼り付けます。[証明書本文] では、PEM エンコードされたパブリックキー証明書ファイルの内容をコピーして貼り付けます。自己署名証明書を使用しておらず、ブラウザが暗黙的に証明書を受け入れることが重要である場合に限り、[証明書チェーン] に、PEM エンコードされた証明書チェーンファイルの内容をコピーして貼り付けます。
9. [TLS リスナー] [ALPN ポリシー] で、ALPN を有効にするポリシーを選択するか、[なし] を選択して ALPN を無効にします。詳細については、「ALPN ポリシー」を参照してください。
10. (オプション) タグを追加するには、[リスナータグ] を展開します。[新しいタグを追加] を選択し、タグのキーと値を入力します。
11. [Add] (追加) を選択します。
12. [TLS リスナー] オプションの証明書リストに証明書を追加するには、「証明書リストに証明書を追加する」を参照してください。

AWS CLI

対象グループを作成するには

デフォルトのアクションに使用できるターゲットグループがない場合は、create-target-group コマンドを使用して今すぐ作成します。例については「ターゲットグループの作成」を参照してください。

TCP リスナーを追加するには

create-listener コマンドを使用して、TCP プロトコルを指定します。

```
aws elbv2 create-listener \
    --load-balancer-arn load-balancer-arn \
    --protocol TCP \
    --port 80 \
    --default-actions Type=forward,TargetGroupArn=target-group-arn
```

複数のターゲットグループを持つ TCP リスナーを追加するには

create-listener コマンドを使用して、TCP プロトコル、ターゲットグループ、重みを指定します。

```
aws elbv2 create-listener \
    --load-balancer-arn load-balancer-arn \
    --protocol TCP \
    --port 80 \
    --default-actions '[{
        "Type":"forward",
        "ForwardConfig":{
            "TargetGroups":[
                {"TargetGroupArn":"target-group-1-arn","Weight":10},
                {"TargetGroupArn":"target-group-2-arn","Weight":30}
            ]
        }
    }]'
```

TLS リスナーを追加するには

TLS プロトコルを指定する create-listener コマンドを使用します。

```
aws elbv2 create-listener \
```

```
    --load-balancer-arn load-balancer-arn \
    --protocol TLS \
    --port 443 \
    --certificates CertificateArn=certificate-arn \
    --ssl-policy ELBSecurityPolicy-TLS13-1-2-Res-2021-06 \
    --default-actions Type=forward,TargetGroupArn=target-group-arn
```

UDP リスナーを追加するには

UDP プロトコルを指定する create-listener コマンドを使用します。

```
aws elbv2 create-listener \
    --load-balancer-arn load-balancer-arn \
    --protocol UDP \
    --port 53 \
    --default-actions Type=forward,TargetGroupArn=target-group-arn
```

QUIC リスナーを追加するには

QUIC プロトコルを指定する create-listener コマンドを使用します。

```
aws elbv2 create-listener \
    --load-balancer-arn load-balancer-arn \
    --protocol QUIC \
    --port 443 \
    --default-actions Type=forward,TargetGroupArn=target-group-arn
```

CloudFormation

TCP リスナーを追加するには

TCP プロトコルを使用して、AWS::ElasticLoadBalancingV2::Listener タイプのリソースを定義します。

```
Resources:
  myTCPListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: TCP
      Port: 80
```

```
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
```

複数のターゲットグループを持つ TCP リスナーを追加するには

TCP プロトコルを使用して、AWS::ElasticLoadBalancingV2::Listener タイプのリソースを定義します。

```
Resources:
    myTCPListener:
        Type: 'AWS::ElasticLoadBalancingV2::Listener'
        Properties:
        LoadBalancerArn: !Ref myLoadBalancer
        Protocol: TCP
        Port: 80
        DefaultActions:
            - Type: forward
            ForwardConfig:
                TargetGroups:
                - TargetGroupArn: !Ref myTargetGroup1,
                    Weight: 10
                - TargetGroupArn: !Ref myTargetGroup2,
                    Weight: 30
            TargetGroupStickinessConfig:
                Enabled: true
```

TLS リスナーを追加するには

TLS プロトコルを使用して、AWS::ElasticLoadBalancingV2::Listener タイプのリソースを定義します。

```
Resources:
  myTLSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: TLS
      Port: 443
      SslPolicy: "ELBSecurityPolicy-TLS13-1-2-Res-2021-06"
      Certificates:
        - CertificateArn: "certificate-arn"
```

```
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
```

UDP リスナーを追加するには

UDP プロトコルを使用して、AWS::ElasticLoadBalancingV2::Listener タイプのリソースを定義します。

```
Resources:
  myUDPListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: UDP
      Port: 53
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
```

QUIC リスナーを追加するには

QUIC プロトコルを使用して、AWS::ElasticLoadBalancingV2::Listener タイプのリソースを定義します。

```
Resources:
  myQUICListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: QUIC
      Port: 443
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
```

Network Load Balancer のサーバー証明書

Network Load Balancer のセキュアリスナーを作成するときは、少なくとも 1 つの証明書をロードバランサーにデプロイする必要があります。ロードバランサーには X.509 証明書 (サーバー証明書) が

必要です。証明書とは、認証局 (CA) によって発行された識別用デジタル形式です。証明書には、認識用情報、有効期間、パブリックキー、シリアル番号と発行者のデジタル署名が含まれます。

ロードバランサーで使用する証明書を作成するときに、ドメイン名を指定する必要があります。TLS 接続を検証できるように、証明書のドメイン名は、カスタムドメイン名レコードと一致する必要があります。一致しない場合、トラフィックは暗号化されません。

`www.example.com` などの証明書の完全修飾ドメイン名 (FQDN) または `example.com` などの apex ドメイン名を指定する必要があります。また、同じドメインで複数のサイト名を保護するために、アスタリスク (*) をワイルドカードとして使用できます。ワイルドカード証明書をリクエストする場合、アスタリスク (*) はドメイン名の一番左の位置に付ける必要があり、1 つのサブドメインレベルのみを保護できます。例えば、`*.example.com` は `corp.example.com`、`images.example.com` を保護しますが、`test.login.example.com` を保護することはできません。また、`*.example.com` は、`example.com` のサブドメインのみを保護し、ネイキッドドメインまたは apex ドメイン (`example.com`) は保護しないことに注意してください。ワイルドカード名は、証明書の [サブジェクト] フィールドと [サブジェクト代替名] 拡張子に表示されます。公開証明書の詳細については、「AWS Certificate Manager ユーザーガイド」の「公開証明書のリクエスト」を参照してください。

AWS Certificate Manager (ACM) を使用して、ロードバランサーの証明書を作成することをお勧めします。ACM は Elastic Load Balancing と統合して、ロードバランサーに証明書をデプロイできます。詳細については、「AWS Certificate Manager ユーザーガイド」を参照してください。

または、TLS ツールを使用して証明書署名リクエスト (CSR) を作成し、CA によって署名された CSR を取得して証明書を生成し、証明書を ACM にインポートするか、証明書を AWS Identity and Access Management (IAM) にアップロードすることもできます。詳細については、「AWS Certificate Manager ユーザーガイド」の「証明書のインポート」または「IAM ユーザーガイド」の「サーバー証明書の使用」を参照してください 。

サポートされているキーアルゴリズム

- RSA 1024 ビット
- RSA 2048 ビット
- RSA 3072 ビット
- ECDSA 256 ビット
- ECDSA 384 ビット
- ECDSA 521 ビット

デフォルトの証明書

TLS リスナーを作成するには、1 つ以上の証明書を指定する必要があります。この証明書は、default certificate として知られています。TLS リスナーを作成した後、デフォルトの証明書を置き換えることができます。詳細については、「デフォルトの証明書の置き換え」を参照してください。

証明書のリスト内の追加の証明書を指定する場合、クライアントがホスト名を指定するために Server Name Indication (SNI) プロトコルを使用せずに接続した場合、または証明書リストに一致する証明書がない場合にのみデフォルトの証明書が使用されます。

追加の証明書を指定せずに単一のロードバランサーを介して複数の安全なアプリケーションをホストする必要がある場合は、ワイルドカード証明書を使用するか、または追加ドメインごとにサブジェクト代替名 (SAN) を証明書に追加できます。

証明書リスト

TLS リスナーを作成すると、デフォルトの証明書と空の証明書リストが作成されます。リスナーの証明書リストに証明書を追加することもできます。証明書リストを使用すると、ロードバランサーは同じポートで複数のドメインをサポートし、ドメインごとに異なる証明書を提供できます。詳細については、「証明書リストに証明書を追加する」を参照してください。

ロードバランサーは、SNI をサポートするスマート証明書の選択アルゴリズムを使用します。クライアントから提供されたホスト名が証明書リスト内の単一の証明書と一致する場合、ロードバランサーはこの証明書を選択します。クライアントが提供するホスト名が証明書リストの複数の証明書と一致する場合、ロードバランサーはクライアントがサポートできる最適な証明書を選択します。証明書の選択は、次の条件と順序に基づいて行われます。

- パブリックキーアルゴリズム (RSA よりも ECDSA が優先)
- ハッシュアルゴリズム (MD5 よりも SHA が優先)
- キーの長さ (最大が優先)
- 有効期間

ロードバランサーアクセスログエントリは、クライアントが指定したホスト名とクライアントが提出する証明書を示します。詳細については、「アクセスログのエントリ」を参照してください。

証明書の更新

各証明書には有効期間が記載されています。有効期間が終了する前に、必ずロードバランサーの各証明書を更新するか、置き換える必要があります。これには、デフォルトの証明書と証明書リスト内の

証明書が含まれます。証明書を更新または置き換えしても、ロードバランサーノードが受信し、正常なターゲットへのルーティングを保留中の未処理のリクエストには影響しません。証明書更新後、新しいリクエストは更新された証明書を使用します。証明書置き換え後、新しいリクエストは新しい証明書を使用します。

証明書の更新と置き換えは次のとおりに管理できます。

- によって提供され AWS Certificate Manager 、ロードバランサーにデプロイされた証明書は、自動的に更新できます。ACM は、期限切れになる前に証明書の更新を試みます。詳細については、AWS Certificate Manager ユーザーガイドの 管理された更新 を参照してください。
- 証明書を ACM にインポートした場合は、証明書の有効期限をモニタリングし、期限切れ前に更新する必要があります。詳細については、AWS Certificate Manager ユーザーガイドの 証明書のインポート を参照してください。
- IAM に証明書をインポートする場合、新しい証明書を作成し、この新しい証明書を ACM あるいは IAM にインポートします。ロードバランサーにこの新しい証明書を追加し、期限切れの証明書をロードバランサーから削除します。

Network Load Balancer のセキュリティポリシー

TLS リスナーを作成するときは、セキュリティポリシーを選択する必要があります。セキュリティポリシーによって、ロードバランサーとクライアント間の SSL ネゴシエーションでサポートされる暗号とプロトコルが決まります。要件が変化した場合や、新しいセキュリティポリシーがリリースされた場合は、ロードバランサーのセキュリティポリシーを更新できます。詳細については、「セキュリティポリシーの更新」を参照してください。

考慮事項

- TLS リスナーにはセキュリティポリシーが必要です。リスナーの作成時にセキュリティポリシーを指定しない場合、デフォルトのセキュリティポリシーが使用されます。デフォルトのセキュリティポリシーは、TLS リスナーの作成方法によって異なります。
  - コンソール – デフォルトセキュリティポリシーは `ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09` です。
  - その他の方法 (、 AWS CLI AWS CloudFormation、 など AWS CDK) – デフォルトのセキュリティポリシーは です`ELBSecurityPolicy-2016-08`。
- 名前に PQ を含むセキュリティポリシーは、ハイブリッドポスト量子キー交換を提供します。互換性のために、従来の ML-KEM キー交換アルゴリズムとポスト量子 ML-KEM キー交換アルゴリズムの両方をサポートしています。クライアントは、キー交換にハイブリッドポスト量子 TLS を使用

するには、ML-KEM キー交換をサポートする必要があります。ハイブリッドポスト量子ポリシーは、SecP256r1MLKEM768, SecP384r1MLKEM1024、および X25519MLKEM768 アルゴリズムをサポートしています。詳細については、「ポスト量子暗号化」を参照してください。

- AWS では、新しいポスト量子 TLS (PQ-TLS) ベースのセキュリティポリシー`ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09`または を実装することをお勧めします`ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09`。このポリシーは、ハイブリッド PQ-TLS、TLS 1.3 のみ、または TLS 1.2 のみをネゴシエートできるクライアントをサポートすることで下位互換性を確保し、ポスト量子暗号化への移行中のサービスの中断を最小限に抑えます。クライアントアプリケーションがキー交換オペレーションのために PQ-TLS をネゴシエートする機能を開発するにつれて、より制限の厳しいセキュリティポリシーに段階的に移行できます。
- Network Load Balancer に送信される TLS リクエストに関するアクセスログを有効にすると、TLS トラフィックパターンの分析、セキュリティポリシーのアップグレードの管理、問題のトラブルシューティングを行うことができます。ロードバランサーのアクセスログを有効にし、対応するアクセスログエントリを調べます。詳細については、「アクセスログ」および「Network Load Balancer のクエリ例」を参照してください。
- ロードバランサーへのアクセスリクエストの TLS プロトコルバージョン (ログフィールド位置 5) とキー交換 (ログフィールド位置 13) を表示するには、アクセスログを有効にし、対応するログエントリを調べます。詳細については、「アクセスログ」を参照してください。
- IAM およびサービスコントロールポリシー (SCPs) でそれぞれ Elastic Load Balancing 条件キーを使用することで、 AWS アカウント および AWS Organizations 全体のユーザーが利用できるセキュリティポリシーを制限できます。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 (SCP)」を参照してください。
- TLS 1.3 のみをサポートするポリシーは、Forward Secrecy (FS) をサポートしています。TLS_* および ECDHE_* 形式の暗号のみを持つ TLS 1.3 および TLS 1.2 をサポートするポリシーも FS を提供します。
- Network Load Balancer は、TLS 1.2 の拡張マスターシークレット (EMS) 拡張機能をサポートしています。

バックエンド接続

フロントエンド接続に使用するセキュリティポリシーは選択できますが、バックエンド接続に使用するセキュリティポリシーは選択できません。バックエンド接続のセキュリティポリシーは、リスナーのセキュリティポリシーによって異なります。リスナーが を使用している場合:

- FIPS ポスト量子 TLS ポリシー - バックエンド接続の使用 `ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09`

- FIPS ポリシー - バックエンド接続の使用 `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04`
- ポスト量子 TLS ポリシー - バックエンド接続の使用 `ELBSecurityPolicy-TLS13-1-0-PQ-2025-09`
- TLS 1.3 ポリシー - バックエンド接続の使用 `ELBSecurityPolicy-TLS13-1-0-2021-06`
- 他のすべての TLS ポリシーのバックエンド接続では、 `ELBSecurityPolicy-2016-08`

プロトコルと暗号は describe-ssl-policies AWS CLI コマンドを使用して記述できます。または以下の表を参照してください。

セキュリティポリシー

TLS セキュリティポリシー

TLS セキュリティポリシーを使用すると、TLS プロトコルの特定のバージョンを無効にしてコンプライアンスおよびセキュリティ標準を満たす、または廃止済みの暗号を必要とするレガシークライアントをサポートすることができます。

TLS 1.3 のみをサポートするポリシーは、Forward Secrecy (FS) をサポートしています。TLS_* および ECDHE_* 形式の暗号のみを持つ TLS 1.3 および TLS 1.2 をサポートするポリシーも FS を提供します。

内容

- ポリシー別のプロトコル
- ポリシー別の暗号
- 暗号別のポリシー

ポリシー別のプロトコル

以下は、各 TLS セキュリティポリシーがサポートしているプロトコルの一覧です。

セキュリティポリシー	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-TLS13-1-3-2021-06	はい	いいえ	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-3-PQ-2025-09	はい	いいえ	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-2021-06	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-PQ-2025-09	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Res-2021-06	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06	はい	はい	いいえ	いいえ

セキュリティポリシー	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-1-2021-06	はい	はい	はい	いいえ
ELBSecurityPolicy-TLS13-1-0-2021-06	はい	はい	はい	はい
ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	はい	はい	はい	はい
ELBSecurityPolicy-TLS-1-2-Ext-2018-06	いいえ	はい	いいえ	いいえ
ELBSecurityPolicy-TLS-1-2-2017-01	いいえ	はい	いいえ	いいえ
ELBSecurityPolicy-TLS-1-1-2017-01	いいえ	はい	はい	いいえ
ELBSecurityPolicy-2016-08	いいえ	はい	はい	はい
ELBSecurityPolicy-2015-05	いいえ	はい	はい	はい

ポリシー別の暗号

以下は、各 TLS セキュリティポリシーがサポートしている暗号の一覧です。

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS13-1-3-2021-06 ELBSecurityPolicy-TLS13-1-3-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384

セキュリティポリシー	暗号
	• TLS_CHACHA20_POLY1305_SHA256
ELBSecurityPolicy-TLS13-1-2-2021-06 ELBSecurityPolicy-TLS13-1-2-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384
ELBSecurityPolicy-TLS13-1-2-Res-2021-06 ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • AES128-GCM-SHA256 • AES128-SHA256 • AES256-GCM-SHA384 • AES256-SHA256

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS13-1-1-2021-06	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS13-1-0-2021-06 ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • TLS_CHACHA20_POLY1305_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS-1-2-Ext-2018-06	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS-1-2-2017-01	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • AES128-GCM-SHA256 • AES128-SHA256 • AES256-GCM-SHA384 • AES256-SHA256

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS-1-1-2017-01	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

セキュリティポリシー	暗号
ELBSecurityPolicy-2016-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

セキュリティポリシー	暗号
ELBSecurityPolicy-2015-05	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-ECDSA-AES256-SHA • ECDHE-RSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

暗号別のポリシー

以下は、各暗号をサポートしている TLS セキュリティポリシーの一覧です。

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – TLS_AES_128_GCM_SHA256 IANA – TLS_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-3-2021-06 • ELBSecurityPolicy-TLS13-1-3-PQ-2025-09	1301

暗号名	セキュリティポリシー	暗号スイート
	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – TLS_AES_256_GCM_SHA384 IANA – TLS_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-3-2021-06 • ELBSecurityPolicy-TLS13-1-3-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	1302

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – TLS_CHACHA20_POLY1305_SHA256 IANA – TLS_CHACHA20_POLY1305_SHA256	• ELBSecurityPolicy-TLS13-1-3-2021-06 • ELBSecurityPolicy-TLS13-1-3-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09	1303

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c02b

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-RSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c02f

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES128-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c023

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-RSA-AES128-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c027

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES128-SHA IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c009
OpenSSL – ECDHE-RSA-AES128-SHA IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c013

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c02c

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-RSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-2021-06 • ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c030

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES256-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c024

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-RSA-AES256-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-2021-06 • ELBSecurityPolicy-TLS13-1-2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c028

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES256-SHA IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c00a
OpenSSL – ECDHE-RSA-AES256-SHA IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	c014

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – AES128-GCM-SHA256 IANA – TLS_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	9c

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – AES128-SHA256 IANA – TLS_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	3c

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – AES128-SHA IANA – TLS_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	2f

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – AES256-GCM-SHA384 IANA – TLS_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	9d

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – AES256-SHA256 IANA – TLS_RSA_WITH_AES_256_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext1-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-2-2017-01 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	3d

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – AES256-SHA IANA – TLS_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06 • ELBSecurityPolicy-TLS13-1-2-Ext2-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-2021-06 • ELBSecurityPolicy-TLS13-1-0-2021-06 • ELBSecurityPolicy-TLS13-1-0-PQ-2025-09 • ELBSecurityPolicy-TLS-1-2-Ext-2018-06 • ELBSecurityPolicy-TLS-1-1-2017-01 • ELBSecurityPolicy-2016-08	35

FIPS セキュリティポリシー

連邦情報処理規格 (Federal Information Processing Standards/FIPS) は、機密情報を保護する暗号モジュールのセキュリティ要件を規定する米国政府とカナダ政府のセキュリティ基準です。詳細については、「AWS クラウドセキュリティコンプライアンス」ページの「[連邦情報処理規格 (FIPS) 140](#)」を参照してください。

FIPS ポリシーはすべて AWS-LC FIPS で検証済みの暗号化モジュールを利用しています。詳細については、サイト「NIST Cryptographic Module Validation Program」の「[AWS-LC Cryptographic Module](#)」のページを参照してください。

> **Important**
>
> ポリシー `ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04` と `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04` はレガシー互換性のためにのみ提供されています。これらは FIPS140 モジュールを使って FIPS 暗号化を使用しますが、TLS 設定に関する最新の NIST ガイダンスに準拠していない場合があります。

内容

- ポリシー別のプロトコル
- ポリシー別の暗号
- 暗号別のポリシー

ポリシー別のプロトコル

以下は、各 FIPS セキュリティポリシーがサポートしているプロトコルの一覧です。

セキュリティポリシー	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04	はい	いいえ	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09	はい	いいえ	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04	はい	はい	いいえ	いいえ

セキュリティポリシー	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09	はい	はい	いいえ	いいえ
ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04	はい	はい	はい	いいえ
ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04	はい	はい	はい	はい
ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	はい	はい	はい	はい

ポリシー別の暗号

以下は、各 FIPS セキュリティポリシーがサポートしている暗号の一覧です。

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384
ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256

セキュリティポリシー	暗号
	• ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384
ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • AES128-GCM-SHA256 • AES128-SHA256 • AES256-GCM-SHA384 • AES256-SHA256
ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

セキュリティポリシー	暗号
ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	• TLS_AES_128_GCM_SHA256 • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA • AES128-GCM-SHA256 • AES128-SHA256 • AES128-SHA • AES256-GCM-SHA384 • AES256-SHA256 • AES256-SHA

暗号別のポリシー

以下は、各暗号をサポートしている FIPS セキュリティポリシーの一覧です。

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – TLS_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04	1301

暗号名	セキュリティポリシー	暗号スイート
IANA – TLS_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04	

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – TLS_AES_256_GCM_SHA384 IANA – TLS_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-3-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	1302

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c02b

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-RSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c02f

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES128-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04	c023

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-RSA-AES128-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c027

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES128-SHA IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c009
OpenSSL – ECDHE-RSA-AES128-SHA IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c013

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c02c

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-RSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Res-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c030

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES256-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c024

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-RSA-AES256-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c028

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES256-SHA IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	c00a
OpenSSL – ECDHE-RSA-AES256-SHA IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04	c014

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – AES128-GCM-SHA256 IANA – TLS_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	9c
OpenSSL – AES128-SHA256 IANA – TLS_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	3c

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – AES128-SHA IANA – TLS_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	2f
OpenSSL – AES256-GCM-SHA384 IANA – TLS_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	9d

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – AES256-SHA256 IANA – TLS_RSA_WITH_AES_256_CBC_SHA256	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	3d
OpenSSL – AES256-SHA IANA – TLS_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-PQ-2025-09 • ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 • ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09	35

FS がサポートするセキュリティポリシー

FS (Forward Secrecy) がサポートするセキュリティポリシーは、一意のランダムセッションキーを使用して、暗号化されたデータの盗聴に対する追加の保護を提供します。これにより、シークレットの長期キーが侵害された場合でも、キャプチャされたデータのデコードを阻止できます。

このセクションのポリシーは FS をサポートしており、名前には「FS」が含まれています。ただし、これらは FS をサポートする唯一のポリシーではありません。TLS 1.3 のみをサポートするポリシーは FS をサポートします。TLS_* および ECDHE_* 形式の暗号のみを持つ TLS 1.3 および TLS 1.2 をサポートするポリシーも FS を提供します。

内容

ポリシー別のプロトコル

以下は、FS がサポートする各セキュリティポリシーがサポートしている、プロトコルの一覧です。

セキュリティポリシー	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityPolicy-FS-1-2-Res-2020-10	いいえ	はい	いいえ	いいえ
ELBSecurityPolicy-FS-1-2-Res-2019-08	いいえ	はい	いいえ	いいえ
ELBSecurityPolicy-FS-1-2-2019-08	いいえ	はい	いいえ	いいえ
ELBSecurityPolicy-FS-1-1-2019-08	いいえ	はい	はい	いいえ
ELBSecurityPolicy-FS-2018-06	いいえ	はい	はい	はい

ポリシー別の暗号

以下は、FS がサポートする各セキュリティポリシーがサポートしている、暗号の一覧です。

セキュリティポリシー	暗号
ELBSecurityPolicy-FS-1-2-Res-2020-10	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384
ELBSecurityPolicy-FS-1-2-Res-2019-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384
ELBSecurityPolicy-FS-1-2-2019-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA
ELBSecurityPolicy-FS-1-1-2019-08	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256

セキュリティポリシー	暗号
	• ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA
ELBSecurityPolicy-FS-2018-06	• ECDHE-ECDSA-AES128-GCM-SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • ECDHE-ECDSA-AES128-SHA256 • ECDHE-RSA-AES128-SHA256 • ECDHE-ECDSA-AES128-SHA • ECDHE-RSA-AES128-SHA • ECDHE-ECDSA-AES256-GCM-SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • ECDHE-ECDSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA384 • ECDHE-RSA-AES256-SHA • ECDHE-ECDSA-AES256-SHA

暗号別のポリシー

以下は、各暗号をサポートしている、FS がサポートするセキュリティポリシーの一覧です。

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-ECDSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c02b
OpenSSL – ECDHE-RSA-AES128-GCM-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c02f
OpenSSL – ECDHE-ECDSA-AES128-SHA256 IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c023
OpenSSL – ECDHE-RSA-AES128-SHA256 IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c027
OpenSSL – ECDHE-ECDSA-AES128-SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c009

暗号名	セキュリティポリシー	暗号スイート
IANA – TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA		
OpenSSL – ECDHE-RSA-AES128-SHA IANA – TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c013
OpenSSL – ECDHE-ECDSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c02c
OpenSSL – ECDHE-RSA-AES256-GCM-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2020-10 • ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c030
OpenSSL – ECDHE-ECDSA-AES256-SHA384 IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c024

暗号名	セキュリティポリシー	暗号スイート
OpenSSL – ECDHE-RSA-AES256-SHA384 IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	• ELBSecurityPolicy-FS-1-2-Res-2019-08 • ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c028
OpenSSL – ECDHE-ECDSA-AES256-SHA IANA – TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c00a
OpenSSL – ECDHE-RSA-AES256-SHA IANA – TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	• ELBSecurityPolicy-FS-1-2-2019-08 • ELBSecurityPolicy-FS-1-1-2019-08 • ELBSecurityPolicy-FS-2018-06	c014

Network Load Balancer のリスナーを更新する

リスナープロトコル、リスナーポート、または転送アクションからのトラフィックを受信するターゲットグループを更新できます。デフォルトアクションはデフォルトルールとも呼ばれ、選択したターゲットグループにリクエストを転送します。

TCP、UDP、または QUIC から TLS にプロトコルを変更した場合、セキュリティポリシーとサーバー証明書を指定する必要があります。TLS から TCP、UDP、または QUIC にプロトコルを変更した場合、セキュリティポリシーとサーバー証明書は削除されます。

TCP、TLS、または QUIC リスナーのデフォルトアクションのターゲットグループが更新されると、新しい接続は新しく設定されたターゲットグループにルーティングされます。ただし、この変更以前に作成されたアクティブな接続には影響しません。これらのアクティブな接続は、トラフィックが送信されている場合は最大 1 時間、トラフィックが送信されていない場合はアイドルタイムアウト期間が経過するまでのいずれか早い方まで、元のターゲットグループのターゲットに関連付けられたままになります。このパラメーター `Connection termination on deregistration` は、ターゲットの登録解除時に適用されるため、リスナーの更新時には適用されません。

QUIC または TCP_QUIC リスナーのポート更新は許可されません。QUIC トラフィックを処理するリスナーのポートを更新するには、リスナーを削除して新しいポートで再作成する必要があります。

Console

リスナーを更新するには

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、[ロードバランサー] を選択します。
3. ロードバランサーの名前を選択して、その詳細ページを開きます。
4. [リスナー] タブで、[プロトコル:ポート] 列のテキストを選択して、リスナーの詳細ページを開きます。
5. [アクション]、[リスナーを編集] の順に選択します。
6. 必要に応じて値を更新します。
   - (オプション) プロトコルを変更します。
   - (オプション) ポートを変更します。
   - (オプション) [デフォルトアクション] の別のターゲットグループを選択します。
   - (オプション) 別のターゲットグループを追加するには、[ターゲットグループを追加] を選択し、必要に応じて重みを更新します。
   - (オプション) データソースを削除するには、[削除] を選択します。
7. (オプション) 必要に応じてタグを追加、更新、または削除します。
8. [Save changes] (変更の保存) をクリックします。

AWS CLI

デフォルトアクションを更新するには

ターゲットグループを変更するには、次の modify-listener コマンドを使用します。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --default-actions Type=forward,TargetGroupArn=new-target-group-arn
```

次の例では、複数のターゲットグループを持つリスナーを更新します。

```
aws elbv2 modify-listener \
```

```
    --listener-arn listener-arn \
    --default-actions '[{
        "Type":"forward",
        "ForwardConfig":{
            "TargetGroups":[
                {"TargetGroupArn":"target-group-1-arn","Weight":10},
                {"TargetGroupArn":"target-group-2-arn","Weight":30}
            ]
        }
    }]'
```

タグを追加するには

add-tags コマンドを使用します。次の例では、2 つのタグを追加します。

```
aws elbv2 add-tags \
    --resource-arns listener-arn \
    --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

タグを削除するには

remove-tags コマンドを使用します。次の例では、指定したキーを使用してタグを削除します。

```
aws elbv2 remove-tags \
    --resource-arns listener-arn \
    --tag-keys project department
```

CloudFormation

デフォルトアクションを更新するには

AWS::ElasticLoadBalancingV2::Listener リソースを更新して、新しいターゲットグループを含めます。

```
Resources:
  myTCPListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: TCP
      Port: 80
      DefaultActions:
        - Type: forward
```

```
            TargetGroupArn: !Ref newTargetGroup
```

または、複数のターゲットグループ間でトラフィックを分散するには、次のように DefaultActions を定義します。

```
DefaultActions:
    - Type: forward
    ForwardConfig:
        TargetGroups:
        - TargetGroupArn: !Ref TargetGroup1
            Weight: 10
        - TargetGroupArn: !Ref TargetGroup2
            Weight: 30
```

タグを追加するには

AWS::ElasticLoadBalancingV2::Listener リソースを更新して、[タグ] プロパティを含めます。

```
Resources:
  myTCPListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: TCP
      Port: 80
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
      Tags:
        - Key: 'project'
          Value: 'lima'
        - Key: 'department'
          Value: 'digital-media'
```

Network Load Balancer リスナーの TCP アイドルタイムアウトを更新する

Network Load Balancer を通じて行う TCP リクエストごとに、その接続の状態が追跡されます。アイドルタイムアウトよりも長い時間、クライアントからもターゲットからもその接続経由でデータが送信されない場合、接続は閉じられます。

考慮事項

- TCP フローの場合、デフォルトのアイドルタイムアウト値は 350 秒です。
- TLS リスナーの接続アイドルタイムアウトは 350 秒であり、変更できません。

Console

TCP アイドルタイムアウトを更新するには

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の [Load Balancing] で、[Load Balancers] を選択します。
3. Network Load Balancer のチェックボックスをオンにします。
4. [リスナー] タブで、TCP リスナーのチェックボックスを選択してから、[アクション]、[リスナーの詳細を表示] を選択します。
5. リスナーの詳細ページの [属性] タブで [編集] を選択します。リスナーが TCP 以外のプロトコルを使用している場合、このタブは存在しません。
6. 60～6,000 秒の TCP アイドルタイムアウトの値を入力します。
7. [Save changes] (変更の保存) をクリックします。

AWS CLI

TCP アイドルタイムアウトを更新するには

`tcp.idle_timeout.seconds` 属性を指定して modify-listener-attributes コマンドを使用します。

```
aws elbv2 modify-listener-attributes \
    --listener-arn listener-arn \
    --attributes Key=tcp.idle_timeout.seconds,Value=500
```

以下は出力の例です。

```
{
    "Attributes": [
        {
            "Key": "tcp.idle_timeout.seconds",
            "Value": "500"
```

```
        }
    ]
}
```

CloudFormation

TCP アイドルタイムアウトを更新するには

[AWS::ElasticLoadBalancingV2::Listener](#) リソースを更新して、`tcp.idle_timeout.seconds` リスナー属性を含めます。

```
Resources:
  myTCPListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: TCP
      Port: 80
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
      ListenerAttributes:
        - Key: "tcp.idle_timeout.seconds"
          Value: "500"
```

Network Load Balancer の TLS リスナーを更新する

TLS リスナーを作成すると、デフォルトの証明書の置き換え、証明書リストからの証明書の追加または削除、セキュリティポリシーの更新、または ALPN ポリシーの更新を行うことができます。

タスク

- デフォルトの証明書の置き換え
- 証明書リストに証明書を追加する
- 証明書リストから証明書を削除する
- セキュリティポリシーの更新
- ALPN ポリシーを更新するには

デフォルトの証明書の置き換え

必要に応じて、TLS リスナーのデフォルト証明書を置き換えることができます。詳細については、「デフォルトの証明書」を参照してください。

Console

デフォルトの証明書を置き換えるには

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、[ロードバランサー] を選択します。
3. ロードバランサーを選択します。
4. [リスナー] タブで、[プロトコル:ポート] 列のテキストを選択して、リスナーの詳細ページを開きます。
5. [証明書] タブで、[デフォルトを変更] を選択します。
6. [ACM および IAM 証明書] 表内の新しいデフォルト証明書を選択します。
7. (オプション) デフォルトでは、[リスナー証明書リストに以前のデフォルト証明書を追加] を選択します。現在 SNI のリスナー証明書がなく、TLS セッションの再開に依存していない限り、このオプションを選択しておくことをお勧めします。
8. [デフォルトとして保存] を選択します。

AWS CLI

デフォルトの証明書を置き換えるには

modify-listener コマンドを使用します。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --certificates CertificateArn=new-default-certificate-arn
```

CloudFormation

デフォルトの証明書を置き換えるには

新しいデフォルト証明書を使用して、AWS::ElasticLoadBalancingV2::Listener リソースを更新します。

```
Resources:
```

```
  myTLSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: TLS
      Port: 443
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
      SslPolicy: "ELBSecurityPolicy-TLS13-1-2-2021-06"
      Certificates:
        - CertificateArn: "new-default-certificate-arn"
```

証明書リストに証明書を追加する

次の手順でリスナーの証明書リストに証明書を追加できます。最初に TLS リスナーを作成したときは、証明書リストは空です。デフォルトの証明書として置き換えても、この証明書が SNI プロトコルで使用されるように、デフォルトの証明書を証明書リストに追加できます。詳細については、「証明書リスト」を参照してください。

Console

証明書リストに証明書を追加するには

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、[ロードバランサー] を選択します。
3. ロードバランサーの名前を選択して、その詳細ページを開きます。
4. [リスナー] タブで、[プロトコル:ポート] 列のテキストを選択して、リスナーの詳細ページを開きます。
5. [証明書] タブを選択します。
6. デフォルトの証明書をリストに追加するには、[デフォルトをリストに追加] を選択します。
7. デフォルト以外の証明書をリストに追加するには、次の手順を実行します。

    a. [証明書を追加] を選択します。
    b. ACM または IAM によって既に管理されている証明書を追加するには、その証明書のチェックボックスを選択して [保留中として以下を含める] を選択します。
    c. ACM または IAM が管理していない証明書を追加するには、[証明書をインポート] を選択し、フォームに記入して、[インポート] を選択します。

d. [保留中の証明書を追加] を選択します。

AWS CLI

証明書リストに証明書を追加するには

add-listener-certificates コマンドを使用します。

```
aws elbv2 add-listener-certificates \
    --listener-arn listener-arn \
    --certificates \
        CertificateArn=certificate-arn-1 \
        CertificateArn=certificate-arn-2 \
        CertificateArn=certificate-arn-3
```

CloudFormation

証明書リストに証明書を追加するには

AWS::ElasticLoadBalancingV2::ListenerCertificate タイプのリソースを定義します。

```
Resources:
  myCertificateList:
    Type: 'AWS::ElasticLoadBalancingV2::ListenerCertificate'
    Properties:
      ListenerArn: !Ref myTLSListener
      Certificates:
        - CertificateArn: "certificate-arn-1"
        - CertificateArn: "certificate-arn-2"
        - CertificateArn: "certificate-arn-3"

  myTLSListener:
    Type: AWS::ElasticLoadBalancingV2::Listener
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: TLSS
      Port: 443
      SslPolicy: "ELBSecurityPolicy-TLS13-1-2-2021-06"
      Certificates:
        - CertificateArn: "certificate-arn-1"
      DefaultActions:
        - Type: forward
```

```
            TargetGroupArn: !Ref myTargetGroup
```

証明書リストから証明書を削除する

次の手順で TLS リスナーの証明書リストから証明書を削除できます。証明書を削除すると、リスナーはその証明書を使用して接続を作成できなくなります。クライアントが影響を受けないようにするには、新しい証明書をリストに追加し、リストから証明書を削除する前に接続が機能していることを確認します。

TLS リスナーのデフォルトの証明書を削除するには、デフォルトの証明書の置き換え を参照してください。

Console

証明書リストから証明書を削除するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[ロードバランサー] を選択します。
3. ロードバランサーの名前を選択して、その詳細ページを開きます。
4. [リスナー] タブで、[プロトコル:ポート] 列のテキストを選択して、リスナーの詳細ページを開きます。
5. [証明書] タブで、証明書のチェックボックスを選択し、[削除] を選択します。
6. 確認を求められたら、**confirm** と入力し、[削除] を選択します。

AWS CLI

証明書リストから証明書を削除するには

remove-listener-certificates コマンドを使用します。

```
aws elbv2 remove-listener-certificates \
    --listener-arn listener-arn \
    --certificates CertificateArn=certificate-arn
```

セキュリティポリシーの更新

TLS リスナーを作成するときに、ニーズを満たすセキュリティポリシーを選択できます。新しいセキュリティのポリシーを追加したら、TLS リスナーを更新して新しいセキュリティポリシーを使用できます。Network Load Balancer は、カスタムセキュリティポリシーをサポートしていません。詳細については、「Network Load Balancer のセキュリティポリシー」を参照してください。

セキュリティポリシーを更新すると、ロードバランサーが大量のトラフィックを処理している場合に中断が発生する可能性があります。ロードバランサーが大量のトラフィックを処理しているときに中断の可能性を減らすには、トラフィックの処理に役立つ追加のロードバランサーを作成するか、LCU 予約をリクエストします。

Console

セキュリティポリシーを更新するには

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、[ロードバランサー] を選択します。
3. ロードバランサーの名前を選択して、その詳細ページを開きます。
4. [リスナー] タブで、[プロトコル:ポート] 列のテキストを選択して、リスナーの詳細ページを開きます。
5. [アクション]、[リスナーを編集] の順に選択します。
6. [セキュアリスナーの設定] セクションの [セキュリティポリシー] で、新しいセキュリティポリシーを選択します。
7. [Save changes] (変更の保存) をクリックします。

AWS CLI

セキュリティポリシーを更新するには

modify-listener コマンドを使用します。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --ssl-policy ELBSecurityPolicy-TLS13-1-2-Res-2021-06
```

CloudFormation

セキュリティポリシーを更新するには

新しいセキュリティポリシーを使用して、AWS::ElasticLoadBalancingV2::Listener リソースを更新します。

```
Resources:
  myTLSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: TLS
      Port: 443
      SslPolicy: "ELBSecurityPolicy-TLS13-1-2-2021-06"
      Certificates:
        - CertificateArn: "default-certificate-arn"
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
```

ALPN ポリシーを更新するには

必要に応じて、TLS リスナーの ALPN ポリシーを更新できます。詳細については、「ALPN ポリシー」を参照してください。

Console

ALPN ポリシーを更新するには

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、[ロードバランサー] を選択します。
3. ロードバランサーの名前を選択して、その詳細ページを開きます。
4. [リスナー] タブで、[プロトコル:ポート] 列のテキストを選択して、リスナーの詳細ページを開きます。
5. [アクション]、[リスナーを編集] の順に選択します。
6. [セキュアリスナー設定] セクションの [ALPN ポリシー] では、ALPN を有効にするポリシーを選択するか、[なし] を選択して ALPN を無効にします。
7. [Save changes] (変更の保存) をクリックします。

AWS CLI

ALPN ポリシーを更新するには

modify-listener コマンドを使用します。

```
aws elbv2 modify-listener \
    --listener-arn listener-arn \
    --alpn-policy HTTP2Preferred
```

CloudFormation

ALPN ポリシーを更新するには

AWS::ElasticLoadBalancingV2::Listener リソースを更新して、ALPN ポリシーを含めます。

```
Resources:
  myTLSListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: TLS
      Port: 443
      SslPolicy: "ELBSecurityPolicy-TLS13-1-2-Res-2021-06"
      AlpnPolicy:
        - HTTP2Preferred
      Certificates:
        - CertificateArn: "certificate-arn"
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
```

Network Load Balancer のリスナーを削除する

リスナーを削除する前に、アプリケーションへの影響を考慮してください。

- [TCP および TLS リスナー] ロードバランサーはリスナーでの新しい接続の受け入れを直ちに停止します。進行中の TLS ハンドシェイクは失敗する可能性があります。既存の接続は、自然に閉じるかタイムアウトするまで開いたままになります。既存の接続に対する処理中のリクエストは正常に完了します。
- [UDP および QUIC リスナー] 転送中のパケットは送信先に到達しない可能性があります。

Console

リスナーを削除するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインで、[ロードバランサー] を選択します。
3. ロードバランサーのチェックボックスをオンにします。
4. [リスナー] タブで、リスナーのチェックボックスを選択してから、[アクション]、[リスナーの削除] を選択します。
5. 確認を求められたら、「**confirm**」を入力し、[削除] を選択します。

AWS CLI

リスナーを削除するには

delete-listener コマンドを使用します。

```
aws elbv2 delete-listener \
    --listener-arn listener-arn
```

Network Load Balancers のターゲットグループ

各ターゲットグループは、1 つ以上の登録されているターゲットにリクエストをルーティングするために使用されます。リスナーを作成するときは、デフォルトアクションのターゲットグループを指定します。トラフィックは、リスナー規則で指定されたターゲットグループに転送されます。さまざまなタイプのリクエストに応じて別のターゲットグループを作成できます。たとえば、一般的なリクエスト用に 1 つのターゲットグループを作成し、アプリケーションのマイクロサービスへのリクエスト用に別のターゲットグループを作成できます。詳細については、「Network Load Balancer のコンポーネント」を参照してください。

ロードバランサーのヘルスチェック設定は、ターゲットグループ単位で定義します。各ターゲットグループはデフォルトのヘルスチェック設定を使用します。ただし、ターゲットグループを作成したときや、後で変更したときに上書きした場合を除きます。リスナーのルールでターゲットグループを指定すると、ロードバランサーは、ロードバランサーで有効なアベイラビリティーゾーンにある、ターゲットグループに登録されたすべてのターゲットの状態を継続的にモニタリングします。ロードバランサーは、正常な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。詳細については、「Network Load Balancer ターゲットグループのヘルスチェック」を参照してください。

目次

ルーティング設定

デフォルトでは、ロードバランサーはターゲットグループの作成時に指定したプロトコルとポート番号を使用して、リクエストをターゲットにルーティングします。または、ターゲットグループへの登録時にターゲットへのトラフィックのルーティングに使用されるポートを上書きすることもできます。

Network Load Balancer のターゲットグループは、次のプロトコルとポートをサポートします。

- プロトコル: TCP、TLS、UDP、TCP_UDP、QUIC、TCP_QUIC
- ポート: 1 ～ 65535

ターゲットグループに TLS プロトコルが設定されている場合、ロードバランサーは、ターゲットにインストールした証明書を使用して、ターゲットと TLS 接続を確立します。ロードバランサーはこれらの証明書を検証しません。したがって、自己署名証明書または期限切れの証明書を使用できます。ロードバランサーは仮想プライベートクラウド (VPC) 内にあるため、ロードバランサーとターゲット間のトラフィックはパケットレベルで認証されるため、ターゲットの証明書が有効でない場合でも、中間者攻撃やスプーフィングのリスクはありません。

次の表は、リスナープロトコルとターゲットグループの設定のサポートされている組み合わせをまとめたものです。

リスナープロトコル	ターゲットグループプロトコル	ターゲットグループの種類	ヘルスチェックプロトコル
TCP	TCP | TCP_UDP | TCP_QUIC	インスタンス | ip	HTTP | HTTPS | TCP
TCP	TCP	alb	HTTP | HTTPS
TLS	TCP | TLS	インスタンス | ip	HTTP | HTTPS | TCP
UDP	UDP | TCP_UDP	インスタンス | ip	HTTP | HTTPS | TCP
TCP_UDP	TCP_UDP	インスタンス | ip	HTTP | HTTPS | TCP
QUIC	QUIC | TCP_QUIC	インスタンス | ip	HTTP | HTTPS | TCP
TCP_QUIC	TCP_QUIC	インスタンス | ip	HTTP | HTTPS | TCP

[Target type (ターゲットタイプ)]

ターゲットグループを作成するときは、そのターゲットの種類を指定します。ターゲットの種類は、ターゲットの指定方法を決定します。ターゲットグループを作成した後で、ターゲットタイプを変更することはできません。

可能なターゲットの種類は次のとおりです。

`instance`

インスタンス ID で指定されたターゲット。

`ip`

IP アドレスで指定されたターゲット。

`alb`

ターゲットは Application Load Balancer です。

ターゲットの種類が `ip` の場合、次のいずれかの CIDR ブロックから IP アドレスを指定できます。

- ターゲットグループの VPC のサブネット
- 10.0.0.0/8 (RFC 1918)
- 100.64.0.0/10 (RFC 6598)
- 172.16.0.0/12 (RFC 1918)
- 192.168.0.0/16 (RFC 1918)

> **Important**
>
> パブリックにルーティング可能な IP アドレスは指定できません。

サポートされているすべての CIDR ブロックによって、次のターゲットをターゲットグループに登録できます。

- AWS IP アドレスとポートでアドレス可能な リソース (データベースなど) 。
- Direct Connect または Site-to-Site VPN 接続 AWS を介して にリンクされたオンプレミスリソース。

ターゲットグループでクライアント IP の保存が無効化されている場合、ロードバランサーは Network Load Balancer の IP アドレスと一意のターゲット (IP アドレスとポート) の組み合わせごとに 1 分あたり約 55,000 の接続をサポートできます。これらの接続数を超えた場合、ポート割り当てエラーが発生する可能性が高くなります。ポート割り当てエラーが発生した場合は、ターゲットグループにさらに多くのターゲットを追加します。

共有 VPC で (参加者として) Network Load Balancer を起動した場合、登録できるのは、共有されているサブネット内のターゲットだけです。

ターゲットタイプが `alb` の場合、単一の Application Load Balancer をターゲットとして登録できます。詳細については、「Network Load Balancer のターゲットとして Application Load Balancer を使用する」を参照してください。

Network Load Balancer は、`lambda` ターゲットタイプをサポートしていません。Application Load Balancer は、`lambda` ターゲットタイプをサポートする唯一のロードバランサーです。詳細については、「Application Load Balancer ユーザーガイド」の「ターゲットとしての Lambda 関数」を参照してください。

Network Load Balancer に登録されているインスタンスでマイクロサービスを使用している場合、ロードバランサーを使用してインスタンス間の通信を提供することはできません。ただし、ロードバランサーがインターネット向けであるか、インスタンスが IP アドレスで登録されている場合は除きます。詳しくは、「ターゲットからそのロードバランサーへのリクエストが接続タイムアウトになる」を参照してください。

リクエストのルーティングと IP アドレス

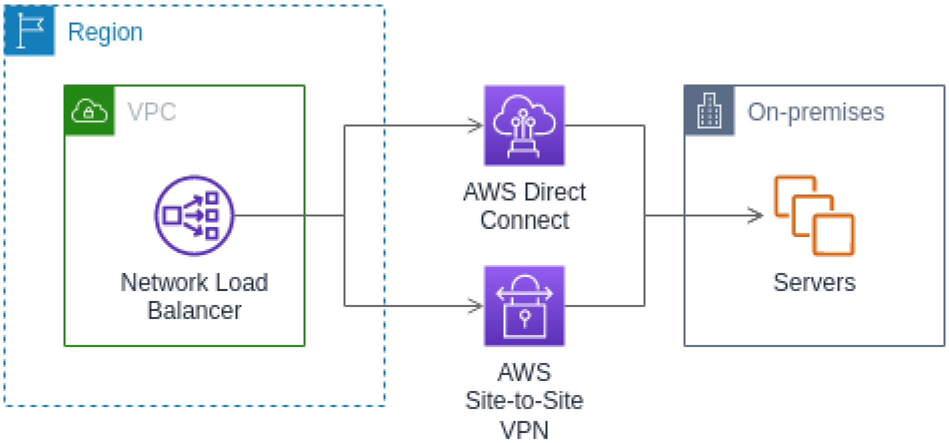
インスタンス ID を使用してターゲットを指定すると、トラフィックはインスタンスのプライマリネットワークインターフェイスで指定されたプライマリプライベート IP アドレスを使用して、インスタンスにルーティングされます。ロードバランサーは、データパケットの宛先 IP アドレスを書き換えてから、ターゲットインスタンスに転送します。

IP アドレスを使用してターゲットを指定する場合は、1 つまたは複数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 アドレスを使用して、トラフィックをインスタンスにルーティングできます。これにより、インスタンスの複数のアプリケーションが同じポートを使用できるようになります。各ネットワークインターフェイスはそれぞれ独自のセキュリティグループを割り当てることができます。ロードバランサーは、宛先 IP アドレスを書き換えてから、ターゲットに転送します。

インスタンスへのトラフィックの許可の詳細については、ターゲットセキュリティグループ を参照してください。

ターゲットとしてのオンプレミスリソース

Direct Connect または Site-to-Site VPN 接続を介してリンクされたオンプレミスリソースは、ターゲットタイプが の場合、ターゲットとして機能します`ip`。



オンプレミスのリソースを使用する場合、これらのターゲットの IP アドレスは、引き続き次の CIDR ブロックのいずれかから取得する必要があります。

- 10.0.0.0/8 (RFC 1918)
- 100.64.0.0/10 (RFC 6598)
- 172.16.0.0/12 (RFC 1918)
- 192.168.0.0/16 (RFC 1918)

詳細については Direct Connect、「 とは」を参照してください Direct Connect。

詳細については AWS Site-to-Site VPN、「 とは」を参照してください AWS Site-to-Site VPN。

IP アドレスタイプ

新しいターゲットグループを作成するときは、ターゲットグループの IP アドレスタイプを選択できます。これは、ターゲットとの通信、およびそれらのヘルスステータスのチェックに使用される IP バージョンを制御します。

お使いの Network Load Balancer のターゲットグループは、次の IP アドレスタイプをサポートしています。

`ipv4`

ロードバランサーは IPv4 を使用してターゲットと通信します。

`ipv6`

ロードバランサーは IPv6 を使用してターゲットと通信します。

考慮事項

- ロードバランサーは、ターゲットグループの IP アドレスのタイプに基づいてターゲットと通信します。IPv4 ターゲットグループのターゲットはロードバランサーからの IPv4 トラフィックを受け入れる必要があり、IPv6 ターゲットグループのターゲットはロードバランサーからの IPv6 トラフィックを受け入れる必要があります。
- `ipv4` ロードバランサーで IPv6 ターゲットグループを使用することはできません。
- `dualstack` ロードバランサーの UDP リスナーで IPv4 ターゲットグループを使用することはできません。
- IPv6 ターゲットグループを使用して Application Load Balancer を登録することはできません。
- QUIC または TCP_QUIC プロトコルで IPv6 ターゲットグループを使用することはできません。

登録済みターゲット

ロードバランサーは、クライアントにとって単一の通信先として機能し、正常な登録済みターゲットに受信トラフィックを分散します。各ターゲットグループでは、ロードバランサーが有効になっている各アベイラビリティーゾーンで少なくとも 1 つのターゲットが登録されている必要があります。各ターゲットは、1 つ以上のターゲットグループに登録できます。

アプリケーションの需要が高まった場合、需要に対処するため、1 つまたは複数のターゲット グループに追加のターゲットを登録できます。設定したしきい値に関係なく、登録処理が完了し、ターゲットが最初の初期ヘルスチェックに合格するとすぐに、ロードバランサーは新しく登録したターゲットへのトラフィックのルーティングを開始します。

アプリケーションの需要が低下した場合や、ターゲットを保守する必要がある場合、ターゲットグループからターゲットを登録解除することができます。ターゲットを登録解除するとターゲットグループから削除されますが、ターゲットにそれ以外の影響は及びません。登録解除するとすぐに、ロードバランサーはターゲットへのトラフィックのルーティングを停止します。ターゲットは、未処理のリクエストが完了するまで `draining` 状態になります。トラフィックの受信を再開する準備ができると、ターゲットをターゲットグループに再度登録することができます。

インスタンス ID でターゲットを登録する場合は、Auto Scaling グループでロードバランサーを使用できます。Auto Scaling グループにターゲットグループをアタッチすると、ターゲットの起動時に Auto Scaling によりターゲットグループにターゲットが登録されます。詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「Auto Scaling グループへのロードバランサーのアタッチ」を参照してください。

要件と考慮事項

- インスタンスで使用されているインスタンスタイプが C1、CC1、CC2、CG1、CG2、CR1、G1、G2、HI1、HS1、M1、M2、M3、T1 のいずれかである場合、インスタンス ID でインスタンスを登録することはできません。
- IPv6 ターゲットグループにインスタンス ID でターゲットを登録する場合、ターゲットにはプライマリ IPv6 アドレスが割り当てられている必要があります。詳細については、「Amazon EC2 ユーザーガイド」の「IPv6 アドレス」を参照してください。
- インスタンス ID でターゲットを登録する場合、インスタンスは Network Load Balancer と同じ VPC にある必要があります。ロードバランサー VPC (同じリージョンまたは異なるリージョン) とピア接続されている VPC にインスタンスがある場合、そのインスタンスをインスタンス ID で登録することはできません。このようなインスタンスは IP アドレスで登録できます。
- ターゲットを IP アドレスで登録し、その IP アドレスがロードバランサーと同じ VPC にある場合、ロードバランサーは、到達可能なサブネットからターゲットがアクセスしていることを確認します。
- ロードバランサーは、有効になっているアベイラビリティーゾーン内のターゲットのみにトラフィックをルーティングします。有効になっていないゾーン内のターゲットは使用されません。
- UDP、TCP_UDP、QUIC、TCP_QUIC ターゲットグループの場合、インスタンスがロードバランサー VPC の外部に存在するか、インスタンスタイプとして C1、CC1、CC2、CG1、CG2、CR1、G1、G2、HI1、HS1、M1、M2、M3、T1 のいずれかを使用しているときは、IP アドレスでインスタンスを登録しないでください。ロードバランサー VPC の外部に存在するか、サポートされていないインスタンスタイプを使用するターゲットは、ロードバランサーからのトラフィックを受信できても、応答できない場合があります。

ターゲットグループの属性

ターゲットグループは属性を編集することで設定できます。詳細については、「ターゲットグループ属性を編集する」を参照してください。

次のターゲット グループの属性がサポートされています。これらの属性は、ターゲットグループタイプが instance または ip の場合にのみ変更できます。ターゲットグループタイプが alb の場合、これらの属性は常にデフォルト値を使用します。

deregistration_delay.timeout_seconds

登録解除するターゲットの状態が draining から unused に変わるのを Elastic Load Balancing が待機する時間。範囲は 0 ~ 3600 秒です。デフォルト値は 300 秒です。QUIC トラフィックの場合、値は常に 300 秒です。

deregistration_delay.connection_termination.enabled

ロードバランサーが登録解除タイムアウトの終了時に接続を終了するかどうかを示します。値は true または false です。新しい UDP/TCP_UDP ターゲットグループの場合、デフォルトは true です。それ以外の場合は、デフォルトは false です。この属性は QUIC トラフィックには適用されません。

load_balancing.cross_zone.enabled

クロスゾーンロードバランサーが有効かどうかを示します。値は true、false または use_load_balancer_configuration です。デフォルトは use_load_balancer_configuration です。

preserve_client_ip.enabled

クライアント IP の保存が有効かどうかを示します。値は true または false です。ターゲットグループの種類が IP アドレスで、ターゲットグループプロトコルが TCP または TLS の場合、デフォルトは無効です。それ以外の場合、デフォルトは有効です。UDP、TCP_UDP、QUIC、TCP_QUIC ターゲットグループのクライアント IP 保存を無効にすることはできません。

proxy_protocol_v2.enabled

Proxy Protocol バージョン 2 が有効になっているかどうかを示します。Proxy Protocol は、デフォルトで無効になっています。

stickiness.enabled

スティッキーセッションが有効かどうかを示します。値は true または false です。デフォルトは false です。この属性は QUIC トラフィックには適用されません。

stickiness.type

維持の種類です。有効な値は source_ip です。

`target_group_health.dns_failover.minimum_healthy_targets.count`

正常でなければならないターゲットの最小数。正常なターゲットの数がこの値を下回っている場合は、DNS でそのゾーンを異常とマークして、トラフィックが正常なゾーンにのみルーティングされるようにします。指定できる値は、`off` または 1 から最大ターゲット数までの整数です。`off` の場合、DNS フェイルアウェイは無効化されます。つまり、ターゲットグループ内のすべてのターゲットが異常であっても、ゾーンは DNS から削除されません。デフォルトは 1 です。

`target_group_health.dns_failover.minimum_healthy_targets.percentage`

正常でなければならないターゲットの最小割合。正常なターゲットの割合がこの値を下回っている場合は、DNS でそのゾーンを異常とマークして、トラフィックが正常なゾーンにのみルーティングされるようにします。指定できる値は、`off` または 1 から 100 までの整数です。`off` の場合、DNS フェイルアウェイは無効化されます。つまり、ターゲットグループ内のすべてのターゲットが異常であっても、ゾーンは DNS から削除されません。デフォルトは `off` です。

`target_group_health.unhealthy_state_routing.minimum_healthy_targets.count`

正常でなければならないターゲットの最小数。正常なターゲットの数がこの値を下回っている場合は、異常なターゲットを含むすべてのターゲットにトラフィックを送信します。指定できる値は、1～最大ターゲット数です。デフォルトは 1 です。

`target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage`

正常でなければならないターゲットの最小割合。正常なターゲットの割合がこの値を下回っている場合は、異常なターゲットを含むすべてのターゲットにトラフィックを送信します。指定できる値は、`off` または 1 から 100 までの整数です。デフォルトは `off` です。

`target_health_state.unhealthy.connection_termination.enabled`

ロードバランサーが異常なターゲットへの接続を終了するかどうかを示します。値は `true` または `false` です。デフォルトは `true` です。

`target_health_state.unhealthy.draining_interval_seconds`

異常なターゲットの状態が `unhealthy.draining` から `unhealthy` に変わるのを Elastic Load Balancing が待機する時間。範囲は 0～360,000 秒です。デフォルト値は0秒です。

[注:] この属性は、`target_health_state.unhealthy.connection_termination.enabled` が `false` の場合にのみ設定できます。

ターゲットグループの正常性

デフォルトでは、ターゲットグループが少なくとも 1 つの正常なターゲットを持っている限り、そのターゲットグループは正常であると見なされます。フリートが大きい場合、トラフィックを処理する正常なターゲットが 1 つだけでは十分ではありません。代わりに、正常でなければならないターゲットの最小数または割合、および正常なターゲットが指定されたしきい値を下回ったときにロードバランサーが実行するアクションを指定できます。これはアプリケーションの可用性を向上します。

内容

異常な状態アクション

以下のアクションに対して正常なしきい値を設定できます。

- DNS フェイルオーバー – ゾーン内の正常なターゲットがしきい値を下回ると、そのゾーンのロードバランサーノードの IP アドレスが DNS で異常とマークされます。そのため、クライアントがロードバランサーの DNS 名を解決すると、トラフィックは正常なゾーンにのみルーティングされます。
- ルーティングフェイルオーバー – ゾーン内の正常なターゲットがしきい値を下回ると、ロードバランサーは、異常なターゲットを含め、ロードバランサーノードで使用可能なすべてのターゲットにトラフィックを送信します。これにより、特にターゲットが一時的にヘルスチェックに合格しなかった場合に、クライアント接続が成功する可能性が高まり、正常なターゲットが過負荷になるリスクが軽減されます。

要件と考慮事項

- アクションに両方のタイプのしきい値 (数と割合) を指定した場合、ロードバランサーはどちらかのしきい値を超えるとアクションを実行します。
- 両方のアクションにしきい値を指定する場合、DNS フェイルオーバーのしきい値はルーティングフェイルオーバーのしきい値以上である必要があります。これにより、DNS フェイルオーバーはルーティングフェイルオーバーの有無にかかわらず発生します。

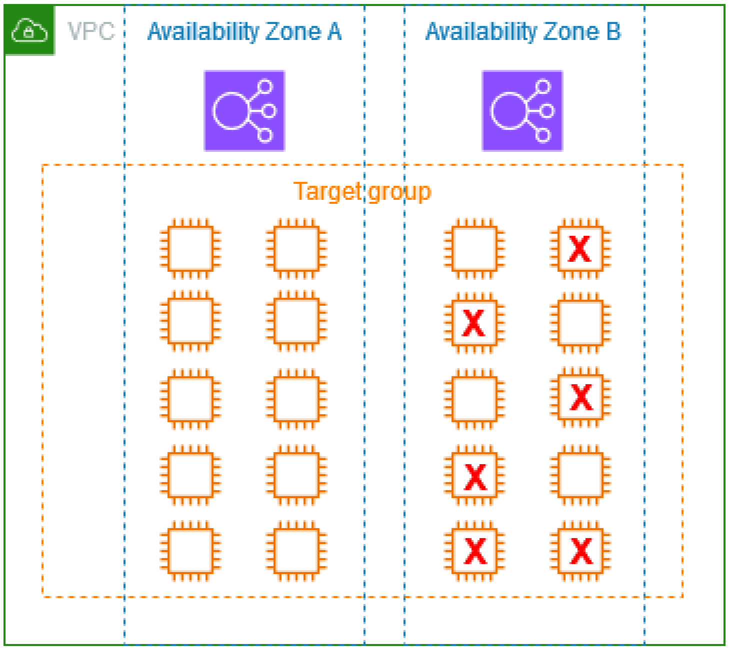
- しきい値を割合として指定すると、ターゲットグループに登録されているターゲットの総数に基づいて、値が動的に計算されます。
- ターゲットの合計数は、クロスゾーンロードバランサーがオフになっているかオンになっているかによって決まります。クロスゾーンロードバランサーがオフの場合、各ノードは独自のゾーン内のターゲットにのみトラフィックを送信します。つまり、しきい値は有効になっている各ゾーンのターゲット数に個別に適用されます。クロスゾーンロードバランサーがオンの場合、各ノードは有効なすべてのゾーンのすべてのターゲットにトラフィックを送信します。つまり、指定されたしきい値が有効になっているすべてのゾーンのターゲットの総数に適用されます。詳細については、「クロスゾーンロードバランサー」を参照してください。
- DNS フェイルオーバーが発生すると、ロードバランサーに関連するすべてのターゲットグループに影響します。特にクロスゾーンロードバランサーがオフになっている場合は、この追加のトラフィックを処理するのに十分な容量が残りのゾーンにあることを確認してください。
- DNS フェイルオーバーでは、ロードバランサーの DNS ホスト名から異常のあるゾーンの IP アドレスを削除します。ただし、ローカルクライアントの DNS キャッシュには、DNS レコードの有効期限 (TTL) が期限切れになる (60 秒) まで、これらの IP アドレスが含まれる場合があります。
- DNS フェイルオーバーでは、Network Load Balancer に複数のターゲットグループがアタッチされていて、ゾーン内で 1 つのターゲットグループが異常である場合、そのゾーン内の別のターゲットグループが正常であっても、DNS フェイルオーバーが発生します。
- DNS フェイルオーバーでは、すべてのロードバランサーゾーンが異常と見なされると、ロードバランサーは異常なゾーンを含むすべてのゾーンにトラフィックを送信します。
- DNS フェイルオーバーにつながる可能性のある正常なターゲットが十分にあるかどうか以外にも、ゾーンのヘルスなどの要因があります。

例

次の例は、ターゲットグループのヘルス設定がどのように適用されるかを示しています。

シナリオ

- 2 つのアベイラビリティーゾーン A と B をサポートするロードバランサー
- 各アベイラビリティーゾーンには 10 の登録済みターゲットが含まれています
- ターゲットグループには、次のターゲットグループのヘルス設定があります。
  - DNS フェイルオーバー - 50%
  - ルーティングフェイルオーバー - 50%
- アベイラビリティーゾーン B で 6 つのターゲットが失敗



クロスゾーンロードバランサーがオフの場合

- 各アベイラビリティーゾーンのロードバランサーノードは、アベイラビリティーゾーンの 10 個のターゲットにのみトラフィックを送信できます。
- アベイラビリティーゾーン A には 10 個の正常なターゲットがあり、これは正常なターゲットの必要な割合を満たしています。ロードバランサーは引き続き、10 の正常なターゲット間でトラフィックを分散します。
- アベイラビリティーゾーン B には正常なターゲットが 4 つしかなく、これはアベイラビリティーゾーン B のロードバランサーノードのターゲットの 40% です。これは正常なターゲットの必要なパーセンテージを下回っているため、ロードバランサーは次のアクションを実行します。
  - DNS フェイルオーバー - アベイラビリティーゾーン B が DNS で異常とマークされています。クライアントはロードバランサー名をアベイラビリティーゾーン B のロードバランサーノードに解決できず、アベイラビリティーゾーン A は正常であるため、クライアントはアベイラビリティーゾーン A に新しい接続を送信します。
  - ルーティングフェイルオーバー - 新しい接続がアベイラビリティーゾーン B に明示的に送信されると、ロードバランサーは、異常なターゲットを含むアベイラビリティーゾーン B のすべてのターゲットにトラフィックを分散します。これにより、残りの正常なターゲット間でのシステム停止を防ぐことができます。

クロスゾーンロードバランサーがオンの場合

- 各ロードバランサーノードは、両方のアベイラビリティーゾーンの 20 の登録済みターゲットすべてにトラフィックを送信できます。
- アベイラビリティーゾーン A には 10 個の正常なターゲット、アベイラビリティーゾーン B には 4 個の正常なターゲット、合計 14 個の正常なターゲットがあります。これは両方のアベイラビリティーゾーンのロードバランサーノードのターゲットの 70% であり、正常なターゲットの必要な割合を満たしています。
- ロードバランサーは、両方のアベイラビリティーゾーンの 14 個の正常なターゲット間でトラフィックを分散します。

ロードバランサーの Route 53 DNS フェイルオーバーを使用する

Route 53 を使用して DNS クエリをロードバランサーにルーティングする場合は、同時に Route 53 によりロードバランサーの DNS フェイルオーバーを設定することもできます。フェイルオーバー設定では、ロードバランサー用のターゲットグループのターゲットに関する正常性チェックが Route 53 によって行われ、利用可能かどうかが判断されます。ロードバランサーに正常なターゲットが登録されていない場合、またはロードバランサー自体で不具合が発生している場合、Route 53 は、トラフィックを別の利用可能なリソース (正常なロードバランサーや、Amazon S3 にある静的ウェブサイトなど) にルーティングします。

例えば、`www.example.com` 用のウェブアプリケーションがあり、異なるリージョンにある 2 つのロードバランサーの背後で冗長なインスタンスを実行するとします。1 つのリージョンのロードバランサーは、主にトラフィックのルーティング先として使用し、もう 1 つのリージョンのロードバランサーは、エラー発生時のバックアップとして使用します DNS フェイルオーバーを設定する場合は、プライマリおよびセカンダリ (バックアップ) ロードバランサーを指定できます。Route 53 は、プライマリロードバランサーが利用可能な場合はプライマリロードバランサーにトラフィックをルーティングし、利用できない場合はセカンダリロードバランサーにルーティングします。

ターゲットヘルスの評価の仕組み

- Network Load Balancer のエイリアスレコードで、ターゲットの正常性の評価が `Yes` に設定されていると、`alias target` 値で指定されたリソースの正常性が、Route 53 により評価されます。Route 53 はターゲットグループのヘルスチェックを使用します。
- Network Load Balancer 内のターゲットグループがすべて正常な場合、Route 53 はそのエイリアスレコードを正常とマークします。ターゲットグループのしきい値を設定し、そのしきい値を満たすと、ヘルスチェックに合格します。あるいは、ターゲットグループが正常なターゲットを 1 つで

も含んでいれば、そのターゲットグループはヘルスチェックに合格します。ヘルスチェックに合格すると、Route 53 はルーティングポリシーに従ってレコードを返します。フェイルオーバールーティングポリシーを使用すると、Route 53 はプライマリレコードを返します。

- Network Load Balancer のターゲットグループにアタッチされたすべてのターゲットグループに異常がある場合、そのエイリアスレコードは Route 53 のヘルスチェックに失敗します (fail-open) 。ターゲットヘルスの評価を使用すると、フェイルオーバールーティングポリシーによってトラフィックがセカンダリリソースにリダイレクトされます。
- Network Load Balancer のすべてのターゲットグループが空 (ターゲットが存在しない状態) の場合、Route 53 は対象のレコードを異常と見なします (fail-open)。ターゲットヘルスの評価を使用すると、フェイルオーバールーティングポリシーによってトラフィックがセカンダリリソースにリダイレクトされます。

詳細については、 AWS ブログの「ロードバランサーターゲットグループのヘルスしきい値を使用して可用性を向上させる」および Amazon Route 53 デベロッパーガイドの「DNS フェイルオーバーの設定」を参照してください。

Network Load Balancer のターゲットグループを作成する

Network Load Balancer のターゲットをターゲットグループに登録します。デフォルトでは、ロードバランサーはターゲットグループに指定したポートとプロトコルを使用して登録済みターゲットにリクエストを送信します。ターゲットグループに各ターゲットを登録するときに、このポートを上書きできます。

トラフィックをターゲットグループ内のターゲットにルーティングするには、リスナーを作成し、リスナーのデフォルトアクションでターゲットグループを指定します。詳細については、「デフォルトアクション」を参照してください。複数のリスナーで同じターゲットグループを指定できますが、これらのリスナーは同じ Network Load Balancer に属している必要があります。ロードバランサーでターゲットグループを使用するには、ターゲットグループが他のロードバランサーのリスナーによって使用されていないことを確認する必要があります。

ターゲットグループのタグはいつでも追加または削除できます。詳細については、「Network Load Balancer のターゲットを登録する」を参照してください。ターゲットグループのヘルスチェック設定を変更することもできます。詳細については、「Network Load Balancer ターゲットグループのヘルスチェック設定を更新する」を参照してください。

要件

- ターゲットグループを作成した後で、ターゲットタイプまたは IP アドレスタイプを変更することはできません。
- ターゲットグループ内のすべてのターゲットは、ターゲットグループと同じ IP アドレスタイプである必要があります (IPv4 または IPv6)。
- デュアルスタックロードバランサーで IPv6 ターゲットグループを使用する必要があります。
- `dualstack` ロードバランサーの UDP リスナーで IPv4 ターゲットグループを使用することはできません。
- QUIC または TCP_QUIC プロトコルで IPv6 ターゲットグループを使用することはできません。

Console

ターゲットグループを作成するには

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、[ターゲットグループ] を選択します。
3. [ターゲットグループの作成] を選択します。
4. [基本設定] ページで、以下を実行します。

   a. [ターゲットタイプを選択] で、インスタンス ID でターゲットを登録する場合は [インスタンス]、IP アドレスでターゲットを登録する場合は [IP アドレス]、Application Load Balancer をターゲットとして登録する場合は [Application Load Balancer] を選択します。

   b. [ターゲットグループ名] に、ターゲットグループの名前を入力します。この名前はリージョンごと、アカウントごとに一意である必要があり、最大 32 文字の英数字またはハイフンのみを使用する必要があり、先頭と末尾にハイフンを使用することはできません。

   c. [Protocol] で、次のようにプロトコルを選択します。

      - リスナープロトコルが TCP の場合は、[TCP] または [TCP_UDP] を選択します。
      - リスナープロトコルが TLS の場合は、[TCP] または [TLS] を選択します。
      - リスナープロトコルが UDP の場合は、 [UDP] または [TCP_UDP] を選択します。
      - リスナープロトコルが TCP_UDP の場合は、[TCP_UDP] を選択します。
      - リスナープロトコルが QUIC の場合は、[QUIC] を選択します。

- リスナープロトコルが TCP_UDP の場合は、[TCP_UDP] を選択します。
- ターゲットタイプが Application Load Balancer の場合、プロトコルは TCP である必要があります。

d. (オプション) [ポート] で、必要に応じてデフォルト値を変更します。

ターゲットタイプが Application Load Balancer の場合、ポートは Application Load Balancer のリスナーポートと一致する必要があります。

e. [IP アドレスタイプ] で、IPv4 または IPv6 を選択します。このオプションは、ターゲットタイプが [インスタンス] または [IP アドレス] の場合にのみ使用できます。

f. [VPC] には、ターゲットを登録する仮想プライベートクラウド (VPC) を選択します。

5. [ヘルスチェック] ペインで、必要に応じてデフォルト設定を変更します。[ヘルスチェックの詳細設定] で、ヘルスチェックポート、カウント、タイムアウト、インターバルを選択し、成功コードを指定します。ヘルスチェックが [異常なしきい値] のカウントを連続して超えると、ロードバランサーはターゲットを停止中の状態にします。ヘルスチェックが [正常なしきい値] のカウントを連続して超えると、ロードバランサーはターゲットを稼働状態に戻します。詳細については、「???」を参照してください。
6. (オプション) タグを追加するには、[タグ] を展開して、[タグを追加] を選択し、タグキーとタグ値を入力します。
7. [次へ] を選択します。
8. (オプション) ターゲットを登録します。ターゲットグループのターゲットタイプによって、指定する情報が決まります。今すぐターゲットを登録する準備ができていない場合は、後で登録できます。
   - インスタンス – EC2 インスタンスを選択し、ポートを入力して、[保留中として以下を含める] を選択します。
   - IP アドレス – IP アドレスまたはその他のプライベート IP アドレスを含む VPC を選択し、IP アドレスとポートを入力して、[保留中として以下を含める] を選択します。
   - Application Load Balancer – Application Load Balancer を選択します。詳細については、「ターゲットとして Application Load Balancer を使用する」を参照してください。
9. [ターゲットグループの作成] を選択します。

AWS CLI

対象グループを作成するには

create-target-group コマンドを使用します。次の例では、TCP プロトコル、IP アドレスで登録されたターゲット、1 つのタグ、デフォルトのヘルスチェック設定を使用してターゲットグループを作成します。

```
aws elbv2 create-target-group \
    --name my-target-group \
    --protocol TCP \
    --port 80 \
    --target-type ip \
    --vpc-id vpc-1234567890abcdef0 \
    --tags Key=department,Value=123
```

ターゲットを登録するには

ターゲットグループを使用してターゲットを登録するには、register-targets コマンドを使用します。例については「the section called “ターゲットの登録”」を参照してください。

CloudFormation

対象グループを作成するには

AWS::ElasticLoadBalancingV2::TargetGroup リソースタイプを定義します。次の例では、TCP プロトコル、IP アドレスによって登録されたターゲット、1 つのタグ、デフォルトのヘルスチェック設定、2 つの登録されたターゲットを含むターゲットグループを作成します。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: TCP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: 10.0.50.10
          Port: 80
        - Id: 10.0.50.20
          Port: 80
```

Network Load Balancer のターゲットグループのヘルス設定を更新する

デフォルトでは、Network Load Balancer はターゲットの状態をモニタリングし、リクエストを正常なターゲットにルーティングします。ただし、ロードバランサーに十分な正常なターゲットがない場合、登録されたすべてのターゲットにトラフィックが自動的に送信されます (フェイルオープン)。ターゲットグループのターゲットグループのヘルス設定を変更して、DNS フェイルオーバーとルーティングフェイルオーバーのしきい値を定義できます。詳細については、「the section called “ターゲットグループの正常性”」を参照してください。

Console

ターゲットグループのヘルス設定を更新するには

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の [ロードバランシング] で [ターゲットグループ] を選択します。
3. ターゲット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. [Attributes] タブで、[Edit] を選択します。
5. [Target group health requirements] (ターゲットグループのヘルス要件) を拡張します。
6. [設定タイプ] には、DNS フェイルオーバーとルーティングフェイルオーバーの両方に同じしきい値を設定する [統合設定] を選択することをお勧めします。
7. [Healthy state requirements] (正常な状態の要件) については、次のいずれかを実行します。
   - [Minimum healthy target count] (正常なターゲットの最小数) を選択し、1 からターゲットグループの最大ターゲット数までの数値を入力します。
   - [Minimum healthy target percentage] (最小の正常なターゲット割合) を選択し、1 から 100 までの数値を入力します。
8. 情報テキストは、ターゲットグループに対してクロスゾーン負荷分散が有効になっているかどうかを示します。クロスゾーン負荷分散が無効になっている場合は、それを有効にして十分な容量を確保できます。[ターゲット選択設定] で、[クロスゾーン負荷分散] を更新します。

   次のテキストは、クロスゾーン負荷分散が無効になっていることを示しています。

   ```
   Healthy state requirements apply to each zone independently.
   ```

次のテキストは、クロスゾーン負荷分散が有効になっていることを示しています。

```
Healthy state requirements apply to the total targets across all applicable
 zones.
```

9. [Save changes] (変更の保存) をクリックします。

AWS CLI

ターゲットグループのヘルス設定を更新するには

modify-target-group-attributes コマンドを使用します。次の例では、両方の異常な状態アクションの正常しきい値を 50% に設定しています。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes \

 "Key=target_group_health.dns_failover.minimum_healthy_targets.percentage,Value=50"
 \

 "Key=target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage,Value=5
```

CloudFormation

ターゲットグループのヘルス設定を変更するには

AWS::ElasticLoadBalancingV2::TargetGroup リソースを更新します。次の例では、両方の異常な状態アクションの正常しきい値を 50% に設定しています。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: TCP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
```

```
        - Key: "target_group_health.dns_failover.minimum_healthy_targets.percentage"
          Value: "50"
        - Key:
 "target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage"
          Value: "50"
```

Network Load Balancer ターゲットグループのヘルスチェック

ターゲットを 1 つ以上のターゲットグループに登録します。登録処理が完了し、ターゲットが最初のヘルスチェックに合格するとすぐに、ロードバランサーは新しく登録したターゲットへのリクエストのルーティングを開始します。登録プロセスが完了し、ヘルスチェックが開始されるまで数分かかることがあります。

Network Load Balancers はアクティブおよびパッシブヘルスチェックを使用して、ターゲットがリクエストを処理できるかどうかを判断します。デフォルトでは、各ロードバランサーノードは、アベイラビリティーゾーン内の登録済みターゲット間でのみリクエストをルーティングします。クロスゾーン負荷分散を有効にすると、各ロードバランサーノードは、有効なすべてのアベイラビリティーゾーンの正常なターゲットにリクエストをルーティングします。詳細については、「クロスゾーンロードバランサー」を参照してください。

パッシブのヘルスチェックでは、ロードバランサーはターゲットの接続への応答状態を確認します。パッシブのヘルスチェックでは、ロードバランサーはアクティブのヘルスチェックで異常が報告される前に異常なターゲットを検出できます。パッシブなヘルスチェックは無効、設定、または監視することはできません。パッシブのヘルスチェックは UDP トラフィックと、維持設定がオンになっているターゲットグループではサポートされていません。詳細については、「スティッキーセッション」を参照してください。

ターゲットが異常になると、ロードバランサーは、ターゲットに関連付けられたクライアント接続で受信したパケットの TCP RST を送信します (異常なターゲットがトリガーしたロードバランサーが起動しなかった場合以外)。

1 つ以上のターゲットグループで、有効にしたアベイラビリティーゾーン内に正常なターゲットがない場合、DNS から該当するサブネットの IP アドレスを削除し、そのアベイラビリティーゾーンのターゲットにリクエストをルーティングできないようにします。有効なすべてのアベイラビリティーゾーン内で、すべてのターゲットが同時にヘルスチェックに失敗すると、ロードバランサーはオープンに失敗します。Network Load Balancer は、空のターゲットグループがある場合にもフェールオープンします。フェールオープンの効果は、ヘルスステータスに関わらず、有効なすべてのアベイラビリティーゾーン内のすべてのターゲットへのトラフィックを許可することです。

ターゲットグループが HTTPS ヘルスチェックで構成されている場合、登録されたターゲットが TLS 1.3 のみをサポートしている場合にはそのターゲットはヘルスチェックに失敗します。これらのターゲットは、TLS 1.2 などの以前のバージョンの TLS をサポートしている必要があります。

HTTP または HTTPS ヘルスチェックリクエストの場合、ホストヘッダーには、ターゲットの IP アドレスおよびヘルスチェックポートではなく、ロードバランサーノードの IP アドレスおよびリスナーポートが含まれます。

TLS リスナーを Network Load Balancer に追加すると、リスナーの接続テストが実行されます。TLS の終了では TCP 接続も終了され、新しい TCP 接続がロードバランサーとターゲット間で確立されます。したがって、TLS リスナーに登録されたターゲットに対してロードバランサーからこのテスト用に送信された TCP 接続が表示される場合があります。これらの TCP 接続は識別できます。Network Load Balancer のソース IP アドレスがあり、接続にデータパケットは含まれていないためです。

UDP および QUIC サービスの場合、ターゲットの可用性は、ターゲットグループの非 UDP ヘルスチェックを使用してテストできます。使用可能なヘルスチェック（TCP、HTTP、または HTTPS）およびターゲット上の任意のポートを使用して、お使いのサービスの可用性を確認できます。ヘルスチェックを受信しているサービスが失敗した場合、ターゲットは使用不可とみなされます。お使いのサービスのヘルスチェックの精度を向上させるには、ヘルスチェックポートをリッスンして UDP または QUIC サービスのステータスを追跡し、サービスが使用できない場合はヘルスチェックが失敗するようにサービスを設定します。

詳細については、「the section called “ターゲットグループの正常性”」を参照してください。

内容

ヘルスチェックの設定

以下の設定を使用して、ターゲットグループのターゲットのアクティブなヘルスチェックを設定します。ヘルスチェックが UnhealthyThresholdCount 連続失敗数のしきい値を超えると、ロードバラン

サーはターゲットをサービス停止中の状態にします。ヘルスチェックが HealthyThresholdCount 連続成功数のしきい値を超えると、ロードバランサーはターゲットを実行中の状態に戻します。

設定	説明	デフォルト
HealthCheckProtocol	ターゲットでヘルスチェックを実行するときにロードバランサーが使用するプロトコル。使用可能なプロトコルは HTTP、HTTPS、および TCP です。デフォルトは TCP プロトコルです。ターゲットタイプが `alb` の場合、サポートされているヘルスチェックプロトコルは HTTP および HTTPS です。	TCP
HealthCheckPort	ターゲットでヘルスチェックを実行するときにロードバランサーが使用するポート。デフォルトでは、各ターゲットがロードバランサーからトラフィックを受信するポートが使用されます。	各ターゲットがロードバランサーからトラフィックを受信するポート。
HealthCheckPath	[HTTP/HTTPS ヘルスチェック] ヘルスチェックのターゲットの送信先であるヘルスチェックパス。デフォルトは / です。	/
HealthCheckTimeoutSeconds	ヘルスチェックを失敗と見なす、ターゲットからレスポンスがない時間 (秒単位)。範囲は 2 ～120 秒です。デフォルト値は、HTTP の場合は 6 秒、TCP および HTTPS ヘルスチェックの場合は 10 秒です。	HTTP ヘルスチェックの場合は 6 秒、TCP および HTTPS ヘルスチェックの場合は 10 秒です。
HealthCheckIntervalSeconds	個々のターゲットのヘルスチェックの概算間隔 (秒単位)。範囲は 5 ～ 300 秒です。デフォルト値は 30 秒です。	30 秒

設定	説明	デフォルト
	Network Load Balancer のヘルスチェックは分散され、コンセンサスメカニズムを使用してターゲットのヘルスを判断します。そのため、ターゲットは設定されているヘルスチェック数よりも多くのヘルスチェックを受けます。HTTP ヘルスチェックを使用している場合にターゲットへの影響を軽減するには、静的 HTML ファイルなどより単純な送信先をターゲットで使用するか、TCP ヘルスチェックに切り替えます。	
HealthyThresholdCount	非正常なインスタンスが正常であると見なすまでに必要なヘルスチェックの連続成功回数。範囲は 2 ~ 10 です。デフォルトは 5 です。	5
UnhealthyThresholdCount	非正常なインスタンスが非正常であると見なすまでに必要なヘルスチェックの連続失敗回数。範囲は 2 ~ 10 です。デフォルトは 2 です。	2
マッチャー	[HTTP/HTTPS ヘルスチェック] ターゲットからの正常なレスポンスを確認するために使用する HTTP コード。範囲は 200 から 599 です。デフォルトは 200～399 です。	200-399

ターゲットヘルスステータス

ロードバランサーがターゲットにヘルスチェックリクエストを送信する前に、ターゲットグループに登録し、リスナールールでターゲットグループを指定して、ターゲットのアベイラビリティーゾーンがロードバランサーに対して有効になっていることを確認する必要があります。

次の表は、登録されたターゲットのヘルスステータスの可能値を示しています。

値	説明
`initial`	ロードバランサーは、ターゲットを登録中か、ターゲットで最初のヘルスチェックを実行中です。 関連する理由コード:`Elb.RegistrationInProgress` |`Elb.InitialHealthChecking`
`healthy`	ターゲットは正常です。 関連する理由コード:なし
`unhealthy`	ターゲットはヘルスチェックに応答しなかったか、ヘルスチェックに合格しなかったか、またはターゲットが停止状態にあります。 関連する理由コード : `Target.FailedHealthChecks`
`draining`	ターゲットは登録解除中で、Connection Draining 中です。 関連する理由コード : `Target.DeregistrationInProgress`
`unhealthy.draining`	ターゲットはヘルスチェックに応答しなかったか、ヘルスチェックに合格しなかったか、または猶予期間に入っています。この猶予期間中は、ターゲットは既存の接続をサポートし、新しい接続は受け入れません。 関連する理由コード : `Target.FailedHealthChecks`
`unavailable`	ターゲットヘルスは使用できません。 関連する理由コード : `Elb.InternalError`
`unused`	ターゲットがターゲットグループに登録されていないか、ターゲットグループがリスナールールで使用され

値	説明
	ていないか、または、有効ではないアベイラビリティーゾーンにターゲットがあります。 関連する理由コード :`Target.NotRegistered` |`Target.NotInUse` |`Target.InvalidState` |`Target.IpUnusable`

ヘルスチェックの理由コード

ターゲットのステータスが `Healthy` 以外の値の場合、API は問題の理由コードと説明を返し、コンソールのツールヒントで同じ説明が表示されます。`Elb` で始まる理由コードはロードバランサー側で発生し、`Target` で始まる理由コードはターゲット側で発生します。

理由コード	説明
`Elb.InitialHealthChecking`	最初のヘルスチェックが進行中です
`Elb.InternalError`	内部エラーのため、ヘルスチェックに失敗しました
`Elb.RegistrationInProgress`	ターゲットの登録中です
`Target.DeregistrationInProgress`	ターゲットの登録解除中です
`Target.FailedHealthChecks`	ヘルスチェックに失敗しました
`Target.InvalidState`	ターゲットが停止状態にあります ターゲットは終了状態にあります ターゲットは終了状態か、または停止状態にあります ターゲットは無効な状態にあります
`Target.IpUnusable`	IP アドレスはロードバランサーによって使用されているので、ターゲットとして使用できません

理由コード	説明
`Target.NotInUse`	ターゲットグループは、ロードバランサーからトラフィックを受信するように設定されていません ロードバランサーが有効になっていないアベイラビリティーゾーンにターゲットがあります
`Target.NotRegistered`	ターゲットはターゲットグループに登録されていません

Network Load Balancer ターゲットのヘルスをチェックする

ターゲットグループに登録されたターゲットのヘルスステータスをチェックできます。ヘルスチェックの失敗に関するヘルプについては、「トラブルシューティング: 登録されたターゲットが実行中でない」を参照してください。

Console

ターゲットのヘルスをチェックするには

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の [ロードバランシング] で [ターゲットグループ] を選択します。
3. ターゲット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. [詳細] ペインには、ターゲットの総数と各ヘルスステータスのターゲット数が表示されます。
5. [Targets] (ターゲット) タブの [Health status] (ヘルスステータス) 列は、各ターゲットのステータスを示します。
6. ターゲットのステータスの値が `Healthy` 以外の場合は、[Health status details] (ヘルスステータスの詳細) 列に詳細情報が表示されます。

異常なターゲットに関する E メール通知を受信するには

CloudWatch アラームを使用して、異常なターゲットに関する詳細を送信する Lambda 関数をトリガーします。ステップバイステップの手順については、ブログ投稿「ロードバランサーの異常なターゲットを特定する」を参照してください。

AWS CLI

ターゲットのヘルスをチェックするには

describe-target-health コマンドを使用します。この例では、出力をフィルタリングして、正常でないターゲットのみを含めます。正常でないターゲットでは、出力に理由コードが含まれます。

```
aws elbv2 describe-target-health \
    --target-group-arn target-group-arn \
    --query "TargetHealthDescriptions[?TargetHealth.State!='healthy'].
[Target.Id,TargetHealth.State,TargetHealth.Reason]" \
    --output table
```

以下は出力の例です。

```
----------------------------------------------
|            DescribeTargetHealth            |
+--------------+---------+-------------------+
|  172.31.0.57 |  unused |  Target.NotInUse  |
|  172.31.0.50 |  unused |  Target.NotInUse  |
+--------------+---------+-------------------+
```

ターゲットの状態と理由コード

次のリストは、各ターゲット状態において考えられる理由コードを表示しています。

ターゲットの状態は healthy です

理由コードが指定されていません。

ターゲットの状態は initial です

- `Elb.RegistrationInProgress` - ターゲットはロードバランサーに登録中です。
- `Elb.InitialHealthChecking` - ロードバランサーは、ヘルスステータスを判断するために必要なヘルスチェックの最小数をターゲットに送信しています。

ターゲットの状態は unhealthy です

- `Target.FailedHealthChecks` - ターゲットへの接続を確立するときにロードバランサーがエラーを受信したか、ターゲットレスポンスの形式が正しくありません。

ターゲットの状態は unused です

- `Target.NotRegistered` - ターゲットはターゲットグループに登録されていません。

- `Target.NotInUse` - ターゲットグループはロードバランサーによって使用されていないか、ターゲットがロードバランサーに対して有効化されていないアベイラビリティーゾーンにあります。
- `Target.InvalidState` - ターゲットは停止状態か終了状態です。
- `Target.IpUnusable` - ターゲット IP アドレスは、ロードバランサーによる使用向けに予約されています。

ターゲットの状態は draining です

- `Target.DeregistrationInProgress` - ターゲットは登録解除中であり、登録解除の遅延期間が終了していません。

ターゲットの状態は unavailable です

- `Elb.InternalError` - 内部エラーのため、ターゲットヘルスを使用できません。

Network Load Balancer ターゲットグループのヘルスチェック設定を更新する

ターゲットグループのヘルスチェック設定は随時変更できます。ヘルスチェック設定のリストについては、「the section called “ヘルスチェックの設定”」を参照してください。

Console

ヘルスチェックの設定を更新するには

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の [ロードバランシング] で [ターゲットグループ] を選択します。
3. ターゲット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. [ヘルスチェック] タブで、[編集] を選択します。
5. [ヘルスチェックの編集を設定] ページで、必要に応じて設定を変更します。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

ヘルスチェックの設定を更新するには

modify-target-group コマンドを使用します。次の例では、HealthyThresholdCount と HealthCheckTimeoutSeconds の設定を更新します。

```
aws elbv2 modify-target-group \
    --target-group-arn target-group-arn \
    --healthy-threshold-count 3 \
    --health-check-timeout-seconds 20
```

CloudFormation

ヘルスチェックの設定を更新するには

AWS::ElasticLoadBalancingV2::TargetGroup リソースを更新して、更新されたヘルスチェック設定を含めます。次の例では、HealthyThresholdCount と HealthCheckTimeoutSeconds の設定を更新します。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: TCP
      Port: 80
      TargetType: instance
      VpcId: !Ref myVPC
      HealthyThresholdCount: 3
      HealthCheckTimeoutSeconds: 20
```

Network Load Balancer のターゲットグループ属性を編集する

Network Load Balancer のターゲットグループを作成したら、そのターゲットグループ属性を編集できます。

ターゲットグループの属性

- クライアント IP の保存
- 登録解除の遅延
- Proxy Protocol
- スティッキーセッション
- ターゲットグループに対するクロスゾーン負荷分散
- 異常のあるターゲットの接続終了

- 異常なドレイニング間隔

クライアント IP の保存

Network Load Balancer は、リクエストをバックエンドターゲットにルーティングするときに、クライアントのソース IP アドレスを保持できます。クライアント IP 保存を無効にした場合、Network Load Balancer のプライベート IP アドレスが送信元 IP アドレスになります。

デフォルトでは、UDP、TCP_UDP、QUIC、TCP_QUIC プロトコルを使用するインスタンスおよび IP タイプのターゲットグループに対して、クライアント IP の保存が有効になっています (無効にすることはできません)。ただし、`preserve_client_ip.enabled` ターゲットグループ属性を使用して、TCP および TLS ターゲットグループのクライアント IP の保存を有効または無効にできます。

デフォルト設定

- インスタンスタイプのターゲットグループ: 有効
- IP タイプのターゲットグループ (UDP、TCP_UDP、QUIC、TCP_QUIC): 有効
- IP タイプのターゲットグループ (TCP、TLS): 無効

クライアント IP 保存が有効になっている場合

次の表は、クライアント IP 保存が有効になっているときにターゲットが受け取る IP アドレスを示しています。

ターゲット	IPv4 クライアントリクエスト	IPv6 クライアントリクエスト
インスタンスタイプ (IPv4)	クライアント IPv4 アドレス	ロードバランサー IPv4 アドレス
IP タイプ (IPv4)	クライアント IPv4 アドレス	ロードバランサー IPv4 アドレス
IP タイプ (IPv6)	ロードバランサー IPv6 アドレス	クライアント IPv6 アドレス

クライアント IP 保存が無効になっている場合

次の表は、クライアント IP 保存が無効になっているときにターゲットが受け取る IP アドレスを示しています。

ターゲット	IPv4 クライアントリクエスト	IPv6 クライアントリクエスト
インスタンスタイプ (IPv4)	ロードバランサー IPv4 アドレス	ロードバランサー IPv4 アドレス
IP タイプ (IPv4)	ロードバランサー IPv4 アドレス	ロードバランサー IPv4 アドレス
IP タイプ (IPv6)	ロードバランサー IPv6 アドレス	ロードバランサー IPv6 アドレス

要件と考慮事項

- クライアント IP 保存の変更は、新しい TCP 接続に対してのみ有効です。
- クライアント IP 保存を有効にした場合、トラフィックは Network Load Balancer からターゲットに直接フローする必要があります。ターゲットは、ロードバランサーと同じ VPC 内、または同じリージョン内のピア接続 VPC 内に配置されている必要があります。
- ターゲットがトランジットゲートウェイを経由して到達される場合、クライアント IP の保存はサポートされません。
- ターゲットが Network Load Balancer と同じ VPC にあっても、ゲートウェイロード バランサーエンドポイントを使用して Network Load Balancer とターゲット (インスタンスまたは IP アドレス) の間のトラフィックを検査する場合、クライアント IP の保持はサポートされません。
- インスタンスタイプが C1、CC1、CC2、CG1、CG2、CR1、G1、G2、HI1、HS1、M1、M2、M3、T1である場合、クライアント IP 保存をサポートしません。クライアント IP 保存を無効にして、これらのインスタンスタイプを IP アドレスとして登録することをお勧めします。
- クライアント IP の保存は、 AWS PrivateLinkからのインバウンドトラフィックには影響しません。 AWS PrivateLink トラフィックの送信元 IP アドレスは、常に Network Load Balancer のプライベート IP アドレスです。
- ターゲットグループに、 AWS PrivateLink ネットワークインターフェイスまたは別のNetwork Load Balancer のネットワークインターフェイスが含まれている場合、クライアント IP の保存はサポートされません。これにより、それらのターゲットとの通信が失われます。

- クライアント IP 保存は、IPv6 から IPv4 に変換されたトラフィックには影響しません。このタイプのトラフィックの送信元 IP アドレスは、常に Network Load Balancer のプライベート IP アドレスです。
- Application Load Balancer タイプでターゲットを指定すると、すべての着信トラフィックのクライアント IP が Network Load Balancer によって保存され、Application Load Balancer に送信されます。次に、Application Load Balancer は、それをターゲットに送信する前にクライアント IP を `X-Forwarded-For` リクエストに追加します。
- NAT ループバック (ヘアピニングとも呼ばれる) は、クライアント IP 保存が有効になっている場合はサポートされません。これは、内部 Network Load Balancer を使用していて、Network Load Balancer の背後に登録されたターゲットが同じ Network Load Balancer への接続を作成する場合に発生します。接続を作成しようとしているターゲットに接続をルーティングして、接続エラーが発生する可能性があります。同じ Network Load Balancer の背後にあるターゲットから Network Load Balancer に接続しないことをお勧めします。または、クライアント IP 保存を無効にすることで、この種の接続エラーを防ぐこともできます。クライアント IP アドレスが必要な場合は、Proxy Protocol v2 を使用して取得できます。詳細については、「Proxy Protocol」を参照してください。
- クライアント IP の保存が無効な場合、Network Load Balancer は一意の各ターゲット (IP アドレスとポート) に対して 55,000 の同時接続または 1 分あたり約 55,000 の接続をサポートします。これらの接続数を超えた場合、ポート割り当てエラーが発生する可能性が高くなり、新しい接続を確立できなくなることがあります。詳細については、「バックエンドフローのポート割り当てエラー」を参照してください。

Console

クライアント IP 保存を変更するには

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の [Load Balancing (ロードバランシング)] で [Target Groups (ターゲットグループ)] を選択します。
3. ターゲット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. [属性] タブで [編集] を選択し、[トラフィック設定] ペインを見つけます。
5. クライアント IP 保存を有効にするには、[Preserve client IP addresses] (クライアント IP アドレスの保持) をオンにします。クライアント IP 保存を無効にするには、[Preserve client IP addresses] (クライアント IP アドレスの保持) をオフにします。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

クライアント IP 保護を有効にするには

preserve_client_ip.enabled 属性を指定して modify-target-group-attributes コマンドを使用します。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=preserve_client_ip.enabled,Value=true"
```

CloudFormation

クライアント IP 保護を有効にするには

AWS::ElasticLoadBalancingV2::TargetGroup リソースを更新して、preserve_client_ip.enabled 属性を含めます。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: TCP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "preserve_client_ip.enabled"
          Value: "true"
```

登録解除の遅延

ターゲットを登録解除すると、ロードバランサーはターゲットへの新しい接続の作成を停止します。ロードバランサーは Connection Draining を使用して、既存の接続での処理中のトラフィックを完了させます。登録解除されたターゲットが正常であり、既存の接続がアイドル状態でない場合、ロードバランサーはそのターゲットのトラフィックの送信を継続することができます。既存の接続が確実に終了されるようにするには、以下を行います。接続終了のターゲットグループ属性を有効にする、インスタンスの登録を解除する前にインスタンスが異常であることを確認する、クライアント接続を定期的に閉じる。

登録解除するターゲットの初期状態は `draining` です。この間、ターゲットは新しい接続の受信を停止します。ただし、設定の伝播の遅延により、ターゲットは引き続き接続を受信する可能性があります。デフォルトでは、ロードバランサーは登録解除するターゲットの状態を 300 秒後に `unused` に変更します。登録解除するターゲットの状態が `unused` に変わるのをロードバランサーが待機する時間の長さを変更するには、登録解除の遅延値を更新します。リクエストを確実に完了するには、120 秒以上の値を指定することをお勧めします。QUIC トラフィックの場合、値は常に 300 秒であり、調整できません。

接続終了のターゲットグループ属性を有効にすると、登録解除されたターゲットへの接続は、登録解除タイムアウトの終了直後に閉じられます。

Console

登録解除遅延属性を変更するには

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の [Load Balancing (ロードバランシング)] で [Target Groups (ターゲットグループ)] を選択します。
3. ターゲット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. [Attributes] タブで、[Edit] を選択します。
5. 登録解除タイムアウトを変更するには、[登録解除の遅延] に新しい値を入力します。ターゲットの登録解除後に既存の接続が閉じられるようにするには、[Terminate connections on deregistration] (登録解除時に接続終了) を選択します。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

登録解除遅延属性を変更するには

`deregistration_delay.timeout_seconds` および `deregistration_delay.connection_termination.enabled` 属性を指定して modify-target-group-attributes コマンドを使用します。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes \
      "Key=deregistration_delay.timeout_seconds,Value=60" \
      "Key=deregistration_delay.connection_termination.enabled,Value=true"
```

CloudFormation

登録解除遅延属性を変更するには

AWS::ElasticLoadBalancingV2::TargetGroup リソースを更新して、deregistration_delay.timeout_seconds および deregistration_delay.connection_termination.enabled 属性を含めます。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: TCP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "deregistration_delay.timeout_seconds"
          Value: "60"
        - Key: "deregistration_delay.connection_termination.enabled"
          Value: "true"
```

Proxy Protocol

Network Load Balancer は、プロキシプロトコルバージョン 2 を使用して、送信元と送信先などの追加の接続情報を送信します。Proxy Protocol バージョン 2 は、Proxy Protocol ヘッダーのバイナリエンコードを提供します。

ロードバランサーは、TCP リスナーを使用して TCP データにプロキシプロトコルヘッダーを付加します。既存のデータは破棄または上書きされません。これには、ネットワークパスのクライアントまたは他のプロキシ、ロードバランサー、またはサーバーによって送信された受信プロキシプロトコルヘッダーが含まれます。したがって、複数のプロキシプロトコルヘッダーを受け取ることができます。また、Network Load Balancer の外部のターゲットへの別のネットワークパスが存在する場合、最初のプロキシプロトコルヘッダーは、ロードバランサーからのものでない可能性があります。

TLS リスナーは、クライアントまたはその他のプロキシから送信されたプロキシプロトコルヘッダーを含む受信接続をサポートしていません。

QUIC トラフィックはプロキシプロトコルバージョン 2 をサポートしていません。

IP アドレスでターゲットを指定すると、アプリケーションに提供される送信元 IP アドレスは、ターゲットグループのプロトコルに応じて次のように異なります。

- TCP と TLS: デフォルトでは、クライアント IP 保存は無効になっており、アプリケーションに提供される送信元 IP アドレスはロードバランサーノードのプライベート IP アドレスです。クライアントの IP アドレスを保存するには、ターゲットが同じ VPC 内またはピア接続 VPC 内にあり、クライアント IP 保存が有効になっていることを確認します。クライアントの IP アドレスが必要で、これらの条件が満たされていない場合は、プロキシプロトコルを有効にし、プロキシプロトコルヘッダーからクライアント IP アドレスを取得します。
- UDP と TCP_UDP: クライアント IP 保存はこれらのプロトコルではデフォルトで有効になっており、無効にすることはできないため、送信元 IP アドレスはクライアントの IP アドレスです。インスタンス ID でターゲットを指定すると、アプリケーションに提供される送信元 IP アドレスは、クライアントの IP アドレスになります。ただし、必要に応じて Proxy Protocol を有効にし、Proxy Protocol ヘッダーからクライアント IP アドレスを取得できます。

ヘルスチェックの接続

Proxy Protocol を有効にした後、Proxy Protocol ヘッダーも、ロードバランサーからのヘルスチェック接続に含まれます。ただし、ヘルスチェック接続では、クライアント接続情報は Proxy Protocol ヘッダーでは送信されません。

ターゲットがプロキシプロトコルヘッダーを解析できない場合、ヘルスチェックに失敗する可能性があります。たとえば、HTTP 400: Bad request というエラーを返す場合があります。

VPC エンドポイントサービス

VPC エンドポイントサービスを通じたサービスコンシューマーからのトラフィックの場合、アプリケーションに提供される送信元の IP アドレスは、ロードバランサーノードのプライベート IP アドレスです。アプリケーションでサービスコンシューマーの IP アドレスが必要な場合は、Proxy Protocol を有効にし、Proxy Protocol ヘッダーからその IP アドレスを取得します。

Proxy Protocol ヘッダーには、エンドポイントの ID も含まれています。この情報は、次のようにカスタム Type-Length-Value (TLV) ベクトルを使用してエンコードされます。

フィールド	長さ (オクテット単位)	説明
タイプ	1	PP2_TYPE_AWS (0xEA)

フィールド	長さ (オクテット単位)	説明
長さ。	2	値の長さ
値	1	PP2_SUBTYPE_AWS_VPCE_ID (0x01)
	変数 (値の長さから 1 を引いた値)	エンドポイントの ID

TLV タイプ 0xEA を解析する例については、https://github.com/aws/elastic-load-balancing-tools/tree/master/proprot を参照してください。

Proxy Protocol の有効化

ターゲットグループで Proxy Protocol を有効にする前に、アプリケーションが Proxy Protocol v2 ヘッダーを予期し、解析できることを確認します。それ以外の場合、アプリケーションは失敗する可能性があります。詳細については、「Proxy Protocol バージョン 1 および 2」を参照してください。

Console

プロキシプロトコルバージョン 2 を有効にするには

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の [Load Balancing (ロードバランシング)] で [Target Groups (ターゲットグループ)] を選択します。
3. ターゲット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. [Attributes] タブで、[Edit] を選択します。
5. [属性の編集] ページで、[プロキシプロトコル v2] を選択します。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

プロキシプロトコルバージョン 2 を有効にするには

`proxy_protocol_v2.enabled` 属性を指定して modify-target-group-attributes コマンドを使用します。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
```

```
    --attributes "Key=proxy_protocol_v2.enabled,Value=true"
```

CloudFormation

プロキシプロトコルバージョン 2 を有効にするには

AWS::ElasticLoadBalancingV2::TargetGroup リソースを更新して、proxy_protocol_v2.enabled 属性を含めます。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: TCP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "proxy_protocol_v2.enabled"
          Value: "true"
```

スティッキーセッション

スティッキーセッションは、クライアントトラフィックをターゲットグループ内の同じターゲットにルーティングするためのメカニズムです。これは、クライアントに連続したエクスペリエンスを提供するために状態情報を維持するサーバーに役立ちます。

考慮事項

- スティッキーセッションを使用すると、接続とフローの分散が不均一になり、ターゲットの可用性に影響する場合があります。たとえば、同じ NAT デバイスの背後にあるすべてのクライアントの送信元 IP アドレスは同じです。したがって、これらのクライアントからのすべてのトラフィックは、同じターゲットにルーティングされます。
- いずれかのターゲットのヘルス状態が変更されたり、ターゲットグループに対してターゲットを登録または登録解除したりすると、ロードバランサーによってターゲットグループのスティッキーセッションがリセットされる場合があります。
- ターゲットグループに対して維持属性が有効になっている場合、パッシブヘルスチェックはサポートされません。詳細については、「ターゲットグループのヘルスチェック」を参照してください。

- スティッキーセッションは、TLS または QUIC リスナーでサポートされません。

Console

スティッキーセッションを有効にするには

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の [Load Balancing (ロードバランシング)] で [Target Groups (ターゲットグループ)] を選択します。
3. ターゲット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. [Attributes] タブで、[Edit] を選択します。
5. [Target selection configuration] (ターゲット選択設定) で、[Stickiness] (スティッキネス) をオンにします。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

スティッキーセッションを有効にするには

`stickiness.enabled` 属性を指定して modify-target-group-attributes コマンドを使用します。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=stickiness.enabled,Value=true"
```

CloudFormation

スティッキーセッションを有効にするには

AWS::ElasticLoadBalancingV2::TargetGroup リソースを更新して、`stickiness.enabled` 属性を含めます。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: TCP
```

```
        Port: 80
        TargetType: ip
        VpcId: !Ref myVPC
        TargetGroupAttributes:
          - Key: "stickiness.enabled"
            Value: "true"
```

ターゲットグループに対するクロスゾーン負荷分散

ロードバランサーのノードは、クライアントからのリクエストを登録済みターゲットに分散させます。クロスゾーンロードバランサーがオンの場合、各ロードバランサーノードは、すべての登録済みアベイラビリティーゾーンの登録済みターゲットにトラフィックを分散します。クロスゾーンロードバランサーがオフの場合、各ロードバランサーノードは、そのアベイラビリティーゾーンの登録済みターゲットのみにトラフィックを分散します。これは、ゾーンの障害ドメインがリージョナルドメインよりも優先される場合に使用できます。これにより、正常なゾーンが異常なゾーンの影響を受けないようにしたり、全体的なレイテンシーを改善したりすることができます。

Network Load Balancer では、クロスゾーンロードバランサーは、ロードバランサーレベルでのデフォルトで無効になっていますが、いつでも有効にすることができます。ターゲットグループの場合、デフォルトではロードバランサー設定を使用しますが、ターゲットグループレベルでクロスゾーンロードバランサーを明示的に有効または無効にすることでデフォルトを上書きできます。

考慮事項

- Network Load Balancer のクロスゾーン負荷分散を有効にする場合、EC2 データ転送料金が適用されます。詳細については、「AWS Data Exports ユーザーガイド」の「データ転送料金について」を参照してください。
- ターゲットグループ設定によって、ターゲットグループのロードバランサー動作が決まります。たとえば、クロスゾーンロードバランサーがロードバランサーレベルで有効で、ターゲットグループレベルで無効になっている場合、ターゲットグループに送信されるトラフィックはアベイラビリティーゾーン間でルーティングされません。
- クロスゾーンロードバランサーが無効の場合は、各ゾーンが関連するワークロードを処理できるように、各ロードバランサーのアベイラビリティーゾーンに十分なターゲット容量があることを確認してください。
- クロスゾーンロードバランサーが無効になっている場合は、すべてのターゲットグループが同じアベイラビリティーゾーンの参加になっていることを確認してください。空のアベイラビリティーゾーンは異常であるとみなされます。

- ターゲットグループタイプが instance または ip の場合、ターゲットグループレベルでクロスゾーンロードバランシングを有効または無効にすることができます。ターゲットグループタイプが alb の場合、ターゲットグループは常にロードバランサーからクロスゾーンロードバランシング設定を継承します。

ロードバランサーレベルのクロスゾーンロードバランシングの有効化について詳しくは、「the section called “クロスゾーンロードバランサー”」を参照してください。

Console

ターゲットグループのクロスゾーンロードバランサーを有効にするには

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の [Load Balancing] (ロードバランサー) で [Target Groups] (ターゲットグループ) を選択します。
3. ターゲット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. [属性] タブで、[編集] を選択します。
5. [Edit target group attributes] (ターゲットグループ属性の編集) ページで、[Cross-zone load balancing] (クロスゾーンロードバランサー) で [On] (オン) を選択します。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

ターゲットグループのクロスゾーンロードバランサーを有効にするには

load_balancing.cross_zone.enabled 属性を指定して modify-target-group-attributes コマンドを使用します。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes "Key=load_balancing.cross_zone.enabled,Value=true"
```

CloudFormation

ターゲットグループのクロスゾーンロードバランサーを有効にするには

AWS::ElasticLoadBalancingV2::TargetGroup リソースを更新して、load_balancing.cross_zone.enabled 属性を含めます。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: TCP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "load_balancing.cross_zone.enabled"
          Value: "true"
```

異常のあるターゲットの接続終了

接続の終了はデフォルトで有効になっています。Network Load Balancer のターゲットが設定されたヘルスチェックに失敗し、正常でないと見なされると、ロードバランサーは確立された接続を終了し、ターゲットへの新しい接続のルーティングを停止します。接続終了を無効にしても、ターゲットは異常と見なされて新しい接続を受信しませんが、確立された接続はアクティブなままなので、正常に閉じることができます。

異常なターゲットの接続終了は、ターゲットグループレベルで設定されます。

Console

接続終了属性を変更するには

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の [ロードバランシング] で [ターゲットグループ] を選択します。
3. ターゲット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. [Attributes] タブで、[Edit] を選択します。
5. [Target unhealthy state management] の下で、[Terminate connections when targets become unhealthy] を有効にするか無効にするかを選択します。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

接続終了属性を無効にするには

target_health_state.unhealthy.connection_termination.enabled 属性を指定して modify-target-group-attributes コマンドを使用します。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes
 "Key=target_health_state.unhealthy.connection_termination.enabled,Value=false"
```

CloudFormation

接続終了属性を無効にするには

AWS::ElasticLoadBalancingV2::TargetGroup リソースを更新して、target_health_state.unhealthy.connection_termination.enabled 属性を含めます。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: TCP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "target_health_state.unhealthy.connection_termination.enabled"
          Value: "false"
```

異常なドレイニング間隔

unhealthy.draining 状態のターゲットは異常と見なされ、新しい接続を受信しませんが、設定された間隔の間は確立された接続が保持されます。異常な接続間隔によって、ターゲットが unhealthy 状態になるまで unhealthy.draining 状態のまま維持する時間が決まります。異常な接続間隔の間にターゲットがヘルスチェックに合格すると、その状態は再び healthy になります。登録解除がトリガーされると、ターゲットの状態が draining になり、登録解除遅延タイムアウトが開始されます。

要件

異常なドレイニング間隔を有効にする前に、接続の終了を無効にする必要があります。

Console

異常なドレイニング間隔を変更するには

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の [ロードバランシング] で [ターゲットグループ] を選択します。
3. ターゲット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. [Attributes] タブで、[Edit] を選択します。
5. [Target unhealthy state management] (ターゲットの異常状態の管理) で、[Terminate connections when targets become unhealthy] (ターゲットが異常になったら接続を終了する) がオフになっていることを確認します。
6. [異常なドレイニング間隔] の値を入力します。
7. [Save changes] (変更の保存) をクリックします。

AWS CLI

異常なドレイニング間隔を変更するには

`target_health_state.unhealthy.draining_interval_seconds` 属性を指定して modify-target-group-attributes コマンドを使用します。

```
aws elbv2 modify-target-group-attributes \
    --target-group-arn target-group-arn \
    --attributes
 "Key=target_health_state.unhealthy.draining_interval_seconds,Value=60"
```

CloudFormation

異常なドレイニング間隔を変更するには

AWS::ElasticLoadBalancingV2::TargetGroup リソースを更新して、`target_health_state.unhealthy.draining_interval_seconds` 属性を含めます。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: TCP
      Port: 80
```

```
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "target_health_state.unhealthy.draining_interval_seconds"
          Value: "60"
```

Network Load Balancer のターゲットを登録する

ターゲットがリクエストを処理する準備ができたら、そのターゲットを 1 つ以上のターゲットグループに登録します。ターゲットグループのターゲットタイプにより、ターゲットを登録する方法が決定されます。たとえば、インスタンス ID、IP アドレス、または Application Load Balancer を登録できます。登録処理が完了し、ターゲットが最初のヘルスチェックに合格すると、Network Load Balancer はすぐにターゲットへのリクエストのルーティングを開始します。登録プロセスが完了し、ヘルスチェックが開始されるまで数分かかることがあります。詳細については、「Network Load Balancer ターゲットグループのヘルスチェック」を参照してください。

現在登録されているターゲットの需要が上昇した場合、需要に対応するために追加ターゲットを登録できます。登録されたターゲットの需要が減少した場合は、ターゲットグループからターゲットの登録を解除できます。登録解除プロセスが完了し、ロードバランサーがターゲットへのリクエストのルーティングを停止するまで数分かかることがあります。その後需要が増加した場合は、登録解除したターゲットをターゲットグループに再度登録できます。ターゲットをサービスする必要がある場合は、そのターゲットを登録解除し、サービスの完了時に再度登録できます。

ターゲットを登録解除すると、Elastic Load Balancing は未処理のリクエストが完了するまで待機します。これは、Connection Drainingと呼ばれます。Connection Drainingの進行中、ターゲットのステータスは `draining` です。登録解除が完了すると、ターゲットのステータスは `unused` に変わります。詳細については、「登録解除の遅延」を参照してください。

インスタンス ID でターゲットを登録する場合は、Auto Scaling グループでロードバランサーを使用できます。Auto Scaling グループにターゲットグループをアタッチし、そのグループがスケールアウトすると、Auto Scaling グループによって起動されたインスタンスが自動的にターゲットグループに登録されます。Auto Scaling グループからロードバランサーをデタッチした場合、インスタンスはターゲットグループから自動的に登録解除されます。詳細については、「Amazon EC2 Auto Scaling ユーザーガイド」の「Auto Scaling グループへのロードバランサーのアタッチ」を参照してください。

内容

- [ネットワーク ACL]
- [共有サブネット]
- [ターゲットの登録]
- [ターゲットの登録解除]

ターゲットセキュリティグループ

ターゲットグループにターゲットを追加する前に、ターゲットに関連するセキュリティグループを Network Load Balancer からのトラフィックを受け入れるように設定します。

ロードバランサーにセキュリティグループが関連付けられている場合のターゲットセキュリティグループに関する推奨事項

- クライアントトラフィックを許可するには: ロードバランサーに関連付けられたセキュリティグループを参照するルールを追加します。
- PrivateLink トラフィックを許可するには: 経由で送信されたトラフィックのインバウンドルールを評価するようにロードバランサーを設定した場合は AWS PrivateLink、トラフィックポートのロードバランサーセキュリティグループからのトラフィックを受け入れるルールを追加します。それ以外の場合は、トラフィックポートのロードバランサーのプライベート IP アドレスからのトラフィックを受け入れるルールを追加します。
- ロードバランサーのヘルスチェックを受け入れるには: ヘルスチェックポートのロードバランサーセキュリティグループからのヘルスチェックトラフィックを受け入れるルールを追加します。

ロードバランサーがセキュリティグループに関連付けられていない場合のターゲットセキュリティグループの推奨事項

- クライアントトラフィックを許可するには: ロードバランサーがクライアント IP アドレスを保持している場合は、承認されたクライアントの IP アドレスからのトラフィックをトラフィックポートで受け付けるルールを追加します。それ以外の場合は、トラフィックポートのロードバランサーのプライベート IP アドレスからのトラフィックを受け入れるルールを追加します。
- プライベートリンクのトラフィックを許可するには: トラフィックポートのロードバランサーのプライベート IP アドレスからのトラフィックを受け入れるルールを追加します。
- ロードバランサーのヘルスチェックを受け入れるには: ヘルスチェックポートのロードバランサーのプライベート IP アドレスからのヘルスチェックトラフィックを受け入れるルールを追加します。

クライアント IP 保存の仕組み

`preserve_client_ip.enabled` 属性を `true` に設定しない限り、Network Load Balancer はクライアント IP アドレスを保持しません。また、デュアルスタックの Network Load Balancer を使用すると、IPv4 アドレスを IPv6 に、または IPv6 アドレスを IPv4 に変換する場合、クライアント IP アドレス保存は機能しません。クライアント IP アドレス保存は、クライアント IP アドレスとターゲット IP アドレスの両方が IPv4 または IPv6 である場合にのみ機能します。

コンソールを使用してロードバランサーのプライベート IP アドレスを見つけるには

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、[ネットワークインターフェイス] を選択します。
3. 検索フィールドに、Network Load Balancer の名前を入力します。ロードバランサーのサブネットあたり 1 つのネットワークインターフェイスがあります。
4. 各ネットワークインターフェイスの [詳細] タブで、[プライベート IPv4 アドレス] からアドレスをコピーします。

詳細については、「Network Load Balancer のセキュリティグループを更新する」を参照してください。

ネットワーク ACL

EC2 インスタンスをターゲットとして登録する場合は、インスタンスのサブネットのネットワーク ACL をチェックして、リスナーポートとヘルスチェックポートの両方でトラフィックを許可していることを確認する必要があります。VPC のデフォルトネットワークアクセスコントロールリスト (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が許可されます。カスタムネットワーク ACL を作成する場合は、適切なトラフィックを許可していることを確認してください。

インスタンスのサブネットに関連付けられているネットワーク ACL では、インターネット向けロードバランサーの次のトラフィックを許可する必要があります。

インスタンスサブネットの推奨ルール

Inbound

送信元	プロトコル	ポート範囲	コメント

###### IP ####	####	########	Allow client traffic (IP Preservation: ON)
VPC CIDR	####	########	Allow client traffic (IP Preservation: VOFF)
VPC CIDR	#######	#######	Allow health check traffic

Outbound

送信先	プロトコル	ポート範囲	コメント
###### IP ####	####	1024-65535	Allow return traffic to client (IP Preservat ion: ON)
VPC CIDR	####	1024-65535	Allow return traffic to client (IP Preservat ion: VOFF)
VPC CIDR	#######	1024-65535	Allow health check traffic

ロードバランサーのサブネットに関連付けられているネットワーク ACL では、インターネット向けロードバランサーの次のトラフィックを許可する必要があります。

ロードバランサーサブネットの推奨ルール

Inbound

送信元	プロトコル	ポート範囲	コメント
###### IP ####	####	####	Allow client traffic
VPC CIDR	####	1024-65535	Allow response from target
VPC CIDR	#######	1024-65535	Allow health check traffic

Outbound

送信先	プロトコル	ポート範囲	コメント
###### IP ####	*####*	1024-65535	Allow responses to clients
VPC CIDR	*####*	*########*	Allow requests to targets
VPC CIDR	*#######*	*#######*	Allow health check to targets

内部ロードバランサーの場合、インスタンスおよびロードバランサーノードのサブネットのネットワーク ACL は、リスナーポートおよび一時ポートにおいて、VPC CIDR とやり取りされる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する必要があります。

共有サブネット

参加者は共有 VPC に Network Load Balancer を作成できます。参加者は、自分と共有されていないサブネットで実行するターゲットを登録することはできません。

Network Load Balancer の共有サブネットは、以下を除くすべての AWS リージョンでサポートされています。

- アジアパシフィック (大阪) `ap-northeast-3`
- アジアパシフィック (香港) `ap-east-1`
- 中東 (バーレーン) `me-south-1`
- AWS 中国 (北京) `cn-north-1`
- AWS 中国 (寧夏) `cn-northwest-1`

ターゲットの登録

各ターゲットグループでは、ロードバランサーが有効になっている各アベイラビリティーゾーンで少なくとも 1 つのターゲットが登録されている必要があります。

ターゲットグループのターゲットタイプにより、登録できるターゲットが決定されます。詳細については、「[Target type (ターゲットタイプ)]」を参照してください。以下の情報を使用して、タイプ

`instance` または `ip` のターゲットグループにターゲットを登録します。ターゲットタイプが `alb` の場合は、「ターゲットとして Application Load Balancer を使用する」を参照してください。

要件と考慮事項

- インスタンスの登録時の状態は `running` である必要があります。
- インスタンスで使用されているインスタンスタイプが C1、CC1、CC2、CG1、CG2、CR1、G1、G2、HI1、HS1、M1、M2、M3、T1 のいずれかである場合、インスタンス ID でインスタンスを登録することはできません。
- インスタンス ID でターゲットを登録する場合、インスタンスは Network Load Balancer と同じ VPC にある必要があります。ロードバランサー VPC (同じリージョンまたは異なるリージョン) とピア接続されている VPC にインスタンスがある場合、そのインスタンスをインスタンス ID で登録することはできません。このようなインスタンスは IP アドレスで登録できます。
- IPv6 ターゲットグループにインスタンス ID でターゲットを登録する場合、ターゲットにはプライマリ IPv6 アドレスが割り当てられている必要があります。詳細については、「Amazon EC2 ユーザーガイド」の「IPv6 アドレス」を参照してください。
- IPv4 ターゲットグループの IP アドレスでターゲットを登録する場合、登録する IP アドレスは次のいずれかの CIDR ブロックからのものである必要があります。
  - ターゲットグループの VPC のサブネット
  - 10.0.0.0/8 (RFC 1918)
  - 100.64.0.0/10 (RFC 6598)
  - 172.16.0.0/12 (RFC 1918)
  - 192.168.0.0/16 (RFC 1918)
- IPv6 ターゲットグループの IP アドレスでターゲットを登録する場合、登録する IP アドレスは VPC IPv6 CIDR ブロック内またはピア接続された VPC の IPv6 CIDR ブロック内にある必要があります。
- ターゲットを IP アドレスで登録し、その IP アドレスがロードバランサーと同じ VPC にある場合、ロードバランサーは、到達可能なサブネットからターゲットがアクセスしていることを確認します。
- UDP、TCP_UDP、QUIC、TCP_QUIC ターゲットグループの場合、インスタンスがロードバランサー VPC の外部に存在するか、インスタンスタイプとして C1、CC1、CC2、CG1、CG2、CR1、G1、G2、HI1、HS1、M1、M2、M3、T1 のいずれかを使用しているときは、IP アドレスでインスタンスを登録しないでください。ロードバランサー VPC の外部に存在するか、サポートされていないインスタンスタイプを使用するターゲットは、ロードバランサーからのトラフィックを受信できても、応答できない場合があります。

QUIC 固有の要件と考慮事項

- QUIC または TCP_QUIC ターゲットグループに登録されたすべてのターゲットには、サーバー ID を指定する必要があります。
- サーバー ID は、Network Load Balancer リスナー内に存在するすべてのターゲットに対して一意である必要があります。
- QUIC サーバー ID は常に 8 バイトです。ターゲットを登録する場合、サーバー ID は `0x` 形式で、16 進数文字が続く必要があります。
- ターゲットがサーバー ID に登録されると、その ID はイミュータブルになります。ターゲットサーバー ID を変更するには、最初に登録を解除してから、新しいサーバー ID に登録する必要があります。
- ターゲット識別子とポートの組み合わせには、1 つのサーバー ID が必要です。同じ VPC 内の同じ IP またはインスタンス ID とポートの組み合わせに別のサーバー ID を使用することはサポートされていません。
- 6 時間以内に別のターゲットに同じサーバー ID を再使用しないでください。

Console

ターゲットを登録するには

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の [Load Balancing (ロードバランシング)] で [Target Groups (ターゲットグループ)] を選択します。
3. ターゲット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. [Targets] タブを選択します。
5. [Register targets] を選択します。
6. ターゲットグループのターゲットタイプが `instance` の場合、使用可能なインスタンスを選択し、必要に応じてデフォルトのポートを上書きしてから、[保留中として以下を含める] を選択します。
7. ターゲットグループのターゲットタイプが `ip` の場合は、各 IP アドレスでネットワークを選択し、IP アドレスとポートを入力して、[保留中として以下を含める] を選択します。
8. ターゲットグループのターゲットタイプが `alb` の場合、必要に応じてデフォルトのポートを上書きし、Application Load Balancer を選択します。詳細については、「ターゲットとして Application Load Balancer を使用する」を参照してください。

9. ターゲットグループのプロトコルが QUIC または TCP_QUIC の場合は、サーバー ID が指定されていることを確認します。
10. [保留中のターゲットを登録] を選択します。

AWS CLI

ターゲットを登録するには

register-targets コマンドを使用します。次の例では、インスタンス ID でターゲットを登録します。ポートが指定されていないため、ロードバランサーはターゲットグループポートを使用します。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

次の例では、IP アドレスでターゲットを登録します。ポートが指定されていないため、ロードバランサーはターゲットグループポートを使用します。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=10.0.50.10 Id=10.0.50.20
```

次の例では、Application Load Balancer をターゲットとして登録します。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=application-load-balancer-arn
```

次の例では、QUIC または TCP_QUIC ターゲットグループにターゲットを登録します。

```
aws elbv2 register-targets \
    --target-group-arn target-group-arn \
    --targets Id=10.0.50.10,QuicServerId=0xa1b2c3d4e5f65890
 Id=10.0.50.20,QuicServerId=0xa1b2c3d4e5f65999
```

CloudFormation

ターゲットを登録するには

AWS::ElasticLoadBalancingV2::TargetGroup リソースを更新して、新しいターゲットを含めます。次の例では、インスタンス ID で 2 つのターゲットを登録します。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: instance
      VpcId: !Ref myVPC
      Targets:
        - Id: !GetAtt Instance1.InstanceId
          Port: 80
        - Id: !GetAtt Instance2.InstanceId
          Port: 80
```

次の例では、インスタンス ID で 2 つのターゲットを QUIC または TCP_QUIC プロトコルターゲットグループに登録します。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: instance
      VpcId: !Ref myVPC
      Targets:
        - Id: !GetAtt Instance1.InstanceId
          Port: 80
          QuicServerId: 0xa1b2c3d4e5f65999
        - Id: !GetAtt Instance2.InstanceId
          Port: 80
          QuicServerId: 0xa1b2c3d4e5f65000
```

ターゲットの登録解除

アプリケーションの需要が低下した場合や、ターゲットを保守する必要がある場合、ターゲットグループからターゲットを登録解除することができます。ターゲットを登録解除するとターゲットグループから削除されますが、ターゲットにそれ以外の影響は及びません。登録解除するとすぐに、ロードバランサーはターゲットへのトラフィックのルーティングを停止します。ターゲットは、未処理のリクエストが完了するまで `draining` 状態になります。

Console

ターゲットの登録を解除するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [Load Balancing (ロードバランシング)] で [Target Groups (ターゲットグループ)] を選択します。
3. ターゲット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. [ターゲット] タブで、削除するターゲットを選択します。
5. [登録解除] を選択します。

AWS CLI

ターゲットの登録を解除するには

[deregister-targets](https://docs.aws.amazon.com/cli/latest/reference/elbv2/deregister-targets.html) コマンドを使用します。次の例では、インスタンス ID で登録された 2 つのターゲットを登録解除します。

```
aws elbv2 deregister-targets \
    --target-group-arn target-group-arn \
    --targets Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

Network Load Balancer のターゲットとして Application Load Balancer を使用する

1 つの Application Load Balancer を含むターゲットグループをターゲットとして作成し、そのグループにトラフィックを転送するように Network Load Balancer を設定できます。このシナリオでは、トラフィックがターゲットに到達するとすぐに、Application Load Balancer がロードバランシングの決

定を引き継ぎます。この設定では、両方のロードバランサーの機能が組み合わされて以下のような利点が生まれます。

- Application Load Balancer のレイヤー 7 リクエストベースのルーティング機能をエンドポイントサービス (AWS PrivateLink) や静的 IP アドレスなど、Network Load Balancer がサポートする機能と組み合わせて使用できます。
- この構成は、シグナリングに HTTP を使用するメディアサービスや、コンテンツをストリーミングするための RTP など、マルチプロトコルに 1 つのエンドポイントを必要とするアプリケーションに使用できます。

この機能は、内部またはインターネット向けの Network Load Balancer のターゲットとしての内部またはインターネット向けの Application Load Balancer とともに使用できます。

考慮事項

- ターゲットグループごとに登録できる Application Load Balancer は 1 つだけです。
- Application Load Balancer を Network Load Balancer のターゲットとして関連付けるには、ロードバランサーが同じアカウント内の同じ VPC に存在する必要があります。
- 1 つの Application Load Balancer は、最大 2 つの Network Load Balancer のターゲットとして関連付けることができます。これを行うには、各 Network Load Balancer について、Application Load Balancer を個別のターゲットグループに登録します。
- Network Load Balancer に登録した各 Application Load Balancer によって、Network Load Balancer ごとにアベイラビリティーゾーンのターゲットの最大数が 50 減少します。両方のロードバランサーのクロスゾーンロードバランシングを無効にして、レイテンシーを最小限に抑え、リージョン内データ転送の料金を回避できます。詳細については、「Network Load Balancer のクォータ」を参照してください。
- ターゲットグループタイプが `alb` の場合、ターゲットグループの属性を変更することはできません。これらの属性は常にデフォルト値を使用します。
- Application Load Balancer をターゲットとして登録すると、すべてのターゲットグループから登録を解除するまで Application Load Balancer を削除することはできません。
- Network Load Balancer と Application Load Balancer 間の通信は常に IPv4 を使用します。

タスク

- 前提条件
- ステップ 1: alb タイプのターゲットグループを作成する

- ステップ 2: Network Load Balancer を作成し、ルーティングを設定する
- ステップ 3: (オプション) VPC エンドポイントサービスを作成する

前提条件

ターゲットとして使用する Application Load Balancer がまだない場合は、ロードバランサー、リスナー、およびそのターゲットグループを作成します。詳細については、「Application Load Balancer ユーザーガイド」の「Application Load Balancer の作成」を参照してください。

ステップ 1: alb タイプのターゲットグループを作成する

ステップ 1: `alb` タイプのターゲットグループを作成します。Application Load Balancer は、ターゲットグループの作成時以降にターゲットとして登録できます。

Console

ターゲットとして Application Load Balancer のターゲットグループを作成するには

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の [ロードバランシング] で [ターゲットグループ] を選択します。
3. [ターゲットグループの作成] を選択します。
4. [基本設定] ペインの [ターゲットタイプを選択] で、Application Load Balancer を選択します。
5. [ターゲットグループ名] に、ターゲットグループの名前を入力します。
6. [Protocol] (プロトコル) では TCP だけが選択できます。ターゲットグループのポートを選択します。このターゲットグループのポートは、Application Load Balancer のリスナーポートと一致する必要があります。このターゲットグループに別のポートを選択した場合は、Application Load Balancer のリスナーポートを更新して一致させることができます。
7. [VPC] には、ターゲットグループの仮想プライベートクラウド (VPC) を選択します。これは、Application Load Balancer で使用されるのと同じ VPC である必要があります。
8. [Health checks] (ヘルスチェック) で、[Health check protocol] (ヘルスチェックプロトコル) として [HTTP] または [HTTPS] を選択します。ヘルスチェックは Application Load Balancer に送信され、指定されたポート、プロトコル、および ping パスを使用してターゲットに転送されます。ヘルスチェックのポートとプロトコルに一致するポートとプロトコルがあるリスナーが Application Load Balancer にあり、これらのヘルスチェックを受信できることを確認します。

9. (オプション) タグ を展開します。追加するタグごとに、[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。
10. [次へ] を選択します。
11. Application Load Balancer を登録する準備ができたら、[今すぐ登録] を選択し、必要に応じてデフォルトポートを上書きして、Application Load Balancer を選択します。Application Load Balancer には、ターゲットグループと同じポート上のリスナーが必要です。このロードバランサーのリスナーを追加または編集してターゲットグループのポートと一致させるか、前のステップに戻ってターゲットグループのポートを変更することができます。

    Application Load Balancer をターゲットとして登録する準備ができていない場合は、[後で登録] を選択し、後でターゲットを登録します。詳細については、「the section called “ターゲットの登録”」を参照してください。
12. [ターゲットグループの作成] を選択します。

AWS CLI

alb タイプの対象グループを作成するには

create-target-group コマンドを使用します。プロトコルは TCP で、ポートは Application Load Balancer のリスナーポートと一致する必要があります。

```
aws elbv2 create-target-group \
    --name my-target-group \
    --protocol TCP \
    --port 80 \
    --target-type alb \
    --vpc-id vpc-1234567890abcdef0 \
    --tags Key=department,Value=123
```

CloudFormation

タイプ alb のターゲットグループを作成するには

AWS::ElasticLoadBalancingV2::TargetGroup リソースタイプを定義します。プロトコルは TCP で、ポートは Application Load Balancer のリスナーポートと一致する必要があります。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
```

```
    Properties:
      Name: my-target-group
      Protocol: TCP
      Port: 80
      TargetType: alb
      VpcId: !Ref myVPC
      Tags:
        - Key: 'department'
          Value: '123'
      Targets:
        - Id: !Ref myApplicationLoadBalancer
          Port: 80
```

ステップ 2: Network Load Balancer を作成し、ルーティングを設定する

Network Load Balancer を作成するときに、Application Load Balancer にトラフィックを転送するようにデフォルトのアクションを設定できます。

Console

Network Load Balancer を作成するには

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の [ロードバランシング] で [ロードバランサー] を選択します。
3. [ロードバランサーを作成] を選択します。
4. [Network Load Balancer] で、[Create] (作成) を選択します。
5. 基本的な設定
    a. [ロードバランサー名] に、Network Load Balancer の名前を入力します。
    b. [スキーム] で、[インターネット向け] または [内部] を選択します。インターネット向け Network Load Balancer は、クライアントからインターネット経由でリクエストをターゲットにルーティングします。内部 Network Load Balancer は、プライベート IP アドレスを使用してターゲットにリクエストをルーティングします。
    c. [ロードバランサーの IP アドレスタイプ] については、クライアントが Network Load Balancer との通信に IPv4 アドレスを使用する場合は [IPv4] を、クライアントが Network Load Balancer との通信に IPv4 アドレスと IPv6 アドレスの両方を使用する場合は [デュアルスタック] を選択します。
6. ネットワークマッピング

a. [VPC] で、Application Load Balancer ターゲットに使用したのと同じ VPC を選択します。インターネット向けロードバランサーでは、インターネットゲートウェイを持つ VPC のみを選択できます。

b. アベイラビリティーゾーンとサブネットの場合は、少なくとも 1 つのアベイラビリティーゾーンを選択し、ゾーンごとに 1 つのサブネットを選択します。Application Load Balancer で有効になっているのと同じアベイラビリティーゾーンを選択することをお勧めします。これにより、可用性、スケーリング、パフォーマンスが最適化されます。

   (オプション) 静的 IP アドレスを使用するには、各アベイラビリティーゾーンの [IPv4 settings] (IPv4 の設定) で [Use an Elastic IP address] (Elastic IP アドレスを使用する) を選択します。静的 IP アドレスを使用すると、ファイアウォールの許可リストに特定の IP アドレスを追加することや、クライアントで IP アドレスをハードコードすることができます。

7. セキュリティグループ

   ロードバランサー VPC のデフォルトのセキュリティグループを AWS が事前に選択します。必要に応じて、追加のセキュリティグループを選択できます。適切なセキュリティグループがない場合は、[新しいセキュリティグループを作成]を選択して今すぐ新しいセキュリティグループを作成します。詳細については、「Amazon VPC ユーザーガイド」の「セキュリティグループの作成」を参照してください。

   > **Warning**
   >
   > この時点で Network Load Balancer にセキュリティグループを関連付けていない場合、後で関連付けすることはできません。

   > **Warning**
   >
   > QUIC または TCP_QUIC リスナーを活用するには、Network Load Balancer にセキュリティグループがあってはなりません。

8. リスナーとルーティング

   a. デフォルトは、ポート 80 で TCP トラフィックを受け付けるリスナーです。トラフィックを Application Load Balancer ターゲットグループに転送できるのは TCP リスナーだ

けです。[プロトコル] は [TCP] のままにしておく必要がありますが、[ポート は必要に応じて変更できます。

この構成では、Application Load Balancer で HTTPS リスナーを使用して TLS トラフィックを終了できます。

b. [デフォルトアクション] で、前のステップで作成したターゲットグループを選択します。

c. (オプション) [リスナータグを追加] をクリックし、タグキーとタグ値を入力します。

9. ロードバランサータグ

   (オプション) [ロードバランサータグ] を展開します。(オプション) [新しいタグを追加] をクリックし、タグキーとタグ値を入力します。詳細については、「タグ」を参照してください。

10. [概要]

   設定を確認し、[ロードバランサーを作成] を選択します。

AWS CLI

Network Load Balancer を作成するには

create-load-balancer コマンドを使用します。Application Load Balancer で有効になっているのと同じアベイラビリティーゾーンを使用することをお勧めします。

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --type network \
    --scheme internal \
    --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
    --security-groups sg-1111222233334444
```

TCP リスナーを追加するには

TCP リスナーを作成するには、create-listener コマンドを使用します。トラフィックを Application Load Balancer に転送できるのは TCP リスナーだけです。デフォルトのアクションには、前のステップで作成したターゲットグループを使用します。

```
aws elbv2 create-listener \
    --load-balancer-arn load-balancer-arn \
```

```
    --protocol TCP \
    --port 80 \
    --default-actions Type=forward,TargetGroupArn=target-group-arn
```

CloudFormation

Network Load Balancer を作成するには

AWS::ElasticLoadBalancingV2::LoadBalancer タイプのリソースと、AWS::ElasticLoadBalancingV2::Listener タイプのリソースを定義します。トラフィックを Application Load Balancer に転送できるのは TCP リスナーだけです。デフォルトのアクションには、前のステップで作成したターゲットグループを使用します。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-load-balancer
      Type: network
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup

  myTCPListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: TCP
      Port: 80
      DefaultActions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
```

ステップ 3: (オプション) VPC エンドポイントサービスを作成する

前のステップで設定した Network Load Balancer をプライベート接続のエンドポイントとして使用するために、 AWS PrivateLinkを有効にすることができます。これにより、ロードバランサーへのプライベート接続がエンドポイントサービスとして確立されます。

Network Load Balancer を使用して VPC エンドポイントサービスを作成するには

1. ナビゲーションペインで、[ロードバランサー] を選択します。
2. Network Load Balancer の名前を選択して、その詳細ページを開きます。
3. [Integrations] (統合) タブで、[VPC エンドポイントサービス (AWS PrivateLink)] を展開します。
4. [エンドポイントサービスの作成] を選択して、[エンドポイントサービス] ページを開きます。残りのステップについては、「AWS PrivateLink ガイド」の「エンドポイントサービスの作成」を参照してください。

Network Load Balancer のターゲットグループにタグを付ける

タグを使用すると、ターゲットグループを目的、所有者、環境などさまざまな方法で分類することができます。

各ターゲットグループに対して複数のタグを追加できます。タグキーは、各ターゲットグループで一意である必要があります。すでにターゲットグループに関連付けられているキーを持つタグを追加すると、そのキーの値が更新されます。

不要になったタグは、削除することができます。

制限事項

- リソースあたりのタグの最大数 – 50
- キーの最大長 – 127 文字 (Unicode)
- 値の最大長 – 255 文字 (Unicode)
- タグのキーと値は大文字と小文字が区別されます。使用できる文字は、UTF-8 で表現できる文字、スペース、および数字と、特殊文字 (+、-、=、.、_、:、/、@) です。ただし、先頭または末尾にはスペースを使用しないでください。
- タグ名または値に aws: プレフィックスを使用しないでください。このプレフィックスは AWS 使用のために予約されています。このプレフィックスが含まれるタグの名前または値は編集または削除できません。このプレフィックスを持つタグは、リソースあたりのタグ数の制限時には計算されません。

Console

ターゲットグループのタグを管理するには

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の [Load Balancing (ロードバランシング)] で [Target Groups (ターゲットグループ)] を選択します。
3. ターゲット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. [タグ] タブで、[タグの管理] を選択し、次の 1 つ以上の操作を行います。
   a. タグを更新するには、[キー] と [値] に新しい値を入力します。
   b. タグを追加するには、[タグの追加] を選択し、[キー] と [値] に値を入力します。
   c. タグを削除するには、タグの横にある [削除] を選択します。
5. [Save changes] (変更の保存) をクリックします。

AWS CLI

タグを追加するには

add-tags コマンドを使用します。次の例では、2 つのタグを追加します。

```
aws elbv2 add-tags \
    --resource-arns target-group-arn \
    --tags "Key=project,value=lima" "Key=department,Value=digital-media"
```

タグを削除するには

remove-tags コマンドを使用します。次の例では、指定したキーを使用してタグを削除します。

```
aws elbv2 remove-tags \
    --resource-arns target-group-arn \
    --tag-keys project department
```

CloudFormation

タグを追加するには

AWS::ElasticLoadBalancingV2::TargetGroup リソースを更新して、Tagsプロパティを含めます。

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: TCP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      Tags:
        - Key: 'project'
          Value: 'lima'
        - Key: 'department'
          Value: 'digital-media'
```

Network Load Balancer のターゲットグループを削除する

ターゲットグループがリスナールールの転送アクションによって参照されていない場合は、これを削除できます。ターゲットグループを削除しても、ターゲットグループに登録されたターゲットには影響が及びません。登録済み EC2 インスタンスが必要なくなった場合は停止または終了できます。

Console

ターゲットグループを削除するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ナビゲーションペインの [ロードバランシング] で [ターゲットグループ] を選択します。
3. ターゲットグループを選択し、[Actions]、[Delete] を選択します。
4. [削除] を選択します。

AWS CLI

ターゲットグループを削除するには

delete-target-group コマンドを使用します。

```
aws elbv2 delete-target-group \
    --target-group-arn target-group-arn
```

Network Load Balancer を監視する

次の機能を使用して、ロードバランサーの監視、トラフィックパターンの分析、ロードバランサーとターゲットに関する問題の解決を実行できます。

CloudWatch メトリクス

Amazon CloudWatch を使用して、ロードバランサーとターゲットのデータポイント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。これらのメトリクスを使用して、システムが正常に実行されていることを確認できます。詳細については、「Network Load Balancer の CloudWatch メトリクス」を参照してください。

VPC フローログ

VPC フローログを使用して、Network Load Balancer との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。詳細については、Amazon VPC ユーザーガイドの VPC フローログを参照してください。

ロードバランサーの各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。ロードバランサーのサブネットあたり 1 つのネットワークインターフェイスがあります。Network Load Balancer のネットワークインターフェイスを特定するには、ネットワークインターフェイスの説明フィールドでロードバランサーの名前を探します。

Network Load Balancer を通じて、各接続に 2 つのエントリがあります。1 つはクライアントとロードバランサー間のフロントエンド接続で、もう 1 つはロードバランサーとターゲットとの間のバックエンド接続です。ターゲットグループのクライアント IP 保存属性が有効な場合、接続はクライアントからの接続としてインスタンスに表示されます。それ以外の場合、接続のソース IP はロードバランサーのプライベート IP アドレスです。インスタンスのセキュリティグループで、クライアントからの接続が許可されないが、ロードバランサーサブネットのネットワーク ACL で許可される場合、ロードバランサーのネットワークインターフェイスのログにはフロントエンドおよびバックエンド接続に対して「ACCEPT OK」と表示され、インスタンスのネットワークインターフェイスのログには接続に対して「REJECT OK」と表示されます。

Network Load Balancer にセキュリティグループが関連付けられている場合、フローログには、セキュリティグループによって許可または拒否されたトラフィックのエントリが含まれます。Network Load Balancer に TLS リスナーを使用すると、フローログエントリには拒否されたエントリのみが反映されます。

Amazon CloudWatch Internet Monitor

Internet Monitor を使用すると、インターネットの問題が でホストされているアプリケーション AWS とエンドユーザー間のパフォーマンスと可用性にどのように影響するかを可視化できます。また、他の サービスの使用に切り替えるか、異なる を介してトラフィックをワークロードに再ルーティングすることで、アプリケーションの予測レイテンシーをほぼリアルタイムで改善する方法を調べることもできます AWS リージョン。詳細については、「Amazon CloudWatch Internet Monitor の使用」を参照してください。

アクセスログ

アクセスログを使用して、ロードバランサーに送信される TLS リクエストについて、詳細情報を収集できます。ログファイルは Amazon S3 に保存されます。これらのアクセスログを使用して、トラフィックパターンの分析や、ターゲットの問題のトラブルシューティングを行うことができます。詳細については、「Network Load Balancer のアクセスログ」を参照してください。

CloudTrail ログ

AWS CloudTrail を使用して、Elastic Load Balancing API に対する呼び出しに関する詳細情報をキャプチャし、ログファイルとして Amazon S3 に保存できます。これらの CloudTrail ログを使用して、行われた呼び出し、呼び出し元のソース IP アドレス、呼び出し元、呼び出し時間などを判断できます。詳細については、「CloudTrail を使用した Elastic Load Balancing の API コールのログ記録」を参照してください。

Network Load Balancer の CloudWatch メトリクス

Elastic Load Balancing は、ロードバランサーとターゲットのデータポイントを Amazon CloudWatch に発行します。CloudWatch では、それらのデータポイントについての統計を、(メトリクスと呼ばれる) 順序付けられた時系列データ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、データポイント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。たとえば、指定した期間中のロードバランサーの正常なターゲットの合計数を監視することができます。各データポイントには、タイムスタンプと、オプションの測定単位が関連付けられています。

メトリクスを使用して、システムが正常に実行されていることを確認できます。例えば、メトリクスが許容範囲外になる場合、CloudWatch アラームを作成して、指定されたメトリクスを監視し、アクション (E メールアドレスに通知を送信するなど) を開始することができます。

Elastic Load Balancing は、ロードバランサー経由でリクエストが伝達される場合にのみ、メトリクスを CloudWatch にレポートします。ロードバランサーを経由するリクエストがある場合、Elastic Load Balancing は 60 秒間隔でメトリクスを測定し、送信します。ロードバランサーを経由するリク

エストがないか、メトリクスのデータがない場合、メトリクスは報告されません。セキュリティグループが関連付けられた Network Load Balancer の場合、セキュリティグループによって拒否されたトラフィックは CloudWatch メトリックスにキャプチャされません。

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。

目次

Network Load Balancer メトリクス

AWS/NetworkELB 名前空間には、次のメトリクスが含まれます。

メトリクス	説明
ActiveFlowCount	クライアントからターゲットへの同時フロー (または接続) の合計数。このメトリクスには、SYN_SENT 状態と ESTABLISHED 状態の接続が含まれます。TCP 接続はロードバランサーで終了しないため、ターゲットへの TCP 接続を開いているクライアントは単一のフローとしてカウントされます。 レポート条件: 常に報告される。 統計値: 最も有用な統計値は Average、Maximum、および Minimum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup
ActiveFlowCount_TCP	クライアントからターゲットへの同時 TCP フロー (または接続) の合計数。このメトリクスには、SYN_SENT 状態と ESTABLISHED 状

メトリクス	説明
	態の接続が含まれます。TCP 接続はロードバランサーで終了しないため、ターゲットへの TCP 接続を開いているクライアントは単一のフローとしてカウントされます。 レポート条件: ゼロ以外の値がある 統計値: 最も有用な統計値は `Average`、`Maximum`、および `Minimum` です。 ディメンション • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer` • `TargetGroup`
`ActiveFlowCount_TLS`	クライアントからターゲットへの同時 TLS フロー (または接続) の合計数。このメトリクスには、SYN_SENT 状態と ESTABLISHED 状態の接続が含まれます。 レポート条件: ゼロ以外の値がある。 統計値: 最も有用な統計値は `Average`、`Maximum`、および `Minimum` です。 ディメンション • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer` • `TargetGroup`

メトリクス	説明
ActiveFlowCount_UDP	クライアントからターゲットへの同時 UDP フロー (または接続) の合計数。 レポート条件: ゼロ以外の値がある。 統計値: 最も有用な統計値は Average、Maximum、および Minimum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup
ActiveZonalShiftHostCount	現在ゾーンシフトにアクティブに参加しているターゲットの数。 レポート条件: ロードバランサーがゾーンシフトにオプトインしている場合に報告されます。 [統計値]: 最も有用な統計値は Maximum および Minimum です。 ディメンション • LoadBalancer , TargetGroup • AvailabilityZone , LoadBalancer , TargetGroup
ClientTLSNegotiationErrorCount	クライアントと TLS リスナー間でネゴシエーション中に失敗した TLS ハンドシェイクの合計数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer

メトリクス	説明
`ConsumedLCUs`	ロードバランサーが使用するロードバランサーキャパシティーユニット (LCU) の数です。1 時間当たりで使用する LCU 数の料金をお支払いいただきます。詳細については、[Elastic Load Balancing の料金表](#)を参照してください。 レポート条件: 常に報告される。 統計: All ディメンション • `LoadBalancer`
`ConsumedLCUs_TCP`	TCP のロードバランサーが使用するロードバランサーキャパシティーユニット (LCU) の数です。1 時間当たりで使用する LCU 数の料金をお支払いいただきます。詳細については、[Elastic Load Balancing の料金表](#)を参照してください。 レポート条件: ゼロ以外の値がある。 統計: All ディメンション • `LoadBalancer`
`ConsumedLCUs_TLS`	TLS のロードバランサーが使用するロードバランサーキャパシティーユニット (LCU) の数です。1 時間当たりで使用する LCU 数の料金をお支払いいただきます。詳細については、[Elastic Load Balancing の料金表](#)を参照してください。 レポート条件: ゼロ以外の値がある。 統計: All ディメンション • `LoadBalancer`

メトリクス	説明
ConsumedLCUs_UDP	UDP のロードバランサーが使用するロードバランサーキャパシティーユニット (LCU) の数です。1 時間当たりで使用する LCU 数の料金をお支払いいただきます。詳細については、Elastic Load Balancing の料金表を参照してください。 レポート条件: ゼロ以外の値がある。 統計: All ディメンション • LoadBalancer
HealthyHostCount	正常と見なされるターゲットの数。このメトリックには、ターゲットとして登録されている Application Load Balancer は含まれません。 レポート条件: 登録されたターゲットがある場合に報告されます。 統計値: 最も有用な統計値は Maximum および Minimum です。 ディメンション • LoadBalancer , TargetGroup • AvailabilityZone , LoadBalancer , TargetGroup
NewFlowCount	期間内にクライアントからターゲットに確立された新しいフロー (または接続) の合計数。 レポート条件: 常に報告される。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup

メトリクス	説明
NewFlowCount_TCP	期間内にクライアントからターゲットに確立された新しい TCP フロー (または接続) の合計数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup
NewFlowCount_TLS	期間内にクライアントからターゲットに確立された新しい TLS フロー (または接続) の合計数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup

メトリクス	説明
NewFlowCount_UDP	期間内にクライアントからターゲットに確立された新しい UDP フロー (または接続) の合計数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup
NewFlowCount_QUIC	期間中にルーティング決定を必要とした UDP データグラムの合計数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
PeakBytesPerSecond	サンプリングウィンドウの間に 10 秒間隔で計算される最大バイトの平均値 (1 秒あたりの処理バイト数)。このメトリクスには、ヘルスチェックトラフィックは含まれません。 レポート条件: 常に報告される 統計: 最も有用な統計は Maximum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer

メトリクス	説明
PeakPacketsPerSecond	サンプリングウィンドウの間に 10 秒間隔で計算される最大パケットレートの平均値 (1 秒あたりの処理パケット数) 。このメトリクスには、ヘルスチェックトラフィックが含まれます。 レポート条件: 常に報告される。 統計: 最も有用な統計は `Maximum` です。 ディメンション • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`
PortAllocationErrorCount	クライアント IP 変換操作中の一時ポート割り当てエラーの総数。0 以外の値は切断されたクライアント接続を示します。 注: Network Load Balancer は一意の各ターゲット (IP アドレスとポート) に対して、クライアントアドレス変換を実行するときに 55,000 の同時接続または 1 分あたり約 55,000 の接続をサポートします。ポート割り当てエラーを修正するには、ターゲットグループにさらに多くのターゲットを追加します。 レポート条件: 常に報告される。 統計: 最も有用な統計は `Sum` です。 ディメンション • `LoadBalancer` • `AvailabilityZone` , `LoadBalancer`

メトリクス	説明
ProcessedBytes	TCP/IP ヘッダーを含む、ロードバランサーによって処理された合計バイト数。この数には、ターゲットとの間のトラフィックからヘルスチェックトラフィックを引いたものが含まれます。 レポート条件: 常に報告される。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
ProcessedBytes_TCP	TCP リスナーによって処理される総バイト数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
ProcessedBytes_TLS	TLS リスナーによって処理される総バイト数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer

メトリクス	説明
ProcessedBytes_UDP	UDP リスナーによって処理される総バイト数。 レポート条件: ゼロ以外の値がある 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
ProcessedBytes_QUIC	QUIC リスナーによって処理される総バイト数。 レポート条件: ゼロ以外の値がある 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
ProcessedPackets	ロードバランサーによって処理される総バイト数。この数には、ヘルスチェックトラフィックを含む、ターゲットとの間のトラフィックが含まれます。 レポート条件: 常に報告される。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer

メトリクス	説明
RejectedFlowCount	ロードバランサーによって拒否されたフロー (または接続) の合計数。 レポート条件: 常に報告される。 統計値: 最も有用な統計値は Average、Maximum、および Minimum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
RejectedFlowCount_TCP	ロードバランサーによって拒否された TCP フロー (または接続) の数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
ReservedLCUs	ロードバランサーキャパシティユニット (LCU) 予約を使用するロードバランサー用に予約された LCU の数。 レポート条件: ゼロ以外の値がある 統計: All ディメンション • LoadBalancer

メトリクス	説明
SecurityGroupBlockedFlowCount_Inbound_ICMP	ロードバランサーセキュリティグループのインバウンドルールによって拒否された新しい ICMP メッセージの数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
SecurityGroupBlockedFlowCount_Inbound_TCP	ロードバランサーセキュリティグループのインバウンドルールによって拒否された新しい TCP フローの数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
SecurityGroupBlockedFlowCount_Inbound_UDP	ロードバランサーセキュリティグループのインバウンドルールによって拒否された新しい UDP フローの数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer

メトリクス	説明
SecurityGroupBlockedFlowCount_Outbound_ICMP	ロードバランサーセキュリティグループのアウトバウンドルールによって拒否された新しい ICMP メッセージの数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
SecurityGroupBlockedFlowCount_Outbound_TCP	ロードバランサーセキュリティグループのアウトバウンドルールによって拒否された新しい TCP フローの数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
SecurityGroupBlockedFlowCount_Outbound_UDP	ロードバランサーセキュリティグループのアウトバウンドルールによって拒否された新しい UDP フローの数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer

メトリクス	説明
TargetTLSNegotiationErrorCount	TLS リスナーとターゲット間でネゴシエーション中に失敗した TLS ハンドシェイクの合計数。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer
TCP_Client_Reset_Count	クライアントからターゲットに送信されたリセット (RST) パケットの合計数。これらのリセットは、クライアントによって生成され、ロードバランサーによって転送されます。 レポート条件: 常に報告される。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
TCP_ELB_Reset_Count	ロードバランサーによって生成されたリセット (RST) パケットの合計数。詳細については、「トラブルシューティング」を参照してください。 レポート条件: 常に報告される。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer

メトリクス	説明
TCP_Target_Reset_Count	ターゲットからクライアントに送信されたリセット (RST) パケットの合計数。これらのリセットは、ターゲットによって生成され、ロードバランサーによって転送されます。 レポート条件: 常に報告される。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
UnHealthyHostCount	異常とみなされるターゲットの数。このメトリックには、ターゲットとして登録されている Application Load Balancer は含まれません。 レポート条件: 登録されたターゲットがある場合に報告されます。 統計値: 最も有用な統計値は Maximum および Minimum です。 ディメンション • LoadBalancer , TargetGroup • AvailabilityZone , LoadBalancer , TargetGroup
UnhealthyRoutingFlowCount	ルーティングフェイルオーバーアクション (フェイルオープン) を使用してルーティングされたフロー (または接続) の数。このメトリクスは TLS リスナーではサポートされていません。 レポート条件: ゼロ以外の値がある。 統計: 最も有用な統計は Sum です。

メトリクス	説明
ZonalHealthStatus	ロードバランサーが正常と見なすアベイラビリティーゾーンの数。ロードバランサーは、正常なアベイラビリティーゾーンごとに 1、異常なアベイラビリティーゾーンごとに 0 を出力します。 レポート条件: ヘルスチェックが有効になっている場合にレポートされます。 統計値: 最も有用な統計値は Maximum および Minimum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
QUIC_Unknown_Server_ID_Packet_Drop_Count	Network Load Balancer のターゲットに関連付けられていないサーバー ID を含む、ドロップされた UDP データグラムの数。 報告基準: QUIC リスナーに対してのみ報告。 統計: 最も有用な統計は Sum です。 ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer

Network Load Balancer のメトリクスディメンション

ロードバランサーのメトリクスを絞り込むには、次のディメンションを使用できます。

ディメンション	説明
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。

ディメンション	説明
`LoadBalancer`	ロードバランサーでメトリクスデータをフィルタリングします。ロードバランサーを次のように指定します。net/ロードバランサー名/1234567890123456 (ロードバランサー ARN の最後の部分)。
`TargetGroup`	ターゲットグループでメトリクスデータをフィルタリングします。ターゲットグループを次のように指定します。targetgroup/ターゲットグループ名/1234567890123456 (ターゲットグループ ARN の最後の部分)。

Network Load Balancer メトリクスの統計

CloudWatch では、Elastic Load Balancing で発行されたメトリクスのデータポイントに基づいた統計が提供されます。統計とは、メトリクスデータを指定した期間で集約したものです。統計を要求した場合、返されるデータストリームはメトリクス名とディメンションによって識別されます。ディメンションは、メトリクスを一意に識別する名前/値のペアです。たとえば、特定のアベイラビリティーゾーンで起動されたロードバランサーの配下のすべての正常な EC2 インスタンスの統計をリクエストできます。

`Minimum` および `Maximum` の統計は、各サンプリングウィンドウの個別のロードバランサーノードから報告されるデータポイントの最小値と最大値を反映します。`HealthyHostCount` の最大値の増加は、`UnHealthyHostCount` の最小値の減少に対応します。最大値 `HealthyHostCount` を監視して、最大値 `HealthyHostCount` が必要最小値を下回ったとき、または `0` になったときにアラームを起動することをお勧めします。これは、ターゲットがいつ異常になったかを特定するのに役立ちます。また、最小値 `UnHealthyHostCount` を監視して、最小値 `UnHealthyHostCount` が `0` を上回ったときにアラームを起動することもお勧めします。これにより、登録されたターゲットが存在しなくなったことに気付くことができます。

`Sum` 統計は、すべてのロードバランサーノードにおける集計値です。メトリクスには期間あたり複数のレポートが含まれているため、`Sum` はすべてのロードバランサーノードで集計されたメトリクスのみに適用されます。

`SampleCount` 統計は測定されたサンプルの数です。メトリクスはサンプリング間隔とイベントに基づいて集計されるため、通常、この統計は有用ではありません。たとえば、`HealthyHostCount` の `SampleCount` は、正常なホストの数ではなく各ロードバランサーノードが報告するサンプル数に基づいています。

ロードバランサーの CloudWatch メトリクスの表示

Amazon EC2 コンソールを使用して、ロードバランサーに関する CloudWatch メトリクスを表示できます。これらのメトリクスは、モニタリング用のグラフのように表示されます。ロードバランサーがアクティブでリクエストを受信しているときにのみ、モニタリング用のグラフにデータポイントが表示されます。

別の方法としては、ロードバランサーのメトリクスの表示に、CloudWatch コンソールを使用することもできます。

コンソールを使用してメトリクスを表示するには

1. Amazon EC2 コンソールの https://console.aws.amazon.com/ec2/ を開いてください。
2. ターゲットグループによってフィルタリングされたメトリクスを表示するには、以下の作業を実行します。
   a. ナビゲーションペインで、[Target Groups] を選択します。
   b. ターゲットグループを選択し、[Monitoring] を選択します。
   c. (オプション) 結果を時間でフィルタリングするには、[Showing data for] から時間範囲を選択します。
   d. 1 つのメトリクスの大きいビューを取得するには、グラフを選択します。
3. ロードバランサーでフィルタリングされたメトリクスを表示するには、以下の操作を実行します。
   a. ナビゲーションペインで、[Load Balancers] を選択します。
   b. ロードバランサーを選択し、[Monitoring] タブを選択します。
   c. (オプション) 結果を時間でフィルタリングするには、[Showing data for] から時間範囲を選択します。
   d. 1 つのメトリクスの大きいビューを取得するには、グラフを選択します。

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 (メトリクス)] を選択してください。
3. [NetworkELB] 名前空間を選択します。

4. (オプション) すべてのディメンションでメトリクスを表示するには、検索フィールドに名称を入力します。

を使用してメトリクスを表示するには AWS CLI

使用可能なメトリクスを表示するには、次の list-metrics コマンドを使用します。

```
aws cloudwatch list-metrics --namespace AWS/NetworkELB
```

を使用してメトリクスの統計を取得するには AWS CLI

get-metric-statistics コマンドを使用して、指定されたメトリクスとディメンションの統計情報を取得します。CloudWatch は、ディメンションの一意の組み合わせをそれぞれ別のメトリクスとして扱うことに注意してください。特に発行されていないディメンションの組み合わせを使用した統計を取得することはできません。メトリクス作成時に使用した同じディメンションを指定する必要があります。

```
aws cloudwatch get-metric-statistics --namespace AWS/NetworkELB \
--metric-name UnHealthyHostCount --statistics Average  --period 3600 \
--dimensions Name=LoadBalancer,Value=net/my-load-balancer/50dc6c495c0c9188 \
Name=TargetGroup,Value=targetgroup/my-targets/73e2d6bc24d8a067 \
--start-time 2017-04-18T00:00:00Z --end-time 2017-04-21T00:00:00Z
```

出力例を次に示します。

```
{
    "Datapoints": [
        {
            "Timestamp": "2017-04-18T22:00:00Z",
            "Average": 0.0,
            "Unit": "Count"
        },
        {
            "Timestamp": "2017-04-18T04:00:00Z",
            "Average": 0.0,
            "Unit": "Count"
        },
        ...
    ],
    "Label": "UnHealthyHostCount"
}
```

Network Load Balancer のアクセスログ

Elastic Load Balancing は、Network Load Balancer に対して確立された TLS 接続について、詳細情報を収集するアクセスログを提供します。これらのアクセスログを使用して、トラフィックパターンを分析し、問題のトラブルシューティングを行えます。

> **Important**
>
> 従来の「レガシー」アクセスログ (このセクションで説明) は引き続き使用できますが、Network Load Balancer は CloudWatch Logs を通じた拡張ログ記録オプションの提供を開始しました。CloudWatch Logs は、Amazon CloudWatch Logs、Amazon Data Firehose、Amazon Simple Storage Service などへの、より柔軟な配信オプションを提供します。これらの改善されたログ記録オプションを設定するには、ロードバランサーの [統合] タブにアクセスしてください。CloudWatch Logs の詳細については、「Network Load Balancer の CloudWatch ログ」を参照してください。

> **Important**
>
> アクセスログが作成されるのは、ロードバランサーに TLS リスナーがあり、TLS リクエストに関する情報のみが含まれる場合のみです。アクセスログは、ベストエフォートベースでリクエストを記録します。アクセスログは、すべてのリクエストを完全に報告するためのものではなく、リクエストの本質を把握するものとして使用することをお勧めします。

アクセスログの作成は、Elastic Load Balancing のオプション機能であり、デフォルトでは無効化されています。ロードバランサーのアクセスログの作成を有効にすると、Elastic Load Balancing はログを圧縮ファイルとしてキャプチャし、指定した Amazon S3 バケット内に保存します。アクセスログの作成はいつでも無効にできます。

Amazon S3 が管理する暗号化キー (SSE-S3) によって、または S3 バケットのカスタマーマネージドキーを使用する Key Management Service (SSE-KMS CMK) を使用して、サーバー側の暗号化を有効にできます。各アクセスログファイルは S3 バケットに保存される前に自動的に暗号化され、アクセス時に復号化されます。暗号化あるいは復号化されたログファイルにアクセスする方法に違いがないため、特別なアクションを実行する必要はありません。各ログファイルは、一意のキーで暗号化されます。この一意のキー自体が、定期的に更新される KMS キーで更新されます。詳細については、Amazon S3暗号化 (SSE-S3) の指定」およびAWS KMS 「 (SSE-KMS) を使用したサーバー側の暗号化の指定」を参照してください。 Amazon S3

アクセスログに対する追加料金はありません。Amazon S3 のストレージコストは発生しますが、Amazon S3 にログファイルを送信するために Elastic Load Balancing が使用する帯域については料金は発生しません。ストレージコストの詳細については、Amazon S3 の料金を参照してください。

目次

アクセスログファイル

Elastic Load Balancing は各ロードバランサーノードのログファイルを 5 分ごとに発行します。ログ配信には結果整合性があります。ロードバランサーでは、同じ期間について複数のログが発行されることがあります。これは通常、サイトに高トラフィックがある場合に発生します。

アクセスログのファイル名には次の形式を使用します。

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/aws-account-id_elasticloadbalancing_region_net.load-balancer-id_end-time_random-string.log.gz
```

bucket (バケット)

S3 バケットの名前。

prefix

バケットのプレフィックス (論理階層)。プレフィックスを指定しない場合、ログはバケットのルートレベルに配置されます。

aws-account-id

所有者の AWS アカウント ID。

region

ロードバランサーおよび S3 バケットのリージョン。

yyyy/mm/dd

ログが配信された日付。

load-balancer-id

ロードバランサーのリソース ID。リソース ID にスラッシュ (/) が含まれている場合、ピリオド (.) に置換されます。

end-time

ログ作成の間隔が終了した日時。たとえば、終了時間 20181220T2340Z には、23:35～23:40 に行われたリクエストのエントリが含まれます。

random-string

システムによって生成されたランダム文字列。

ログファイル名の例は次のようになります。

```
s3://my-bucket/prefix/AWSLogs/123456789012/elasticloadbalancing/us-
east-2/2020/05/01/123456789012_elasticloadbalancing_us-east-2_net.my-
loadbalancer.1234567890abcdef_20200501T0000Z_20sg8hgm.log.gz
```

必要な場合はログファイルを自身のバケットに保管できますが、ログファイルを自動的にアーカイブまたは削除するように Amazon S3 ライフサイクルルールを定義することもできます。詳細については、Amazon S3 ユーザーガイドの「ストレージのライフサイクルの管理」を参照してください。

アクセスログのエントリ

次の表は、アクセスログのエントリのフィールドを順に示しています。すべてのフィールドはスペースで区切られています。新しいフィールドが導入されると、ログエントリの最後に追加されます。ログファイルの処理中に、予期していなかったログエントリの最後のフィールドは無視する必要があります。

フィールド	説明
type	リスナーの種類。サポートされる値は `tls` です。
バージョン	ログエントリのバージョン。現在のバージョンは 2.0 です。
time	TLS 接続の最後に記録された時間 (ISO 8601 形式)。

フィールド	説明
elb	ロードバランサーのリソース ID。
リスナー	接続の TLS リスナーのリソース ID。
client_port	クライアントの IP アドレスとポート。
destination_port	送信先の IP アドレスとポート。クライアントがロードバランサーに直接接続する場合、送信先はリスナーです。クライアントが VPC エンドポイントサービスを介して接続する場合、送信先は VPC エンドポイントです。
connection_time	接続が完了するまでの合計時間 (開始から終了まで) (ミリ秒単位)。
tls_handshake_time	TCP 接続が確立された後に TLS ハンドシェイクが完了するまでの合計時間 (クライアント側の遅延時間を含む) (ミリ秒単位)。この時間は `connection_time` フィールドに含まれます。TLS ハンドシェイクまたは TLS ハンドシェイクの失敗がない場合、この値は - に設定されます。
received_bytes	クライアントからロードバランサーによって受信されたバイト数 (復号後)。
sent_bytes	ロードバランサーからクライアントに送信されたバイト数 (復号前)。
incoming_tls_alert	クライアントからロードバランサーによって受信された TLS アラートの整数値 (存在する場合)。それ以外の場合、この値は - に設定されます。
chosen_cert_arn	クライアントに提供された証明書の ARN。有効なクライアント hello メッセージが送信されない場合、この値は - に設定されます。
chosen_cert_serial	将来の利用のために予約されています。この値は常に - に設定されます。
tls_cipher	クライアントとネゴシエートされた暗号スイート (OpenSSL 形式)。TLS ネゴシエーションが完了しない場合、この値は - に設定されます。

フィールド	説明
tls_protocol_version	クライアントとネゴシエートされた TLS プロトコル (文字列形式)。指定できる値は、`tlsv10`、`tlsv11`、`tlsv12`、`tlsv13` です。TLS ネゴシエーションが完了しない場合、この値は - に設定されます。
tls_keyexchange	TLS または PQ-TLS のハンドシェイク中に使用されるキー交換。TLS または PQ-TLS ネゴシエーションが完了しない場合、この値は に設定されます-。
domain_name	クライアント hello メッセージの server_name 拡張機能の値。この値は URL でエンコードされます。有効なクライアント hello メッセージが送信されない場合、または拡張機能が存在しない場合、この値は - に設定されます。
alpn_fe_protocol	クライアントとネゴシエートされたアプリケーションプロトコル (文字列形式)。指定できる値は、`h2`、`http/1.1`、および `http/1.0` です。TLS リスナーで ALPN ポリシーが設定されていない場合、一致するプロトコルが見つからない場合、または有効なプロトコルリストが送信されない場合、この値は - に設定されます。
alpn_be_protocol	ターゲットとネゴシエートされたアプリケーションプロトコル (文字列形式)。指定できる値は、`h2`、`http/1.1`、および `http/1.0` です。TLS リスナーで ALPN ポリシーが設定されていない場合、一致するプロトコルが見つからない場合、または有効なプロトコルリストが送信されない場合、この値は - に設定されます。
alpn_client_preference_list	クライアントの hello メッセージ内の application_layer_protocol_negotiation 拡張機能の値。この値は URL でエンコードされます。各プロトコルは二重引用符で囲まれ、プロトコルはカンマで区切られます。TLS リスナーで ALPN ポリシーが設定されていない場合、有効なクライアント hello メッセージが送信されない場合、または内線番号が存在しない場合、この値は - に設定されます。文字列は、256 バイトを超える場合は切り捨てられます。
tls_connection_creation_time	TLS 接続の最初に記録された時間 (ISO 8601 形式)。

ログエントリの例

以下にログエントリの例を示します。読みやすくするためだけの目的で、テキストは複数の行に表示されています。

次に、ALPN ポリシーを使用しない TLS リスナーの例を示します。

```
tls 2.0 2018-12-20T02:59:40 net/my-network-loadbalancer/c6e77e28c25b2234
 g3d4b5e8bb8464cd
72.21.218.154:51341 172.100.100.185:443 5 2 98 246 -
arn:aws:acm:us-east-2:671290407336:certificate/2a108f19-aded-46b0-8493-c63eb1ef4a99 -
ECDHE-RSA-AES128-SHA tlsv12 -
my-network-loadbalancer-c6e77e28c25b2234.elb.us-east-2.amazonaws.com
- - - 2018-12-20T02:59:30
```

次に、ALPN ポリシーを使用する TLS リスナーの例を示します。

```
tls 2.0 2020-04-01T08:51:42 net/my-network-loadbalancer/c6e77e28c25b2234
 g3d4b5e8bb8464cd
72.21.218.154:51341 172.100.100.185:443 5 2 98 246 -
arn:aws:acm:us-east-2:671290407336:certificate/2a108f19-aded-46b0-8493-c63eb1ef4a99 -
ECDHE-RSA-AES128-SHA tlsv12 -
my-network-loadbalancer-c6e77e28c25b2234.elb.us-east-2.amazonaws.com
h2 h2 "h2","http/1.1" 2020-04-01T08:51:20
```

アクセスログファイルの処理

アクセス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ファイルを開くと、ファイルは解凍され、情報が表示されます。ファイルをダウンロードする場合、情報を表示するには解凍する必要があります。

ウェブサイトの需要が大きい場合は、ロードバランサーによって数 GB のデータ量のログファイルが生成されることがあります。このような大容量のデータは、行単位で処理できない場合があります。このため、場合によっては、並列処理ソリューションを提供する分析ツールを使用する必要があります。例えば、次の分析ツールを使用するとアクセスログの分析と処理を行うことができます。

- 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Network Load Balancer ログのクエリを参照してください。
- Loggly

- Splunk
- Sumo Logic

Network Load Balancer のアクセスログを有効にする

ロードバランサーのアクセスログの作成を有効にする場合は、ロードバランサーがログを保存する S3; バケットの名前を指定する必要があります。このバケットは、バケットにアクセスログを書き込む許可を Elastic Load Balancing に付与するバケットポリシーが必要です。

> **Important**
>
> アクセスログが作成されるのは、ロードバランサーに TLS リスナーがあり、TLS リクエストに関する情報のみが含まれる場合のみです。

バケットの要件

既存のバケットを使用するか、アクセスログ専用のバケットを作成できます。バケットは、次の要件を満たしている必要があります。

要件

- バケットは、ロードバランサーと同じリージョンに配置されている必要があります。バケットとロードバランサーは、異なるアカウントにより所有できます。
- 指定するプレフィックスに `AWSLogs` を含めることはできません。指定したバケット名とプレフィックスの後に、`AWSLogs` で始まるファイル名部分が追加されます。
- このバケットは、バケットにアクセスログを書き込む許可を付与するバケットポリシーが必要です。バケットポリシーは、バケットのアクセス許可を定義するためにアクセスポリシー言語で記述された JSON ステートメントのコレクションです。

バケットポリシーの例

以下は、ポリシーの例です。`Resource` 要素については、*`amzn-s3-demo-destination-bucket`* をアクセスログの S3 バケットの名前に置き換えます。バケットプレフィックスを使用していない場合は、*`Prefix/`* を必ず省略してください。には`aws:SourceAccount`、ロードバランサーを持つ AWS アカウントの ID を指定します。`aws:SourceArn` については、*`region`* と *`012345678912`* をそれぞれロードバランサーのリージョンとアカウント ID に置き換えます。

JSON

```
{
    "Version":"2012-10-17",
    "Id": "AWSLogDeliveryWrite",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-bucket",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": [
                        "012345678912"
                    ]
                },
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:logs:us-east-1:012345678912:*"
                    ]
                }
            }
        },
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::amzn-s3-demo-destination-
bucket/Prefix/AWSLogs/account-ID/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:SourceAccount": [
                        "012345678912"
                    ]
                },
```

```
                    "ArnLike": {
                        "aws:SourceArn": [
                            "arn:aws:logs:us-east-1:012345678912:*"
                        ]
                    }
                }
            }
        ]
}
```

暗号化

Amazon S3 アクセスログバケットのサーバー側の暗号化は、次のいずれかの方法で有効にできます。

- Amazon S3 が管理するキー (SSE-S3)
- AWS KMS AWS Key Management Service (SSE-KMS) † に保存されているキー

† Network Load Balancer アクセスログでは、 AWS マネージドキーを使用することはできません。カスタマーマネージドキーを使用する必要があります。

詳細については、Amazon S3暗号化 (SSE-S3) の指定」およびAWS KMS 「 (SSE-KMS) を使用したサーバー側の暗号化の指定」を参照してください。 Amazon S3

キーポリシーで、ログの暗号化および復号化する許可をサービスに与える必要があります。以下は、ポリシーの例です。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "delivery.logs.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
```

```
          "kms:Decrypt",
          "kms:ReEncrypt*",
          "kms:GenerateDataKey*",
          "kms:DescribeKey"
        ],
        "Resource": "*"
      }
    ]
}
```

アクセスログを設定

以下の手順を使用して、リクエスト情報を収集して S3 バケットにログファイルを配信するように、アクセスログを設定します。

Console

アクセスログを有効にするには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[ロードバランサー] を選択します。
3. ロードバランサーの名前を選択して、その詳細ページを開きます。
4. [属性] タブで、[編集] を選択します。
5. [モニタリング] で [アクセスログ] をオンにします。
6. [S3 ロケーション] には、ログファイルの S3 URI を入力します。指定する URI は、プレフィックスを使用しているかどうかによって異なります。
   - プレフィックスがある URI: s3://*amzn-s3-demo-logging-bucket*/*logging-prefix*
   - プレフィックスがない URI:: s3://*amzn-s3-demo-logging-bucket*
7. [Save changes] (変更の保存) をクリックします。

AWS CLI

アクセスログを有効にするには

関連する属性で modify-load-balancer-attributes コマンドを使用します。

```
aws elbv2 modify-load-balancer-attributes \
```

```
    --load-balancer-arn load-balancer-arn \
    --attributes \
        Key=access_logs.s3.enabled,Value=true \
        Key=access_logs.s3.bucket,Value=amzn-s3-demo-logging-bucket \
        Key=access_logs.s3.prefix,Value=logging-prefix
```

CloudFormation

アクセスログを有効にするには

AWS::ElasticLoadBalancingV2::LoadBalancer リソースを更新して、関連する属性を含めます。

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-nlb
      Type: network
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "access_logs.s3.enabled"
          Value: "true"
        - Key: "access_logs.s3.bucket"
          Value: "amzn-s3-demo-logging-bucket"
        - Key: "access_logs.s3.prefix"
          Value: "logging-prefix"
```

Network Load Balancer のアクセスログを無効にする

ロードバランサーのアクセスログの作成は、いつでも無効にできます。アクセスログの作成を無効にした後は、削除するまでアクセスログは S3; バケットに残されたままです。詳細については、「Amazon S3 ユーザーガイド」の「S3 バケットの作成、設定、操作」を参照してください。

Console

アクセスログを無効化するには

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、[ロードバランサー] を選択します。
3. ロードバランサーの名前を選択して、その詳細ページを開きます。
4. [属性] タブで、[編集] を選択します。
5. [モニタリング] で [アクセスログ] をオフにします。
6. [Save changes] (変更の保存) をクリックします。

AWS CLI

アクセスログを無効化するには

modify-load-balancer-attributes コマンドを使用します。

```
aws elbv2 modify-load-balancer-attributes \
    --load-balancer-arn load-balancer-arn \
    --attributes Key=access_logs.s3.enabled,Value=false
```

Network Load Balancer をトラブルシューティングする

以下の情報は、Network Load Balancer の問題のトラブルシューティングに役立ちます。

登録されたターゲットが実行中でない

ターゲットが `InService` 状態になるまでに予想以上に時間がかかっている場合、ヘルスチェックに合格していない可能性があります。ターゲットは、ヘルスチェックに合格するまで実行されません。詳細については、「Network Load Balancer ターゲットグループのヘルスチェック」を参照してください。

インスタンスがヘルスチェックに合格していないことを確認したら、以下についてチェックします。

セキュリティグループでトラフィックが許可されていない

インスタンスに関連付けられたセキュリティグループでは、ヘルスチェックポートとヘルスチェックプロトコルを使用して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。詳細については、「ターゲットセキュリティグループ」を参照してください。また、ロードバランサーのセキュリティグループは、インスタンスへのトラフィックを許可する必要があります。詳細については、「Network Load Balancer のセキュリティグループを更新する」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) ではトラフィックが許可されない

インスタンスのサブネットとロードバランサーのサブネットに関連付けられているネットワーク ACL は、ロードバランサーからのトラフィックとヘルスチェックを許可する必要があります。詳細については、「ネットワーク ACL」を参照してください。

リクエストがターゲットにルーティングされない

以下を確認します。

セキュリティグループでトラフィックが許可されていない

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では、リスナーポートからクライアント IP アドレス (ターゲットがインスタンス ID で指定されている場合) またはロードバランサーノード (ターゲットが IP アドレスで指定されている場合) へのトラフィックが許可されている必

要があります。詳細については、「ターゲットセキュリティグループ」を参照してください。また、ロードバランサーのセキュリティグループは、インスタンスへのトラフィックを許可する必要があります。詳細については、「Network Load Balancer のセキュリティグループを更新する」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) ではトラフィックが許可されない

VPC のサブネットに関連付けられているネットワーク ACL では、リスナーポートでロードバランサーとターゲットの双方向の通信が許可されている必要があります。詳細については、「ネットワーク ACL」を参照してください。

有効になっていないアベイラビリティーゾーンにターゲットがある

ターゲットをアベイラビリティーゾーンに登録したが、アベイラビリティーゾーンを有効にしていない場合、登録したターゲットはロードバランサーからのトラフィックを受信しません。

インスタンスがピア接続 VPC にある

ロードバランサー VPC とピア接続されている VPC にインスタンスがある場合、インスタンス ID ではなく IP アドレスで、そのインスタンスをロードバランサーに登録する必要があります。

設定されたサーバー ID がターゲットに設定された ID と一致しません

QUIC リスナーを使用している場合は、ターゲットに設定された ID が Network Load Balancer ターゲットグループで設定された ID と一致することを確認してください。

ターゲットが受け取るヘルスチェックリクエストが想定よりも多い

Network Load Balancer のヘルスチェックは分散され、コンセンサスメカニズムを使用してターゲットのヘルスを判断します。そのため、ターゲットは `HealthCheckIntervalSeconds` 設定で設定されているヘルスチェック数よりも多くのヘルスチェックを受けます。

ターゲットが受け取るヘルスチェックリクエストが想定よりも少ない

`net.ipv4.tcp_tw_recycle` が有効化されているかどうかを確認します。この設定は、ロードバランサーに関する問題が発生することが判っています。`net.ipv4.tcp_tw_reuse` 設定の方が安全であると見なされています。

異常なターゲットがロードバランサーからリクエストを受信する

この状態は、登録されているすべてのターゲットに異常がある場合に発生します。少なくとも 1 つの正常なターゲットが登録されている場合、Network Load Balancer は、この正常な登録済みターゲットに対してのみリクエストをルーティングします。

登録されているのが異常なターゲットのみの場合、Network Load Balancer は、登録されたすべてのターゲットに対しリクエストをルーティングします。これは、fail-open モードと呼ばれます。すべてのターゲットに異常があり、各アベイラビリティーゾーン内にリクエストの送信先となる正常なターゲットが見つからない場合、Network Load Balancer は、DNS からすべての IP アドレスを削除する代わりに、この fail-open モードを使用します。

ホストヘッダーの不一致により、ターゲットが HTTP または HTTPS ヘルスチェックに失敗する

ヘルスチェックリクエストの HTTP ホストヘッダーには、ターゲットの IP アドレスおよびヘルスチェックポートではなく、ロードバランサーノードの IP アドレスおよびリスナーポートが含まれます。受信リクエストをホストヘッダーでマッピングする場合は、ヘルスチェックが任意の HTTP ホストヘッダーと一致することを確認する必要があります。別のオプションとして、別のポートに別々の HTTP サービスを追加し、代わりにそのポートをヘルスチェックに使用するようにターゲットグループを設定することもできます。または、TCP ヘルスチェックの使用を検討してください。

セキュリティグループをロードバランサーに関連付けできない

Network Load Balancer がセキュリティグループなしで作成された場合、作成後にセキュリティグループをサポートすることはできません。セキュリティグループは、作成中にロードバランサに関連付けるか、または最初にセキュリティグループを使用して作成した既存のロードバランサーに関連付けることができます。

すべてのセキュリティグループを削除できない

セキュリティグループを使用して Network Load Balancer が作成された場合は、常に 1 つ以上のセキュリティグループが関連付けられている必要があります。ロードバランサーからすべてのセキュリティグループを同時に削除することはできません。

TCP_ELB_Reset_count メトリクスを増加

クライアントが Network Load Balancer を通じて行う TCP リクエストごとに、その接続の状態が追跡されます。アイドルタイムアウトよりも長い時間、クライアントからもターゲットからもその接続経由でデータが送信されない場合、接続は閉じられます。アイドルタイムアウト期間の経過後にクライアントまたはターゲットがデータを送信した場合、TCP RST パケットを受信して、接続が無効になったことを示します。さらに、ターゲットが異常になると、ロードバランサーは、ターゲットに関連付けられたクライアント接続で受信したパケットの TCP RST を送信します (異常なターゲットがトリガーしたロードバランサーが起動しなかった場合以外)。

`UnhealthyHostCount` メトリクスが増加する直前または増加すると同時に、`TCP_ELB_Reset_Count` メトリクスにスパイクが見られる場合は、ターゲットが失敗し始めたが異常とマークされていないため、TCP RST パケットが送信された可能性があります。`TCP_ELB_Reset_Count` で永続的な増加が見られたら、ターゲットが正常でないとマークされない場合、期限切れのフローでデータを送信しているクライアントの VPC フローログを確認できます。

ターゲットからそのロードバランサーへのリクエストが接続タイムアウトになる

ターゲットグループでクライアント IP 保存が有効になっているかどうかを確認します。 NAT ループバック（ヘアピニングとも呼ばれる）は、クライアント IP 保存が有効になっている場合はサポートされません。

インスタンスが、登録されているロードバランサーのクライアントである場合で、クライアント IP 保護が有効になっている場合、リクエストが別のインスタンスにルーティングされる場合のみ接続が成功します。送信元と同じインスタンスにリクエストがルーティングされている場合、送信元と宛先の IP アドレスが同じであるため、接続がタイムアウトします。IP アドレスが異なる場合でも、同じ EC2 ワーカーノードインスタンスで実行されている Amazon EKS ポッドに適用されることに注意してください。

インスタンスが、それが登録されているロードバランサーにリクエストを送信する必要がある場合は、次のいずれかを実行します。

- クライアント IP の無効化 代わりに、プロキシプロトコル v2 を使用してクライアント IP アドレスを取得します。
- 通信する必要があるコンテナが異なるコンテナインスタンスにあることを確認します。

Network Load Balancer にターゲットを移動する際にパフォーマンスが低下する

Classic Load Balancer と Application Load Balancer はどちらも接続の多重化を使用しますが、Network Load Balancer では使用しません。したがって、ターゲットは Network Load Balancer の背後で複数の TCP 接続を受け取ることができます。必ず、ターゲットが受信する可能性のある接続リクエストのボリュームを処理できるようにしてください。

バックエンドフローのポート割り当てエラー

クライアント IP の保存が無効な場合、Network Load Balancer は一意の各ターゲット (IP アドレスとポート) に対して 55,000 の同時接続または 1 分あたり約 55,000 の接続をサポートします。これらの制限を超えた場合、ポート割り当てエラーが発生する可能性が高くなります。ポート割り当てエラーは、`PortAllocationErrorCount` メトリクスを使用して追跡できます。`ActiveFlowCount` メトリクスを使用してアクティブな接続を追跡できます。詳細については、「Network Load Balancer の CloudWatch メトリクス」を参照してください。

ポート割り当てエラーを修正するには、ターゲットグループにさらに多くのターゲットを追加することをお勧めします。

または、ターゲットグループにターゲットを追加できない場合は、ロードバランサーネットワークインターフェイスに最大 7 つのセカンダリ IP アドレスを追加できます。セカンダリ IP アドレスは、対応するサブネットの IPv4 CIDR ブロックから自動的に割り当てられます。各セカンダリ IP アドレスは、6 つのネットワークアドレスユニットを消費します。セカンダリ IP アドレスを追加した後は、削除できないことに注意してください。セカンダリ IP アドレスを解放する唯一の方法は、ロードバランサーを削除することです。

断続的な TCP 接続確立の失敗または TCP 接続確立の遅延

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、クライアントは同じ送信元一時ポートを使用して異なる送信先 IP アドレスに接続できます。これらの送信先 IP アドレスは、クロスゾーン負荷分散が有効になっている場合は同じロードバランサーから (異なるアベイラビリティーゾーンで）、または同じターゲット IP アドレスと登録されたポートを使用する異なる Network Load Balancer から送信できます。この場合、これらの接続が同じターゲット IP アドレスとポートにルーティングされると、ターゲットは同じクライアント IP アドレスとポートから送信されるため、重複した接続が表示されます。これにより、これらの接続の 1 つを確立するときに、接続エラーや遅延が発生します。これは、クライアントの前にある NAT デバイスがあり、複数の Network Load

Balancer IP アドレスに同時に接続するときに同じ送信元 IP アドレスと送信元ポートが割り当てられている場合に頻繁に発生します。

このタイプの接続エラーは、クライアントまたは NAT デバイスによって割り当てられた送信元の一時ポートの数を増やすか、ロードバランサーのターゲット数を増やして減らすことができます。クライアントは、これらの接続の失敗後に再接続するときに使用するソースポートを変更することをお勧めします。このタイプの接続エラーを防ぐために、単一の Network Load Balancer を使用している場合は、クロスゾーン負荷分散を無効にすることを検討できます。複数の Network Load Balancer を使用している場合は、複数のターゲットグループに登録されている同じターゲット IP アドレスとポートを使用しないことを検討できます。または、クライアント IP 保存の無効化を検討することもできます。クライアント IP が必要な場合は、Proxy Protocol v2 を使用して取得できます。Proxy Protocol v2 の詳細については、「Proxy Protocol」を参照してください。

ロードバランサーのプロビジョニング時に発生する可能性のあるエラー

Network Load Balancer がプロビジョニング中に失敗する理由の 1 つとして、既に割り当てられているか、別の場所で割り当てられている IP アドレス (EC2 インスタンスのセカンダリ IP アドレスとして割り当てられているなど) を使用していることが考えられます。この IP アドレスにより、ロードバランサーの設定が妨げられ、状態は `failed` になります。この問題は、関連付けられた IP アドレスの割り当てを解除し、作成プロセスを再試行することで解決できます。

トラフィックがターゲット間で不均等に分散されている

TCP および TLS リスナーは TCP 接続をルーティングし、UDP リスナーは UDP ストリームをルーティングします。ロードバランサーは、フローハッシュアルゴリズムを使用してターゲットを選択します。クライアントからの 1 つの接続は本質的にスティッキーです。

一部のターゲットが他のターゲットよりも多くのトラフィックを受信するように見える場合は、VPC フローログを確認することをお勧めします。各ターゲット IP アドレスの一意の接続の数を比較します。ターゲットの登録、登録解除、異常なターゲットはこれらの接続番号に影響を与えるため、タイムウィンドウはできるだけ短くしてください。

次に、接続を不均等に分散できるシナリオを示します。

- 少数のターゲットから始めて後で追加のターゲットを登録する場合、元のターゲットは引き続きクライアントと接続します。HTTP ワークロードでは、キープアライブによりクライアントが接続を

再利用できるようになります。ウェブアプリケーションの最大キープアライブを減らすと、クライアントは新しい接続をより頻繁に開くようになります。

- ターゲットグループの維持が有効になっている場合、クライアントの数が少なく、クライアントは単一の送信元 IP アドレスを持つ NAT デバイスを介して通信し、これらのクライアントからの接続は同じターゲットにルーティングされます。
- クロスゾーンロードバランシングが無効で、クライアントがロードバランサーゾーンの 1 つからのロードバランサーの IP アドレスを優先する場合、接続はロードバランサーゾーン間で不均等に分散されます。

DNS の名前解決の対象 IP アドレスの数が有効なアベイラビリティーゾーンの数より少ないです。

アベイラビリティーゾーンに少なくとも 1 つの正常なホストがある場合、Network Load Balancer は有効なアベイラビリティーゾーンごとに IP アドレスを 1 つ提供するのが理想です。特定のアベイラビリティーゾーンに正常なホストがなく、クロスゾーンのロードバランシングが無効になっている場合は、その AZ に対応する Network Load Balancer の IP アドレスが DNS から削除されます。

例えば、Network Load Balancer が有効なアベイラビリティーゾーンを 3 つ持っており、すべてのアベイラビリティーゾーンには、正常なターゲットインスタンスが少なくとも 1 つ登録されているとします。

- アベイラビリティーゾーン A に登録されているターゲットインスタンス (のいずれか) が異常になると、Network Load Balancer でアベイラビリティーゾーン A に対応する IP アドレスが DNS から削除されます。
- 有効なアベイラビリティーゾーンのうち 2 つで、登録されたターゲットインスタンス (のいずれか) に異常がある場合は、対応する 2 つの Network Load Balancer の IP アドレスが DNS から削除されます。
- 有効なすべてのアベイラビリティーゾーンで、登録されたすべてのターゲットインスタンスが正常ではない場合、フェールオープンモードが有効化され、その結果、DNS は有効な 3 つの AZ からのすべての IP アドレスを提供するようになります。

IP フラグメント化されたパケットがターゲットにルーティングされない

Network Load Balancer は、UDP 以外のトラフィックの IP フラグメント化されたパケットをサポートしていません。

リソースマップを使用して異常なターゲットをトラブルシューティングする

Network Load Balancer ターゲットがヘルスチェックに合格しなかった場合は、リソースマップを使用して異常なターゲットを見つけ、エラー理由コードに基づいてアクションを実行できます。詳細については、「Network Load Balancer リソースマップを表示する」を参照してください。

リソースマップには、[概要] と [異常なターゲットマップ] という 2 つのビューがあります。[概要] はデフォルトで選択されており、ロードバランサーのすべてのリソースが表示されます。[異常なターゲットマップ] ビューを選択すると、Network Load Balancer に関連付けられている各ターゲットグループ内の異常なターゲットのみが表示されます。

> **Note**
>
> リソースマップ内の該当するすべてのリソースのヘルスチェックの概要とエラーメッセージを表示するには、[リソースの詳細を表示] を有効にする必要があります。有効になっていない場合は、各リソースを選択して詳細を表示する必要があります。

[ターゲットグループ] 列には、各ターゲットグループの正常なターゲットと異常なターゲットの概要が表示されます。これは、すべてのターゲットがヘルスチェックに合格しなかったのか、特定のターゲットのみが合格しなかったのかを判断するのに役立ちます。ターゲットグループ内のすべてのターゲットがヘルスチェックに合格しなかった場合は、ターゲットグループのヘルスチェック設定を確認します。ターゲットグループの名前を選択して、新しいタブで詳細ページを開きます。

[ターゲット] 列には、各ターゲットの TargetID と現在のヘルスチェックステータスが表示されます。ターゲットに異常がある場合、ヘルスチェックのエラー理由コードが表示されます。1 つのターゲットがヘルスチェックに合格しなかった場合は、ターゲットに十分なリソースがあることを確認します。ターゲットの ID を選択して、新しいタブで詳細ページを開きます。

[エクスポート] を選択すると、Network Load Balancer のリソースマップの現在のビューを PDF としてエクスポートできます。

インスタンスがヘルスチェックに合格していないことを確認したら、エラー理由コードに基づいて、以下の点を確認します。

- [異常: リクエストがタイムアウトしました]
  - ターゲットと Network Load Balancer に関連付けられたセキュリティグループとネットワークアクセスコントロールリスト (ACL) が接続をブロックしていないことを確認します。
  - Network Load Balancer からの接続を受け入れるのに十分な容量がターゲットにあることを確認します。
  - Network Load Balancer のヘルスチェックレスポンスは、各ターゲットのアプリケーションログで確認できます。詳細については、「ヘルスチェックの理由コード」を参照してください。
- [異常: FailedHealthChecks]
  - ターゲットがヘルスチェックポートのトラフィックをリッスンしていることを確認します。

> TLS リスナーを使用している場合
>
> フロントエンド接続に使用するセキュリティポリシーを選択します。バックエンド接続に使用するセキュリティポリシーは、使用中のフロントエンドのセキュリティポリシーに基づいて自動的に選択されます。リスナーに次のものがある場合:
>
> - FIPS ポスト量子 TLS ポリシー - バックエンド接続の使用 `ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09`
> - FIPS ポリシー - バックエンド接続の使用 `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04`
> - ポスト量子 TLS ポリシー - バックエンド接続の使用 `ELBSecurityPolicy-TLS13-1-0-PQ-2025-09`
> - TLS 1.3 ポリシー - バックエンド接続の使用 `ELBSecurityPolicy-TLS13-1-0-2021-06`
> - 他のすべての TLS ポリシーのバックエンド接続では、`ELBSecurityPolicy-2016-08`
>
> 詳細については、「Security policies」を参照してください。

  - ターゲットがサーバー証明書とキーをセキュリティポリシーで指定された正しい形式で提供していることを確認します。
  - ターゲットが 1 つ以上の一致する暗号と、TLS ハンドシェイクを確立するために Network Load Balancer が提供しているプロトコルをサポートしていることを確認します。

Network Load Balancer のクォータ

AWS アカウント には、サービスごとに AWS 、以前は制限と呼ばれていたデフォルトのクォータがあります。特に明記されていない限り、クォータは地域固有です。一部のクォータについては引き上げをリクエストできますが、その他のクォータについてはリクエストできません。

Network Load Balancer のクォータを表示するには、Service Quotas コンソールを開きます。ナビゲーションペインで、[AWS のサービス]、[Elastic Load Balancing] の順に選択します。また、Elastic Load Balancing 用に describe-account-limits (AWS CLI) コマンドを使用することもできます。

クォータの増加をリクエストするには、Service Quotas ユーザーガイドの Requesting a quota increase を参照してください。Service Quotas でクォータがまだ利用できない場合は、サービスクォータ引き上げのリクエストを送信してください。

クォータ

- ロードバランサー
- ターゲットグループ
- ロードバランサーキャパシティユニット

ロードバランサー

AWS アカウント には、Network Load Balancer に関連する次のクォータがあります。

名前	デフォルト	引き上げ可能
Network Load Balancer あたりの証明書	25	あり
Network Load Balancer あたりのリスナー	50	不可
VPC あたりの Network Load Balancer ENI	1,200 [1]	あり
リージョンあたりの Network Load Balancer	50	はい
Network Load Balancer ごとのアベイラビリティーゾーンあたりのターゲット	500 [2, 3]	あり
Network Load Balancer あたりのターゲット	3,000 [3]	あり

[1] それぞれの Network Load Balancer は、ゾーンごとに 1 つのネットワークインターフェイスを使用します。クォータは VPC レベルで設定されます。サブネットまたは VPC を共有する場合、使用量はテナント全体で計算されます。

[2] ターゲットが N ターゲットグループで登録されている場合、この制限に対して N ターゲットとしてカウントされます。Network Load Balancer のターゲットである各 Application Load Balancer は、50 ターゲット (クロスゾーン負荷分散が無効になっている場合)、または 100 ターゲット (クロスゾーン負荷分散が有効になっている場合) としてカウントされます。

[3] クロスゾーンロードバランシングが有効になっている場合、アベイラビリティーゾーンの数に関係なく、ロードバランサーあたりの最大ターゲット数は 500 です。

ターゲットグループ

次のクォータはターゲットグループ用です。

名前	デフォルト	引き上げ可能
リージョンあたりのターゲットグループ	3,000 [1]	あり
リージョンごとのターゲットグループあたりのターゲット (インスタンスまたは IP アドレス)	1,000	あり
リージョンごとのターゲットグループあたりのターゲット (Application Load Balancer)	1	不可

[1] このクォータは、Application Load Balancer および Network Load Balancer によって共有されます。

ロードバランサーキャパシティユニット

次のクォータは、ロードバランサーキャパシティユニット (LCU) 向けです。

名前	デフォルト	引き上げ可能
アベイラビリティーゾーンごとの Network Load Balancer あたりのリザーブド Network Load Balancer キャパシティユニット (LCU)	45000	はい

名前	デフォルト	引き上げ可能
リージョンあたりの予約済み Network Load Balancer キャパシティユニット (LCU)	0	あり

Network Load Balancer のドキュメント履歴

次の表に、Network Load Balancer のリリース情報を示します。

変更	説明	日付
加重ターゲットグループ	このリリースでは、加重ターゲットグループを使用したデフォルトアクションのサポートが追加されました。	2025 年 11 月 19 日
QUIC および TCP_QUIC プロトコルのサポート	このリリースでは、QUIC および TCP_QUIC プロトコルのサポートが追加されました。	2025 年 11 月 13 日
セカンダリ IPv4 アドレス	このリリースでは、ロードバランサーネットワークインターフェイスにセカンダリ IPv4 アドレスを追加するサポートが追加されました。	2025 年 7 月 29 日
アベイラビリティーゾーンの無効化	このリリースでは、既存のロードバランサーのアベイラビリティーゾーンを無効化するサポートが追加されました。	2025 年 2 月 13 日
キャパシティユニットの予約	このリリースでは、ロードバランサーの最小キャパシティを設定するサポートが追加されました。	2024 年 11 月 20 日
デュアルスタックロードバランサーの IPv6 経由の UDP サポート	このリリースでは、クライアントは IPv6 を使用して UDP ベースのアプリケーションにアクセスできるようになりました。	2024 年 10 月 31 日

RSA 3072 ビットおよび ECDSA 256/384/521 ビット証明書	このリリースでは、RSA 3072 ビット証明書、および AWS Certificate Manager (ACM) 経由の楕円曲線デジタル署名アルゴリズム (ECDSA) 256、384、521 ビット証明書のサポートが追加されました。	2024 年 1 月 19 日
FIPS 140-3 TLS の終了	このリリースでは、TLS 接続を終了するときに FIPS 140-3 暗号モジュールを使用するセキュリティポリシーが追加されました。	2023 年 11 月 20 日
ゾーン DNS アフィニティ	このリリースでは、ロードバランサーの DNS を解決して、同じアベイラビリティーゾーン (AZ) で IP アドレスを受信するクライアントのサポートが追加されました。	2023 年 10 月 12 日
異常なターゲット接続の終了を無効にする	このリリースでは、ヘルスチェックに合格しなかったターゲットへのアクティブな接続を維持するサポートが追加されました。	2023 年 10 月 12 日
デフォルトの UDP 接続の終了	このリリースでは、登録解除タイムアウトの終了時にデフォルトで UDP 接続を終了するサポートが追加されました。	2023 年 10 月 12 日

IPv6 を使用してターゲットを登録する	このリリースでは、IPv6 でアドレス指定されたときに、インスタンスをターゲットとして登録するサポートが追加されました。	2023 年 10 月 2 日
Network Load Balancer のセキュリティグループ	このリリースでは、作成時にセキュリティグループを Network Load Balancer に関連付けるためのサポートが追加されています。	2023 年 8 月 10 日
ターゲットグループの正常性	このリリースでは、正常でなければならないターゲットの最小数または割合、およびしきい値に達しない場合にロードバランサーが実行するアクションを設定するサポートが追加されています。	2022 年 11 月 17 日
ヘルスチェックの設定	このリリースでは、ヘルスチェックの設定が改善されています。	2022 年 11 月 17 日
クロスゾーンロードバランサー	このリリースでは、クロスゾーン負荷分散をターゲットグループのレベルで設定するためのサポートが追加されました。	2022 年 11 月 17 日
IPv6 ターゲットグループ	このリリースでは、Network Load Balancer の IPv6 ターゲットグループの設定に対するサポートが追加されました。	2021 年 11 月 23 日

IPv6 内部ロードバランサー	このリリースでは、Network Load Balancer の IPv6 ターゲットグループの設定に対するサポートが追加されました。	2021 年 11 月 23 日
TLS 1.3	このリリースでは TLS バージョン 1.3 をサポートするセキュリティポリシーが追加されました。	2021 年 10 月 14 日
ターゲットとしての Application Load Balancer	このリリースでは、Network Load Balancer のターゲットとして Application Load Balancer を設定するサポートが追加されています。	2021 年 9 月 27 日
クライアント IP の保存	このリリースでは、クライアント IP の保存を設定できるようになりました。	2021 年 2 月 4 日
TLS バージョン 1.2 をサポートする FS のセキュリティポリシー	このリリースでは、TLS バージョン 1.2 をサポートする前方秘匿性 (FS) のセキュリティポリシーが追加されました。	2020 年 11 月 24 日
デュアルスタックモード	このリリースでは、デュアルスタックモードのサポートが追加され、クライアントが IPv4 アドレスと IPv6 アドレスの両方を使用してロードバランサーに接続できるようになります。	2020 年 11 月 13 日

登録解除時の接続終了	このリリースでは、登録解除タイムアウトの終了後に登録解除されたターゲットへの接続を閉じるサポートが追加されました。	2020 年 11 月 13 日
ALPN ポリシー	このリリースでは、Application-Layer Protocol Negotiation (ALPN) プリファレンスリストのサポートが追加されました。	2020 年 5 月 27 日
スティッキーセッション	このリリースでは、送信元 IP アドレスとプロトコルに基づくスティッキーセッションのサポートが追加されています。	2020 年 2 月 28 日
共有サブネット	このリリースでは、別の AWS アカウントと共有するサブネットを指定するためのサポートが追加されています。	2019 年 11 月 26 日
プライベート IP アドレス	このリリースでは、内部ロードバランサーのアベイラビリティーゾーンを有効にするときに指定するサブネットの IPv4 アドレス範囲からプライベート IP アドレスを提供できます。	2019 年 11 月 25 日
サブネットの追加	このリリースでは、ロードバランサーを作成した後で、追加のアベイラビリティーゾーンを有効にするサポートが追加されています。	2019 年 11 月 25 日

[FS のセキュリティポリシー](#)	このリリースでは、新しい 3 つの、事前定義済みの Forward Secrecy セキュリティポリシーへのサポートが追加されました。	2019 年 10 月 8 日
[SNI サポート](#)	このリリースでは、Server Name Indication (SNI) へのサポートを追加しています。	2019 年 9 月 12 日
[UDP プロトコル](#)	このリリースでは、UDP プロトコルのサポートが追加されました。	2019 年 6 月 24 日
[新しいリージョンで利用可能に](#)	このリリースでは、アジアパシフィック (大阪) リージョンの Network Load Balancer のサポートが追加されました。	2019 年 6 月 12 日
[TLS プロトコル](#)	このリリースでは、TLS プロトコルのサポートが追加されました。	2019 年 1 月 24 日
[クロスゾーン負荷分散](#)	このリリースでは、クロスゾーン負荷分散を有効にするためのサポートを追加しています。	2018 年 2 月 22 日
[Proxy Protocol](#)	このリリースでは、Proxy Protocol を有効にするためのサポートが追加されます。	2017 年 11 月 17 日
[IP アドレスをターゲットに設定](#)	このリリースでは、IP アドレスをターゲットとして登録する機能のサポートが追加されます。	2017 年 9 月 21 日

[新しい種類のロードバランサー]()	このリリースの Elastic Load Balancing では、Network Load Balancer が導入されています。	2017 年 9 月 7 日