aws

Network Load Balancers

Elastic Load Balancing

Elastic Load Balancing: Network Load Balancers

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リングできます。

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、および TLS 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リスナールールに指定されたターゲットグループに登録された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。毎秒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、接続リクエストを受信すると、デフォルトルールのターゲットグループからターゲットを選択します。リスナー構成で指定されたポート上の選択したターゲットへの TCP 接続を開こうとします。

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、「アベイラビリティーゾーン」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 (TTL) を守 time-to-liveらず、DNS から削除された後に IP アドレスにリクエストを送信すると、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。

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。

- ターゲットグループレベルでヘルスチェックが定義され、多くの 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るため、各サービスのヘルスを個別にモニタリングするためのサポート。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。

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

Network Load Balancer を作成するには、次のいずれかのチュートリアルを試してください。

- Network Load Balancer の開始方法
- チュートリアル: AWS CLI を使用して 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、Network Load Balancer の料金を参照してください。

Network Load Balancer の開始方法

このチュートリアルでは AWS Management Console、ウェブベースのインターフェイスである を通じて Network Load Balancer を実際に紹介します。最初の Network Load Balancer を作成するには、次のステップを完了します。

タスク

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。

開始する前に

- EC2 インスタンスに使用するアベイラビリティーゾーンを決定します。これらの各アベイラビリティーゾーンに少なくとも 1 つのパブリックサブネットがある Virtual Private Cloud (VPC) を設定します。これらのパブリックサブネットは、ロードバランサーを設定するために使用されます。その代わりに、これらのアベイラビリティーゾーンの他のサブネットで EC2 インスタンスを起動することができます。
- 各アベイラビリティーゾーンで少なくとも 1 つの EC2 インスタンスを起動します。これらのインスタンスのセキュリティグループが、リスナーポート上のクライアントからの TCP アクセスと、VPC からのヘルスチェックリクエストを許可していることを確認します。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。

ステップ 1: ターゲットグループの設定

リクエストルーティングで使用されるターゲットグループを作成します。リスナーのルールは、このターゲットグループ内の登録済みターゲットにリクエストをルーティングします。ロードバランサー

は、ターゲットグループに定義されたヘルスチェック設定を使用してこのターゲットグループ内のターゲットの状態を確認します。

コンソールを使用してターゲットグループを設定するには

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. [Create target group] を選択します。
4. ターゲットタイプを [instances] (インスタンス) として保持します。
5. [ターゲットグループ名] に、新しい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 [TCP] を選択し、[Port] (ポート) に [80] を選択します。
7. [VPC] で、インスタンスが含まれている VPC を選択します。
8. [ヘルスチェック] で、デフォルトの設定を保持します。
9. [次へ] をクリックします。
10. [ターゲットの登録] ページで、次の手順を完了します。これが、ターゲットグループを作成するオプションの手順です。ただし、ロードバランサーをテストし、ターゲットにトラフィックをルーティングしていることを確認する場合は、ターゲットを登録する必要があります。

    a. [使用可能なインスタンス] で、1 つ以上のインスタンスを選択します。
    b. デフォルトのポート 80 のままにして、[保留中として以下を含める] を選択します。
11. [ターゲットグループの作成] を選択します。

ステップ 1: ロードバランサーの種類の選択

Elastic Load Balancing では、異なる種類のロードバランサーがサポートされています。このチュートリアルでは、Network Load Balancer を作成します。

コンソールを使用して Network Load Balancer を作成するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションバーで、ロードバランサーのリージョンを選択します。EC2 インスタンス用に使用したリージョンと同じリージョンを必ず選択してください。
3. ナビゲーションペインの [Load Balancing] で、[Load Balancers] を選択します。
4. [ロードバランサーを作成] を選択します。
5. [Network Load Balancer] で、[Create] (作成) を選択します。

ステップ 2: ロードバランサーとリスナーの設定

Network Load Balancer を作成するには、まず、名前、スキーム、IP アドレスのタイプなど、ロードバランサーの基本設定情報を指定する必要があります。次に、ネットワークに関する情報と 1 つ以上のリスナーを指定します。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。これは、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を使用して設定します。サポートされるプロトコルとポートの詳細については、「リスナーの設定」を参照してください。

ロードバランサーとリスナーを設定するには

1. [ロードバランサー名] に、ロードバランサーの名前を入力します。たとえば、`my-nlb` と指定します。
2. [スキーム] および [IP アドレスタイプ] については、デフォルト値のままにします。
3. [ネットワークマッピング] で、EC2 インスタンスに使用する VPC を選択します。EC2 インスタンスの起動に使用した各アベイラビリティーゾーンについて、アベイラビリティーゾーンを選択し、そのアベイラビリティーゾーンのパブリックサブネットを 1 つ選択します。

   デフォルトでは、 はアベイラビリティーゾーンのサブネットから各ロードバランサーノードに IPv4 アドレスを AWS 割り当てます。あるいは、インターネット向けロードバランサーを作成する場合は、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。
4. セキュリティグループでは、VPC のデフォルトのセキュリティグループがあらかじめ選択されています。必要に応じて、他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する]を選択して、セキュリティニーズを満たすように作成します。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**
   >
   > この時点でロードバランサー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。
5. [リスナーとルーティング] で、デフォルトのプロトコルとポートはそのままにして、リストからターゲットグループを選択します。ポート 80 で TCP トラフィックを受け付けるリスナーが設定され、選択されたターゲットグループにトラフィックをデフォルトで転送します。
6. (オプション) タグを追加して、ロードバランサーを分類します。タグキーは、各ロードバランサーで一意である必要があります。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。

7. 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、ロードバランサーにいくつかのデフォルト属性が適用されます。ロードバランサーの作成後に、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。

ステップ 4: ロードバランサーのテスト

ロードバランサーを作成した後で、EC2 インスタンスにトラフィックを送信するかどうかを検証します。

ロードバランサーをテストするには

1. ロードバランサーが正常に作成されたことが通知されたら、[閉じる] を選択します。
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. 新しく作成した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 `healthy` であれば、ロードバランサーをテストできます。
5. ナビゲーションペインの [Load Balancing] で、[Load Balancers] を選択します。
6. 新しく作成した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。
7. ロードバランサーの DNS 名をコピーします (例: my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com)。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。

ステップ 5: ロードバランサーを削除 (オプション)

ロードバランサーが利用可能になると、ロードバランサー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。不要になったロードバランサーは削除できます。ロードバランサーが削除されると、ロードバランサーの課金も停止されます。ロードバランサーを削除しても、ロードバランサーに登録された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され続けます。

コンソールを使用してロードバランサーを削除するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [Load Balancing] で、[Load Balancers] を選択します。
3. 任意のロードバランサーのチェックボックスを選択し、[Actions] (アクション)、[Delete] (削除) を選択します。
4. 確認を求められたら、「**confirm**」を入力し、[削除] を選択します。

チュートリアル: AWS CLI を使用して Network Load Balancer を作成する

このチュートリアルでは、AWS CLI を通じて Network Load Balancer の実践的な概要を説明します。

開始する前に

- AWS CLI をインストールするか、Network Load Balancers をサポートしていないバージョンを使用している場合は AWS CLI を現行バージョンに更新します。詳細については、AWS Command Line Interface ユーザーガイドの「AWS Command Line Interface のインストール」を参照してください。
- EC2 インスタンスに使用するアベイラビリティーゾーンを決定します。これらの各アベイラビリティーゾーンに少なくとも 1 つのパブリックサブネットがある Virtual Private Cloud (VPC) を設定します。
- IPv4 またはデュアルスタックロードバランサーのどちらを作成するかを決定します。クライアントが IPv4 アドレスだけを使用してロードバランサーと通信する場合、IPv4 を使用します。クライアントが IPv4 および IPv6 アドレスを使用してロードバランサーと通信する場合、デュアルスタックを使用します。また、IPv6 を使用して IPv6 アプリケーションやデュアルスタックサブネットなどのバックエンドターゲットと通信するために、デュアルスタックを使用することもできます。
- 各アベイラビリティーゾーンで少なくとも 1 つの EC2 インスタンスを起動します。これらのインスタンスのセキュリティグループが、リスナーポート上のクライアントからの TCP アクセスと、VPC からのヘルスチェックリクエストを許可していることを確認します。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。

IPv4 ロードバランサーを作成する

最初のロードバランサーを作成するには、次のステップを完了します。

IPv4 ロードバランサーを作成するには

1. create-load-balancer コマンドを使用して IPv4 ロードバランサーを作成し、インスタンスを起動した各アベイラビリティーゾーンのパブリックサブネットを指定します。アベイラビリティーゾーンごとに 1 つだけサブネットを指定できます。

デフォルトでは、AWS CLI を使用して Network Load Balancer を作成すると、VPC のデフォルトセキュリティグループは自動的に使用されません。作成時にロードバランサーにセキュリティグループを関連付けしないと、後で追加できません。作成時に `--security-groups` オプションを使用してロード バランサーのセキュリティ グループを指定することをお勧めします。

```
aws elbv2 create-load-balancer --name my-load-balancer --type network --subnets
 subnet-0e3f5cac72EXAMPLE --security-groups sg-0123456789EXAMPLE
```

出力には、次の形式でロードバランサーの Amazon リソースネーム (ARN) が含まれます。

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/net/my-load-
balancer/1234567890123456
```

2. create-target-group コマンドを使用して、EC2 インスタンスに使用したのと同じ VPC を指定して、IPv4 ターゲットグループを作成します。IPv4 ターゲットグループでは、IP およびインスタンスタイプのターゲットがサポートされています。

```
aws elbv2 create-target-group --name my-targets --protocol TCP --port 80 --vpc-id
 vpc-0598c7d356EXAMPLE
```

出力には、次の形式のターゲットグループの ARN が含まれます。

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/1234567890123456
```

3. インスタンスをターゲットグループに登録するには、register-targets コマンドを使用します。

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

4. ターゲットグループにリクエストを転送するデフォルトルールを持つロードバランサーのリスナーを作成するには、create-listener コマンドを使用します。

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

出力には、次の形式のリスナーの ARN が含まれます。

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:listener/net/my-load-
balancer/1234567890123456/1234567890123456
```

5. （オプション）次のdescribe-target-healthコマンドを使用して、ターゲットグループの登録済みターゲットの正常性を検証できます。

```
aws elbv2 describe-target-health --target-group-arn targetgroup-arn
```

デュアルスタックロードバランサーを作成する

最初のロードバランサーを作成するには、次のステップを完了します。

デュアルスタックロードバランサーを作成するには

1. create-load-balancer コマンドを使用してデュアルスタックロードバランサーを作成し、インスタンスを起動した各アベイラビリティーゾーンのパブリックサブネットを指定します。アベイラビリティーゾーンごとに 1 つだけサブネットを指定できます。

```
aws elbv2 create-load-balancer --name my-load-balancer --type network --subnets
 subnet-0e3f5cac72EXAMPLE --ip-address-type dualstack
```

出力には、次の形式でロードバランサーの Amazon リソースネーム (ARN) が含まれます。

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:loadbalancer/net/my-load-
balancer/1234567890123456
```

2. create-target-group コマンドを使用してターゲットグループを作成し、EC2 インスタンスに使用したのと同じ VPC を指定します。

デュアルスタックロードバランサーで TCP または TLS ターゲットグループを使用する必要があります。

IPv4 ターゲットグループおよび IPv6 ターゲットグループを作成して、デュアルスタックロードバランサーに関連付けることができます。ターゲットグループの IP アドレスタイプによって、ロードバランサーがバックエンドターゲットと通信したり、バックエンドターゲットの状態をチェックしたりするのに使用する IP バージョンが決定されます。

IPv4 ターゲットグループでは、IP およびインスタンスタイプのターゲットがサポートされています。IPv6 ターゲットでは、IP ターゲットのみがサポートされています。

```
aws elbv2 create-target-group --name my-targets --protocol TCP --port 80 --vpc-id
 vpc-0598c7d356EXAMPLE  --ip-address-type [ipv4 or ipv6]
```

出力には、次の形式のターゲットグループの ARN が含まれます。

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/1234567890123456
```

3. インスタンスをターゲットグループに登録するには、register-targets コマンドを使用します。

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

4. create-listener コマンドを使用して、ターゲットグループにリクエストを転送するデフォルトルールを持つロードバランサーのリスナーを作成します。デュアルスタックロードバランサーには TCP または TLS リスナーが必要です。

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

出力には、次の形式のリスナーの ARN が含まれます。

```
arn:aws:elasticloadbalancing:us-east-2:123456789012:listener/net/my-load-
balancer/1234567890123456/1234567890123456
```

5. (オプション) 次のdescribe-target-healthコマンドを使用して、ターゲットグループの登録済みターゲットの正常性を検証できます。

```
aws elbv2 describe-target-health --target-group-arn targetgroup-arn
```

ロードバランサーの Elastic IP アドレスを指定します。

Network Load Balancer を作成するときは、サブネットマッピングを使用して、サブネットごとに 1 つの Elastic IP アドレスを指定できます。

```
aws elbv2 create-load-balancer --name my-load-balancer --type network \
--subnet-mappings SubnetId=subnet-0e3f5cac72EXAMPLE,AllocationId=eipalloc-12345678
```

ロードバランサーの削除

ロードバランサーとターゲットグループが必要なくなった場合は、次のように削除することができます。

```
aws elbv2 delete-load-balancer --load-balancer-arn loadbalancer-arn
aws elbv2 delete-target-group --target-group-arn targetgroup-arn
```

Network Load Balancer

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。クライアントはロードバランサーにリクエストを送信し、ロードバランサーは 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。

ロードバランサーを設定するには、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、ロードバランサーが最も効果的です。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 AWS 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

ロードバランサーの状態

ロードバランサーの状態は、次のいずれかです。

`provisioning`

ロードバランサーはセットアップ中です。

`active`

ロードバランサー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`

ロードバランサークラウドをセットアップできませんでした。

ロードバランサーの属性

ロードバランサーには、次の属性があります。

`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: `false`。

`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: `false`。

`ipv6.deny_all_igw_traffic`

ロードバランサー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ロードバランサー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ロードバランサーでは `false`、内部ロードバランサーでは `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 AWS 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: `false`。

`dns_record.client_routing_policy`

ロードバランサー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。

IP アドレスタイプ

クライアントがロードバランサーで使用できる 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) を使用してロードバランサーに接続する必要があります。IPv4 対応のロードバランサー (インターネット向けと内部向けの両方) では、TCP、UDP、TCP_UDP、および TLS リスナーがサポートされています。

`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 (たとえば、2001:0db8:85a3:0:0:8a2e:0370:7334) の両方を使用してロードバランサーに接続できます。デュアルスタック対応のロードバランサー (インターネット向けと内部向けの両方) では、TCP および TLS リスナーがサポートされています。

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。
- ロードバランサーのデュアルスタックモードを有効にすると、Elastic Load Balancing がロードバランサーの AAAA DNS レコードを提供します。IPv4 アドレスを使用してロードバランサー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ロードバランサー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ロードバランサー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 AWS VPN。

IP アドレスタイプの詳細については、「」を参照してくださいNetwork Load Balancer の IP アドレスの種類。

Network Load Balancer リソースマップ

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、ロードバランサーのアーキテクチャをインタラクティブに表示します。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、ロードバランサーの設定が視覚的に表示されます。

コンソールを使用して Network Load Balancer のリソースマップを表示するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[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. リソースマップタブを選択すると、ロードバランサーのリソースマップが表示されます。

リソースマップコンポーネント

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 の 2 つのビューがあります。 概要はデフォルトで選択され、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、異常なターゲットとそれら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ビューは、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た異常なターゲットのトラブルシューティング」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、リソースタイプごとに 1 つずつ、3 つのリソース列が含まれています。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独自のタイルがあり、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、そのタイル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、そのタイル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**
>
> リソースの詳細の表示をオフにすると、リソースマップ内の追加の詳細を非表示にできます。

- 各リソースタイルには、選択したときにそのリソースの詳細ページに移動するリンクが含まれています。
  - リスナー - リスナープロトコル：ポートを選択します。例えば、次のようになります: `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。例えば、次のようになります: `my-target-group`
  - ターゲット - ターゲット ID を選択します。例えば、次のようになります: `i-1234567890abcdef0`

リソースマップをエクスポートする

Export 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。

アベイラビリティーゾーン

ロードバランサーを作成するときに、ロードバランサーの 1 つまたは複数のアベイラビリティーゾーンを有効にします。ロードバランサーで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が向上します。Network Load Balancer の作成後にそのアベイラビリティーゾーンを無効にすることはできませんが、追加のアベイラビリティーゾーンを有効にすることはできます。

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 つ指定します。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、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 (「ELB net」で始まり、ロードバランサーの名前を含む記述)。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 IPv4 アドレスを取得します。こ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。

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指定することができます。独自の Elastic IP アドレスのいずれも選択しない場合、Elastic Load Balancing はサブネットごとに 1 つの Elastic IP アドレスを提供します。これらの Elastic IP アドレスは、ロードバランサーの存続期間中は変更されない静的 IP アドレスをロードバランサーに提供します。ロードバランサーを作成した後で、これらの Elastic IP アドレスを変更することはできません。

内部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 プライベート IP アドレスをサブネットごとに指定することができます。サブネットから IP アドレスを指定しない場合は、Elastic Load Balancing によって選択されます。これらのプライベート IP アドレスは、ロードバランサーの存続期間中は変更されない静的 IP アドレスをロードバランサーに提供します。ロードバランサーを作成した後で、これらのプライベート IP アドレスを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ロードバランサーの場合、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。内部ロードバランサーの場合、サブネットからプライベート IPv4 アドレス AWS を選択させ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。エラーメッセージは、「'network' タイプを使用したロードバランサーは az_name でサポートされていません」です。制約されていない別の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。
- 自分と共有されているサブネットを指定できます。
- 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。

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。

コンソールを使用してアベイラビリティーゾーンを追加するには

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. [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、そのアベイラビリティーゾーンのチェックボックスをオンにします。そのアベイラビリティーゾーンに対して 1 つのサブネットがある場合、そのサブネットが選択されます。そのアベイラビリティーゾーンに複数のサブネットがある場合は、いずれかのサブネットを選択します。アベイラビリティーゾーンにつき、1 つのサブネットしか選択できないことに注意してください。

   インターネット向けロードバランサーの場合は、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。内部ロードバランサーの場合、プライベート IP アドレスを Elastic Load Balancing で割り当てるのではなく、各サブネットの IPv4 範囲から割り当てることができます。
6. [変更の保存] をクリックします。

を使用してアベイラビリティーゾーンを追加するには AWS CLI

set-subnets コマンドを使用します。

クロスゾーン負荷分散

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。クロスゾーンロードバランサーをオンにすると、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。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サー」を参照してください。

削除保護

ロードバランサーが誤って削除されるのを防ぐため、削除保護を有効にできます。デフォルトでは、ロードバランサーで削除保護が無効になっています。

ロードバランサーの削除保護を有効にした場合、ロードバランサーを削除する前に無効にする必要があります。

コンソールを使用して削除保護を有効にするには

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. [構成] で、[削除保護] をオンにします。
6. [変更の保存] をクリックします。

コンソールを使用して削除保護を無効にするには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[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. [構成] で、[削除保護] をオンにします。
6. [変更の保存] をクリックします。

を使用して削除保護を有効または無効にするには AWS CLI

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

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。

TCP フローのアイドルタイムアウト値を 350 秒に設定します。この値は変更できません。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リセッ

トできます。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。

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。

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。

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、ロードバランサーの 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 サーバーがロードバランサーの 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、ロードバランサー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、ロードバランサー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。

ロードバランサーには、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。これらはロードバランサーノードの IP アドレスです。ロードバランサーの DNS 名はこれらのアドレスに解決されます。たとえば、ロードバランサー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、ロードバランサーノードの 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
```

ロードバランサーには、ロードバランサーノードの DNS レコードがあります。次の構文の DNS 名 (*az*.*name*-*id*.elb.*region*.amazonaws.com) を使用して、ロードバランサーノードの IP アドレスを調べることができます。

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
```

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、正常なロードバランサーの IP アドレスがすべて届きます。これにより、ロードバランサー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ロードバランサー 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。

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ロードバランサーの IP アドレスが優先されます。自身のゾーンに正常なロードバランサー 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ロードバランサーの 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 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 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のアフィニティ — 0% のゾーンアフィニティ

  クライアント DNS クエリは、すべてのロードバランサーアベイラビリティーゾーンの正常なロードバランサー IP アドレスで解決されます。

> **Note**
>
> 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。Route 53 リゾルバーの詳細については、「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。

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ロードバランサー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ロードバランサー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アフィニティを使用する場合、クロスゾーン負荷分散をオフにする必要があります。これにより、クライアントからターゲットへのロードバランサートラフィックが同じアベイラビリティーゾーン内に留まります。この設定では、クライアントトラフィックが同じ 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のクロスゾーンロードバランサー」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、すべてのロードバランサー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。
- 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

ゾーンロードバランサー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、「CloudWatch Network Load Balancer の 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オンにする

この手順のステップでは、Amazon EC2 コンソールでアベイラビリティーゾーンのアフィニティをオンにする方法について説明します。

コンソールを使用してアベイラビリティ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 (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. [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. [変更の保存] をクリックします。

を使用して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 CLI

`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。

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

この手順のステップでは、Amazon EC2 コンソールで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方法について説明します。

コンソールを使用してアベイラビリティゾーンのアフィニティをオフにするには

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. [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、[Any Availability Zone] (任意のアベイラビリティーゾーン) を選択します。
6. [変更の保存] をクリックします。

を使用してアベイラビリティーゾーンのアフィニティを無効にするには AWS CLI

`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。

Network Load Balancer を作成する

ロードバランサー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。

開始する前に、ロードバランサーの仮想プライベートクラウド (VPC) に、ターゲットがある各アベイラビリティーゾーンに少なくとも 1 つのパブリックサブネットがあることを確認してください。トラフィックをターゲットグループにルーティングするには、ターゲットグループを設定し、デフォルトとして設定するターゲットを少なくとも 1 つ登録する必要があります。

を使用してロードバランサーを作成するには、 AWS CLI「」を参照してくださいチュートリアル: AWS CLI を使用して Network Load Balancer を作成する。

を使用してロードバランサーを作成するには AWS Management Console、次のタスクを実行します。

タスク

- ステップ 1: ターゲットグループの設定
- ステップ 2: ターゲットの登録
- ステップ 3: ロードバランサーとリスナーの設定
- ステップ 4: ロードバランサーのテスト

ステップ 1: ターゲットグループの設定

ターゲットグループを設定すると、EC2 インスタンスなどのターゲットを登録できます。このステップで設定するターゲットグループは、ロードバランサーを設定するときに、リスナールールで

ターゲットグループとして使用されます。詳細については、「Network Load Balancers のターゲットグループ」を参照してください。

コンソールを使用してターゲットグループを設定するには

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、[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. [Choose a target type] (ターゲットタイプの選択) で、インスタンス ID でターゲットを登録する場合は [Instances] (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 addresses] (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。

   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。

   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。

   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 アドレスで、プロトコルが TCP または TLS の場合にのみ使用できます。

      IPv6 ターゲットグループをデュアルスタックロードバランサーに関連付ける必要があります。ターゲットグループ内のすべてのターゲットは、同じ IP アドレスタイプである必要があります。ターゲットグループが作成されると、IP アドレスタイプを変更することはできません。

   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、カウント、タイムアウト、インターバル、成功コードを選択します。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、「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。

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. [Next] を選択します。

ステップ 2: ターゲットの登録

EC2 インスタンス、IP アドレス、または Application Load Balancer をターゲットグループに登録できます。これは、ロードバランサーを作成するためのオプションのステップです。ただし、ターゲットを登録して、ロードバランサーがトラフィックをターゲットにルーティングできるようにする必要があります。

1. [ターゲットの登録] ページで、次のように 1 つ以上のターゲットを追加します。
   - ターゲットタイプがインスタンスである場合は、インスタンスを選択し、[保留中として以下を含める] を選択します。
   - ターゲットタイプが IP アドレスの場合は、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。
   - ターゲットタイプが Application Load Balancer の場合、Application Load Balancer を選択します。
2. [ターゲットグループの作成] を選択します。

ステップ 3: ロードバランサーとリスナーの設定

Network Load Balancer を作成するには、まず、名前、スキーム、IP アドレスのタイプなど、ロードバランサーの基本設定情報を指定する必要があります。次に、ネットワークに関する情報と 1 つ以上のリスナーを指定します。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。これは、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を使用して設定します。サポートされるプロトコルとポートの詳細については、「リスナーの設定」を参照してください。

コンソールを使用してロードバランサーとリスナーを設定するには

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、ロードバランサーの名前を入力します。たとえば、**`my-nlb`** と指定します。Network Load Balancer の名前は、リージョンの Application Load Balancer と Network Load Balancer 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ロードバランサーは、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。内部ロードバランサーは、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. [P アドレスタイプ] については、クライアントがロードバランサーとの通信に IPv4 アドレスを使用する場合は [IPv4] を、クライアントがロードバランサーとの通信に 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] では、EC2 インスタンスで使用したのと同じ VPC を選択します。

      [スキーム] で [インターネット向け] を選択した場合は、インターネットゲートウェイを持つ VPC だけを選択できます。

   b. [マッピング] で、1 つまたは複数のアベイラビリティーゾーンと対応するサブネットを選択します。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が向上します。自分と共有されているサブネットを指定できます。

      インターネット向けロードバランサーの場合は、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。または、内部ロードバランサーの場合は、 で割り当てるのではなく、各サブネットの IPv4 範囲からプライベート IP アドレス AWS を割り当てることができます。
7. セキュリティグループでは、VPC のデフォルトのセキュリティグループがあらかじめ選択されています。必要に応じて、他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する]を選択して、セキュリティニーズを満たすように作成します。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**
>
> この時点でロードバランサー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。

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. [デフォルトアクション] で、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。以前にターゲットグループを作成していない場合は、ここでターゲットグループを作成する必要があります。[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 (オプション)。

   c. (オプション) タグを追加して、リスナーを分類します。

   d. [セキュアなリスナー設定] (TLS リスナーでのみ使用可能) で、次の操作を行います。

      i. [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。

      ii. [ALPN ポリシー] の場合は、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。

      iii. [デフォルトの SSL 証明書] で、[ACM から] (推奨) を選択し、証明書を選択します。選択できる証明書がない場合は、証明書を ACM にインポートするか、ACM を使用して証明書をプロビジョニングできます。詳細については、AWS Certificate Manager ユーザーガイドの「証明書の発行と管理」を参照してください。

9. (オプション) ロードバランサーでアドオンサービスを使用できます。例えば、 に アクセラレータAWS Global Acceleratorーを作成し、ロードバランサーを アクセラレーターに関連付けるように選択できます。アクセラレーター名には、a ~ z、A ~ Z、0 ~ 9、. (ピリオド) 、- (ハイフン) の文字を使用できます。アクセラレーターを作成したら、AWS Global Acceleratorコンソールに移動して設定を完了します。詳細については、「ロードバランサーの作成時にアクセラレーターを追加する」を参照してください。

10. タグ

    (オプション) タグを追加して、ロードバランサーを分類します。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、ロードバランサーにいくつかのデフォルト属性が適用されます。ロードバランサーの作成後に、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。

ステップ 4: ロードバランサーのテスト

ロードバランサーを作成したら、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、ロードバランサーが 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。ロードバランサーを削除するには、Network Load Balancer を削除する 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

1. ロードバランサーが作成されたら、[Close] 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、ロードバランサーをテストできます。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. 新しいロードバランサーを選択します。
7. ロードバランサーの DNS 名をコピーします (例: my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com)。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。

Network Load Balancer の IP アドレスの種類

Network Load Balancer は、クライアントが IPv4 アドレスのみ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、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。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。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

- ロードバランサーの作成時に IP アドレスの種類を設定し、いつでも更新できます。
- ロードバランサーに指定する Virtual Private Cloud (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。
- ロードバランサーには、TCP リスナーと TLS リスナーのみが必要です。
- ロードバランサー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。
- ロードバランサー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。

作成時に IP アドレスの種類を設定するには

ロードバランサーの作成 の説明に従って設定を行います。

IP アドレスを更新するには、コンソールを使用して入力します。

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. [変更の保存] をクリックします。

を使用して IP アドレスタイプを更新するには AWS CLI

set-ip-address-type コマンドを使用します。

Network Load Balancer のセキュリティグループ

セキュリティグループを Network Load Balancer に関連付けて、ロードバランサーロードバランサーへのインバウント/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。ロードバランサー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ロードバランサーリスナーに到達し、すべてのトラフィックがロードバランサー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、クライアントはロードバランサー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、ロードバランサーに対して クライアント IP の保存 を有効にしている場合でも、ターゲットはロードバランサ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ロードバランサーに関連付けることはできません。ロードバランサーを作成するときに、セキュリティグループをロードバランサー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、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、ロードバランサー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。これが内部のロード バランサーの場合には、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ロードバランサーの場合は、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。Amazon EC2

アウトバウンド

[プロトコル]	デスティネーション	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: ロードバランサー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ロードバランサー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-111112222233333	########	ターゲットポートのロードバランサーからのインバウンドトラフィックを許可します
protocol	sg-111112222233333	#######	ヘルスチェックポートでロードバランサー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]	デスティネーション	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

関連付けられたセキュリティグループの更新

ロードバランサーの作成時に少なくとも 1 つのセキュリティグループをロードバランサーに関連付けていた場合は、そのロードバランサーのセキュリティグループをいつでも更新できます。

コンソールを使用してセキュリティグループの更新するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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ロードバランサーに関連付けるには、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ロードバランサーから削除するには、そのセキュリティグループを選択解除します。
6. [変更の保存] をクリックします。

を使用してセキュリティグループを更新するには AWS CLI

set-security-groups コマンドを使用します。

セキュリティ設定の更新

デフォルトでは、ロードバランサー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、これらのルールは AWS PrivateLink、重複する IP アドレスから発信される可能性がある を介してロードバランサーに送信されるトラフィックに適用したくない場合があります。この場合、 を介してロードバランサー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ロードバランサーを設定できます AWS PrivateLink。

コンソールを使用してセキュリティ設定を更新するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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 をクリアします。
6. [変更の保存] をクリックします。

を使用してセキュリティ設定を更新するには AWS CLI

set-security-groups コマンドを使用します。

ロードバランサー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、ロードバラ

ンサー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モニタリングします。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、ロードバランサー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 のタグ

タグを使用すると、さまざまな方法でロードバランサーを分類できます。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。

各ロードバランサーに対して複数のタグを追加できます。すでにロードバランサー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、ロードバランサー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 用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。

コンソールを使用して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 (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. [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 [Remove] (削除) ボタンを選択します。
8. 完了したら、[変更を保存] を選択します。

を使用してロードバランサーのタグを更新するには AWS CLI

add-tags コマンドと remove-tags コマンドを使用します。

Network Load Balancer を削除する

ロードバランサーが利用可能になると、ロードバランサー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。不要になったロードバランサーは削除できます。ロードバランサーが削除されると、ロードバランサーの課金も停止されます。

削除保護が有効になった場合、ロードバランサー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ロードバランサーは削除できません。たとえば、ロードバランサーが VPC エンドポイントサービスに関連付けられている場合、関連付けられたロードバランサーを削除するには、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。

ロードバランサーを削除すると、そのリスナーも削除されます。ロードバランサーを削除しても、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、「ターゲットグループの削除」を参照してください。

コンソールを使用してロードバランサーを削除するには

1. ロードバランサー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、ロードバランサーを削除します。

   例：

   - 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 (https://console.aws.amazon.com/ec2/) を開きます。
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. ロードバランサー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。

を使用してロードバランサーを削除するには AWS CLI

delete-load-balancer コマンドを使用します。

ゾーンシフト

ゾーンシフトは Amazon Route 53 Application Recovery Controller (Route 53 ARC) の機能です。ゾーンシフトを使用すると、1 回のアクションでロードバランサー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、ロードバランサーは、影響を受けるアベイラビリティーゾーンへのリソースのトラフィックの送信を停止します。Route 53 ARC は、このゾーンシフトをすぐに作成します。ただし、影響を受けるアベイラビリティーゾーンで進行中の既存の接続が完了するまでには、通常は数分程度の短い時間がかかる場合があります。詳細については、「Amazon Route 53 Application Recovery Controller デベロッパーガイド」の「How a zonal shift works: health checks and zonal IP addresses」(ゾーンシフトの仕組み: ヘルスチェックとゾーン IP アドレス) を参照してください。

ゾーンシフトは、クロスゾーン負荷分散がオフになっている Application Load Balancer と Network Load Balancer でのみサポートされます。クロスゾーンロードバランサーをオンにすると、ゾーンシフトを開始できなくなります。詳細については、「Amazon Route 53 Application Recovery Controller Developer Guide」の「Resources supported for zonal shifts」(ゾーンシフトでサポートされるリソース) を参照してください。

ゾーンシフトを使用する前に、以下を確認してください。

- ゾーンシフトでは、クロスゾーンロードバランサーはサポートされていません。この機能を使用するには、クロスゾーンロードバランシングをオフにする必要があります。
- Application Load Balancer を AWS Global Acceleratorでアクセラレータエンドポイントとして使用する場合、ゾーンシフトはサポートされません。
- 1 つのアベイラビリティーゾーンに対してのみ、特定のロードバランサーのゾーンシフトを開始できます。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 からゾーンロードバランサーの IP アドレスを事前に削除します。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。ロードバランサーのクロスゾーンロードバランシングがオフになっていて、ゾーンシフトを使用してゾーンロードバランサーの IP アドレスを削除すると、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。
- Application Load Balancer が Network Load Balancer のターゲットである場合は、常に Network Load Balancer からゾーンシフトを開始します。Application Load Balancer からゾーンシフトを開始すると、Network Load Balancer はシフトを認識せず、引き続き Application Load Balancer にトラフィックを送信します。

詳細については、「Amazon Route 53 Application Recovery Controller Developer Guide」の「Best practices with Route 53 ARC zonal shifts」(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。

ゾーンシフトを開始する

この手順のステップでは、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。Route 53 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 Route 53 Application Recovery Controller Developer Guide」の「Starting a zonal shift」(ゾーンシフトの開始) を参照してください。

コンソールを使用してゾーンシフトを開始するには

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. ロードバランサー名を選択します。
4. [Integrations] (統合) タブの [Route 53 Application Recovery Controller] (Route 53 Application Recovery Controller) で、[Start zonal shift] (ゾーンシフトの開始) 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。必要に応じて、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. [開始] を選択します。

を使用してゾーンシフトを開始するには AWS CLI

プログラムによるゾーンシフトの操作については、「Zonal Shift API Reference Guide」(ゾーンシフト API リファレンスガイド) を参照してください。

ゾーンシフトの更新

この手順のステップでは、Amazon EC2 コンソールでゾーンシフトを更新する方法について説明します。Amazon Route 53 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 Route 53 Application Recovery Controller Developer Guide」の「Update a zonal shift」(ゾーンシフトの更新) を参照してください。

コンソールを使用してゾーンシフトを更新するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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ロードバランサー名を選択します。
4. [Integrations] (統合) タブの [Route 53 Application Recovery Controller] で、[Update zonal shift] (ゾーンシフトの更新) を選択します。

   これにより、Route 53 ARC コンソールが開き、更新が続行されます。
5. [Set zonal shift expiration time] (ゾーンシフトの有効期限の設定) で、オプションで有効期限を選択または入力します。
6. [Comment] (コメント) には、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。

を使用してゾーンシフトを更新するには AWS CLI

プログラムによるゾーンシフトの操作については、「Zonal Shift API Reference Guide」(ゾーンシフト API リファレンスガイド) を参照してください。

ゾーンシフトのキャンセル

この手順のステップでは、Amazon EC2 コンソールで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 Route 53 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 Route 53 Application Recovery Controller Developer Guide」の「Cancel a zonal shift」(ゾーンシフトのキャンセル) を参照してください。

コンソールを使用してゾーンシフトをキャンセルするには

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ロードバランサー名を選択します。
4. [Integrations] (統合) タブの [Route 53 Application Recovery Controller] で、[Cancel zonal shift] (ゾーンシフトのキャンセル) を選択します。

   これにより、Route 53 ARC コンソールが開き、キャンセルが続行されます。
5. [Cancel zonal shift] (ゾーンシフトをキャンセル) を選択します。
6. ダイアログボックスで、[Confirm] (確認) を選択します。

を使用してゾーンシフトをキャンセルするには AWS CLI

プログラムによるゾーンシフトの操作については、「Zonal Shift API Reference Guide」(ゾーンシフト API リファレンスガイド) を参照してください。

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
- 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 1 つだけデプロイする必要があります。詳細については、「Network Load Balancer の TLS リスナー」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。

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。

Dualstack Network Load Balancer の場合、サポートされているプロトコルは、TCP と TLS だけです。

リスナー WebSockets で を使用できます。

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。

リスナールール

リスナーを作成するときは、ルーティングリクエストのルールを指定します。このルールは、指定されたターゲットグループにリクエストを転送します。このルールを更新するには、「Network Load Balancer のリスナーを更新する」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

- リスナールール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、「サーバー証明書」を参照してください。

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。

コンソールを使用してリスナーを追加するには

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. [Listeners] (リスナー) タブで、[Add listener] (リスナーの追加) を選択します。
5. [Protocol] (プロトコル) で、[TCP]、[UDP]、[TCP_UDP]、または [TLS] 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。Dualstack Network Load Balancer の場合、サポートされているプロトコルは、TCP と TLS だけです。
6. [Default action] (デフォルトアクション) で、利用可能なターゲットグループを選択します。
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] [Default SSL certificate (デフォルトの SSL 証明書)] で、次のいずれかを実行します。
   - 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、証明書を選択します。
   - IAM を使用して証明書をアップロードした場合は、[IAM から] を選択し、証明書を選択します。
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. [Add] (追加) を選択します。
11. (TLS リスナー) SNI プロトコルで使用するオプションの証明書リストを追加するには、証明書リストに証明書を追加する を参照してください。

を使用してリスナーを追加するには AWS CLI

リスナーを作成するには、create-listener コマンドを使用します。

Network Load Balancer の TLS 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、TLS 再ネゴシエーションまたは相互 TLS 認証 (mTL) 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。

TLS リスナーを作成するには、リスナーの追加 を参照してください。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。

サーバー証明書

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、認証機関 (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」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 CSR 署名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 。

内容

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、厳密に 1 つ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。

- 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。

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。必要に応じて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:

- ELBSecurityPolicy-TLS13-1-2-2021-06 ポリシーは、 を使用して作成された TLS リスナーのデフォルトのセキュリティポリシーです AWS Management Console。
  - TLS 1.3 を含み、TLS 1.2 と下位互換性がある`ELBSecurityPolicy-TLS13-1-2-2021-06`セキュリティポリシーをお勧めします。
- ELBSecurityPolicy-2016-08 ポリシーは、 を使用して作成された TLS リスナーのデフォルトのセキュリティポリシーです AWS CLI。
- フロントエンド接続に使用されるセキュリティポリシーを選択できますが、バックエンド接続には選択できません。
  - バックエンド接続では、TLS リスナーが TLS 1.3 セキュリティポリシーを使用している場合、ELBSecurityPolicy-TLS13-1-0-2021-06 セキュリティポリシーが使用されます。それ以外の場合、バックエンド接続には ELBSecurityPolicy-2016-08 セキュリティポリシーが使用されます。

- 特定の TLS プロトコルバージョンの無効化を必要とするコンプライアンスおよびセキュリティ標準を満たすため、または非推奨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には、いずれかのELBSecurityPolicy-TLS-セキュリティポリシーを使用できます。Network Load Balancer に送信された TLS リクエストに関する情報のアクセスログを有効にしたり、TLS トラフィックパターンを分析したり、セキュリティポリシーのアップグレードを管理したり、問題をトラブルシューティングしたりできます。ロードバランサーのアクセスログ記録を有効にし、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。
- 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 AWS アカウント を使用することで、 および 全体の AWS Organizations 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SCPs)」を参照してください。

TLS1.3 セキュリティポリシー

Elastic Load Balancing では、Network Load Balancer に次の TLS 1.3 セキュリティポリシーが用意されています。

- ELBSecurityPolicy-TLS13-1-2-2021-06 (推奨)
- ELBSecurityPolicy-TLS13-1-2-Res-2021-06
- ELBSecurityPolicy-TLS13-1-2-Ext1-2021-06
- ELBSecurityPolicy-TLS13-1-2-Ext2-2021-06
- ELBSecurityPolicy-TLS13-1-1-2021-06
- ELBSecurityPolicy-TLS13-1-0-2021-06
- ELBSecurityPolicy-TLS13-1-3-2021-06

FIPS セキュリティポリシー

連邦情報処理規格 (FIPS) は、機密情報を保護する暗号化モジュールのセキュリティ要件を指定する米国およびカナダ政府の規格です。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。

すべての FIPS ポリシーは、AWS-LC FIPS 検証済み暗号化モジュールを活用します。詳細については、NIST 暗号化モジュール検証プログラムサイトの AWS-LC 暗号化モジュールページを参照してください。

Elastic Load Balancing では、Network Load Balancer に対して次の FIPS セキュリティポリシーが用意されています。

- ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04
- ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04
- ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 (推奨)
- ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04
- ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04
- ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04
- ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04
- ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04

FS がサポートするポリシー

Elastic Load Balancing では、Network Load Balancer に対して以下の FS (フォワードシークレット) がサポートするセキュリティポリシーが用意されています。

- 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

TLS 1.0 - 1.2 セキュリティポリシー

Elastic Load Balancing では、Network Load Balancer に次の TLS 1.0 ~ 1.2 セキュリティポリシーが用意されています。

- ELBSecurityPolicy-TLS-1-2-Ext-2018-06
- ELBSecurityPolicy-TLS-1-2-2017-01
- ELBSecurityPolicy-TLS-1-1-2017-01
- ELBSecurityPolicy-2016-08
- ELBSecurityPolicy-TLS-1-0-2015-04
- ELBSecurityPolicy-2015-05 (と同じELBSecurityPolicy-2016-08)

TLS プロトコルと暗号

TLS 1.3

次の表は、使用可能な TLS 1.3 セキュリティポリシーでサポートされている TLS プロトコルと暗号を示しています。

注： セキュリティポリシー行のポリシー名からELBSecurityPolicy-プレフィックスが削除されました。

例： セキュリティポリシーELBSecurityPolicy-TLS13-1-2-2021-06は として表示されますTLS13-1-2-2021-06。

セキュリティポリシー	TLS13-1-2-2021-06	TLS13-1-3-2021-06	TLS13-1-2-Res-2021-06	TLS13-1-2-Ext2-2021-06	TLS13-1-2-Ext1-2021-06	TLS13-1-1-2021-06	TLS13-1-0-2021-06
TLS Protocols							
Protocol-TLSv1							✓
Protocol-TLSv1.1						✓	✓
Protocol-TLSv1.2	✓		✓	✓	✓	✓	✓
Protocol-TLSv1.3	✓	✓	✓	✓	✓	✓	✓
TLS Ciphers							
TLS_AES_128_GCM_SHA256	✓	✓	✓	✓	✓	✓	✓

セキュリティポリシー	TLS13-1-2-2021-06	TLS13-1-3-2021-06	TLS13-1-2-Res-2021-06	TLS13-1-2-Ext2-2021-06	TLS13-1-2-Ext1-2021-06	TLS13-1-1-2021-06	TLS13-1-0-2021-06
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	✓			✓	✓	✓	✓

セキュリティポリシー	TLS13-1-2-2021-06	TLS13-1-3-2021-06	TLS13-1-2-Res-2021-06	TLS13-1-2-Ext2-2021-06	TLS13-1-2-Ext1-2021-06	TLS13-1-1-2021-06	TLS13-1-0-2021-06
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	✓			✓	✓	✓	✓

セキュリティポリシー	TLS13-1-2-2021-06	TLS13-1-3-2021-06	TLS13-1-2-Res-2021-06	TLS13-1-2-Ext2-2021-06	TLS13-1-2-Ext1-2021-06	TLS13-1-1-2021-06	TLS13-1-0-2021-06
ECDHE-RSA-AES256-SHA				✓		✓	✓
ECDHE-ECDSA-AES256-SHA				✓		✓	✓
AES128-GCM-SHA256				✓	✓	✓	✓
AES128-SHA256				✓	✓	✓	✓
AES128-SHA				✓		✓	✓
AES256-GCM-SHA384				✓	✓	✓	✓
AES256-SHA256				✓	✓	✓	✓
AES256-SHA				✓		✓	✓

CLI を使用して TLS 1.3 ポリシーを使用する TLS リスナーを作成するには

任意の TLS 1.3 セキュリティポリシー で create-listener コマンドを使用します。 ???

この例では、 ELBSecurityPolicy-TLS13-1-2-2021-06 セキュリティポリシーを使用しています。

```
aws elbv2 create-listener --name my-listener \
--protocol TLS --port 443 \
--ssl-policy ELBSecurityPolicy-TLS13-1-2-2021-06
```

CLI を使用して TLS 1.3 ポリシーを使用するように TLS リスナーを変更するには

TLS 1.3 セキュリティポリシー で modify-listener コマンドを使用します。 ???

この例では、 ELBSecurityPolicy-TLS13-1-2-2021-06 セキュリティポリシーを使用しています。

```
aws elbv2 modify-listener \
--listener-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/abcdef01234567890/1234567890abcdef0 \
--ssl-policy ELBSecurityPolicy-TLS13-1-2-2021-06
```

CLI を使用してリスナーが使用するセキュリティポリシーを表示するには

リスナーの で describe-listener コマンドarnを使用します。

```
aws elbv2 describe-listener \
--listener-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/abcdef01234567890/1234567890abcdef0
```

CLI を使用して TLS 1.3 セキュリティポリシーの設定を表示するには

任意の TLS 1.3 セキュリティポリシー で describe-ssl-policies コマンドを使用します。 ???

この例では、 ELBSecurityPolicy-TLS13-1-2-2021-06 セキュリティポリシーを使用しています。

```
aws elbv2 describe-ssl-policies \
--names ELBSecurityPolicy-TLS13-1-2-2021-06
```

FIPS

> **Important**
>
> ポリシー ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 および 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。FIPS140 モジュールを使用して FIPS 暗号化を使用していますが、TLS 設定に関する最新の NIST ガイダンスに準拠していない可能性があります。

次の表は、使用可能な FIPS セキュリティポリシーでサポートされている TLS プロトコルと暗号を示しています。

注： セキュリティポリシー行のポリシー名からELBSecurityPolicy-プレフィックスが削除されました。

例： セキュリティポリシーELBSecurityPolicy-TLS13-1-2-FIPS-2023-04は として表示されますTLS13-1-2-FIPS-2023-04。

セキュリティポリシー	TLS13-1-3-FIPS-2023-04	TLS13-1-2-Res-FIPS-2023-04	TLS13-1-2-FIPS-2023-04	TLS13-1-2-Ext0-FIPS-2023-04	TLS13-1-2-Ext1-FIPS-2023-04	TLS13-1-2-Ext2-FIPS-2023-04	TLS13-1-1-FIPS-2023-04	TLS13-1-0-FIPS-2023-04
TLS Protocols								
Protocol-TLSv1								✓
Protocol-TLSv1.1							✓	✓
Protocol-TLSv1.2	✓	✓	✓	✓	✓	✓	✓	✓

セキュリティポリシー	TLS13-1-3-FIPS-2023-04	TLS13-1-2-Res-FIPS-2023-04	TLS13-1-2-FIPS-2023-04	TLS13-1-2-Ext0-FIPS-2023-04	TLS13-1-2-Ext1-FIPS-2023-04	TLS13-1-2-Ext2-FIPS-2023-04	TLS13-1-1-FIPS-2023-04	TLS13-1-0-FIPS-2023-04
Protocol-TLSv1.3	✓	✓	✓	✓	✓	✓	✓	✓
TLS Ciphers								
TLS_AES_128_GCM_SHA256	✓	✓	✓	✓	✓	✓	✓	✓
TLS_AES_256_GCM_SHA384	✓	✓	✓	✓	✓	✓	✓	✓
ECDHE-ECDSA-AES128-GCM-SHA256		✓	✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES128-GCM-SHA256		✓	✓	✓	✓	✓	✓	✓

セキュリティポリシー	TLS13-1-3-FIPS-2023-04	TLS13-1-2-Res-FIPS-2023-04	TLS13-1-2-FIPS-2023-04	TLS13-1-2-Ext0-FIPS-2023-04	TLS13-1-2-Ext1-FIPS-2023-04	TLS13-1-2-Ext2-FIPS-2023-04	TLS13-1-1-FIPS-2023-04	TLS13-1-0-FIPS-2023-04
ECDHE-ECDSA-AES128-SHA256			✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES128-SHA256			✓	✓	✓	✓	✓	✓
ECDHE-ECDSA-AES128-SHA				✓		✓	✓	✓
ECDHE-RSA-AES128-SHA				✓		✓	✓	✓

セキュリティポリシー	TLS13-1-3-FIPS-2023-04	TLS13-1-2-Res-FIPS-2023-04	TLS13-1-2-FIPS-2023-04	TLS13-1-2-Ext0-FIPS-2023-04	TLS13-1-2-Ext1-FIPS-2023-04	TLS13-1-2-Ext2-FIPS-2023-04	TLS13-1-1-FIPS-2023-04	TLS13-1-0-FIPS-2023-04
ECDHE-ECDSA-AES256-GCM-SHA384		✓	✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES256-GCM-SHA384		✓	✓	✓	✓	✓	✓	✓
ECDHE-ECDSA-AES256-SHA384			✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES256-SHA384			✓	✓	✓	✓	✓	✓

セキュリティポリシー	TLS13-1-3-FIPS-2023-04	TLS13-1-2-Res-FIPS-2023-04	TLS13-1-2-FIPS-2023-04	TLS13-1-2-Ext0-FIPS-2023-04	TLS13-1-2-Ext1-FIPS-2023-04	TLS13-1-2-Ext2-FIPS-2023-04	TLS13-1-1-FIPS-2023-04	TLS13-1-0-FIPS-2023-04
ECDHE-RSA-AES256-SHA				✓		✓	✓	✓
ECDHE-ECDSA-AES256-SHA				✓		✓	✓	✓
AES128-GCM-SHA256					✓	✓	✓	✓
AES128-SHA256					✓	✓	✓	✓
AES128-SHA						✓	✓	✓
AES256-GCM-SHA384					✓	✓	✓	✓
AES256-SHA256					✓	✓	✓	✓

セキュリティポリシー	TLS13-1-3-FIPS-2023-04	TLS13-1-2-Res-FIPS-2023-04	TLS13-1-2-FIPS-2023-04	TLS13-1-2-Ext0-FIPS-2023-04	TLS13-1-2-Ext1-FIPS-2023-04	TLS13-1-2-Ext2-FIPS-2023-04	TLS13-1-1-FIPS-2023-04	TLS13-1-0-FIPS-2023-04
AES256-SHA						✓	✓	✓

CLI を使用して FIPS ポリシーを使用する TLS リスナーを作成するには

任意の FIPS セキュリティポリシー で create-listener コマンドを使用します。 ???

この例では、 ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 セキュリティポリシーを使用しています。

```
aws elbv2 create-listener --name my-listener \
--protocol TLS --port 443 \
--ssl-policy ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04
```

CLI を使用して FIPS ポリシーを使用するように TLS リスナーを変更するには

任意の FIPS セキュリティポリシー で modify-listener コマンドを使用します。 ???

この例では、 ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 セキュリティポリシーを使用しています。

```
aws elbv2 modify-listener \
--listener-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/abcdef01234567890/1234567890abcdef0 \
--ssl-policy ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04
```

CLI を使用してリスナーが使用するセキュリティポリシーを表示するには

リスナーの で describe-listener コマンドarnを使用します。

```
aws elbv2 describe-listener \
```

```
--listener-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/abcdef01234567890/1234567890abcdef0
```

CLI を使用して FIPS セキュリティポリシーの設定を表示するには

任意の FIPS セキュリティポリシー で describe-ssl-policies コマンドを使用します。 ???

この例では、 ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 セキュリティポリシーを使用しています。

```
aws elbv2 describe-ssl-policies \
--names ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04
```

FS

次の表は、使用可能な FS がサポートするセキュリティポリシーでサポートされている TLS プロトコルと暗号を示しています。

注： セキュリティポリシー行のポリシー名からELBSecurityPolicy-プレフィックスが削除されました。

例： セキュリティポリシーELBSecurityPolicy-FS-2018-06は として表示されますFS-2018-06。

セキュリティポリシー	Default	FS-1-2-Res-2020-10	FS-1-2-Res-2019-08	FS-1-2-2019-08	FS-1-1-2019-08	FS-2018-06
TLS Protocols						
Protocol-TLSv1	✓					✓
Protocol-TLSv1.1	✓				✓	✓

セキュリティポリシー	Default	FS-1-2-Res-2020-10	FS-1-2-Res-2019-08	FS-1-2-2019-08	FS-1-1-2019-08	FS-2018-06
Protocol-TLSv1.2	✓	✓	✓	✓	✓	✓
TLS Ciphers						
ECDHE-ECDSA-AES128-GCM-SHA256	✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES128-GCM-SHA256	✓	✓	✓	✓	✓	✓
ECDHE-ECDSA-AES128-SHA256	✓		✓	✓	✓	✓
ECDHE-RSA-AES128-SHA256	✓		✓	✓	✓	✓

セキュリティポリシー	Default	FS-1-2-Res-2020-10	FS-1-2-Res-2019-08	FS-1-2-2019-08	FS-1-1-2019-08	FS-2018-06
ECDHE-ECDSA-AES128-SHA	✓			✓	✓	✓
ECDHE-RSA-AES128-SHA	✓			✓	✓	✓
ECDHE-ECDSA-AES256-GCM-SHA384	✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES256-GCM-SHA384	✓	✓	✓	✓	✓	✓
ECDHE-ECDSA-AES256-SHA384	✓		✓	✓	✓	✓

セキュリティポリシー	Default	FS-1-2-Res-2020-10	FS-1-2-Res-2019-08	FS-1-2-2019-08	FS-1-1-2019-08	FS-2018-06
ECDHE-RSA-AES256-SHA384	✓		✓	✓	✓	✓
ECDHE-RSA-AES256-SHA	✓			✓	✓	✓
ECDHE-ECDSA-AES256-SHA	✓			✓	✓	✓
AES128-GCM-SHA256	✓					
AES128-SHA256	✓					
AES128-SHA	✓					
AES256-GCM-SHA384	✓					

セキュリティポリシー	Default	FS-1-2-Res-2020-10	FS-1-2-Res-2019-08	FS-1-2-2019-08	FS-1-1-2019-08	FS-2018-06
AES256-SHA256	✓					
AES256-SHA	✓					

CLI を使用して FS がサポートするポリシーを使用する TLS リスナーを作成するには

FS がサポートするセキュリティポリシー で create-listener コマンドを使用します。 ???

この例では、 ELBSecurityPolicy-FS-2018-06 セキュリティポリシーを使用しています。

```
aws elbv2 create-listener --name my-listener \
--protocol TLS --port 443 \
--ssl-policy ELBSecurityPolicy-FS-2018-06
```

CLI を使用して FS がサポートするポリシーを使用するように TLS リスナーを変更するには

FS がサポートするセキュリティポリシー で modify-listener コマンドを使用します。 ???

この例では、 ELBSecurityPolicy-FS-2018-06 セキュリティポリシーを使用しています。

```
aws elbv2 modify-listener \
--listener-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/abcdef01234567890/1234567890abcdef0 \
--ssl-policy ELBSecurityPolicy-FS-2018-06
```

CLI を使用してリスナーが使用するセキュリティポリシーを表示するには

リスナーの で describe-listener コマンドarnを使用します。

```
aws elbv2 describe-listener \
--listener-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/abcdef01234567890/1234567890abcdef0
```

CLI を使用して FS がサポートするセキュリティポリシーの設定を表示するには

FS がサポートするセキュリティポリシー で describe-ssl-policies コマンドを使用します。 ???

この例では、 ELBSecurityPolicy-FS-2018-06 セキュリティポリシーを使用しています。

```
aws elbv2 describe-ssl-policies \
--names ELBSecurityPolicy-FS-2018-06
```

TLS 1.0 - 1.2

次の表は、使用可能な TLS 1.0-1.2 セキュリティポリシーでサポートされている TLS プロトコルと暗号を示しています。

注： セキュリティポリシー行のポリシー名からELBSecurityPolicy-プレフィックスが削除されました。

例： セキュリティポリシーELBSecurityPolicy-TLS-1-2-Ext-2018-06は として表示されますTLS-1-2-Ext-2018-06。

セキュリティポリシー	Default	TLS-1-2-Ext-2018-06	TLS-1-2-2017-01	TLS-1-1-2017-01	TLS-1-0-2015-04*
TLS Protocols					
Protocol-TLSv1	✓				✓
Protocol-TLSv1.1	✓			✓	✓

セキュリティポリシー	Default	TLS-1-2-Ext-2018-06	TLS-1-2-2017-01	TLS-1-1-2017-01	TLS-1-0-2015-04*
Protocol-TLSv1.2	✓	✓	✓	✓	✓
TLS Ciphers					
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	✓	✓		✓	✓

セキュリティポリシー	Default	TLS-1-2-Ext-2018-06	TLS-1-2-2017-01	TLS-1-1-2017-01	TLS-1-0-2015-04*
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	✓	✓	✓	✓	✓

セキュリティポリシー	Default	TLS-1-2-Ext-2018-06	TLS-1-2-2017-01	TLS-1-1-2017-01	TLS-1-0-2015-04*
AES128-SHA	✓	✓		✓	✓
AES256-GCM-SHA384	✓	✓	✓	✓	✓
AES256-SHA256	✓	✓	✓	✓	✓
AES256-SHA	✓	✓		✓	✓
DES-CBC3-SHA					✓

* DES-CBC3-SHA 暗号 (弱い暗号) を必要とするレガシークライアントをサポートする必要がない限り、このポリシーは使用しないでください。

CLI を使用して TLS 1.0-1.2 ポリシーを使用する TLS リスナーを作成するには

create-listener コマンドは、TLS 1.0-1.2 でサポートされているセキュリティポリシー で使用します。

この例では、 ELBSecurityPolicy-2016-08 セキュリティポリシーを使用しています。

```
aws elbv2 create-listener --name my-listener \
--protocol TLS --port 443 \
--ssl-policy ELBSecurityPolicy-2016-08
```

CLI を使用して TLS 1.0 ~ 1.2 ポリシーを使用するように TLS リスナーを変更するには

TLS 1.0-1.2 でサポートされているセキュリティポリシー で modify-listener コマンドを使用します。

この例では、 ELBSecurityPolicy-2016-08 セキュリティポリシーを使用しています。

```
aws elbv2 modify-listener \
--listener-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/abcdef01234567890/1234567890abcdef0 \
--ssl-policy ELBSecurityPolicy-2016-08
```

CLI を使用してリスナーが使用するセキュリティポリシーを表示するには

リスナーの で describe-listener コマンドarnを使用します。

```
aws elbv2 describe-listener \
--listener-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:listener/app/my-load-balancer/abcdef01234567890/1234567890abcdef0
```

CLI を使用して TLS 1.0-1.2 セキュリティポリシーの設定を表示するには

TLS 1.0-1.2 でサポートされているセキュリティポリシー で describe-ssl-policies コマンドを使用します。

この例では、 ELBSecurityPolicy-2016-08 セキュリティポリシーを使用しています。

```
aws elbv2 describe-ssl-policies \
--names ELBSecurityPolicy-2016-08
```

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 ポリシーの更新」を参照してください。

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 または UDP 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 または UDP 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。

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。

コンソールを使用してリスナーを更新するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[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. [編集] を選択します。
6. (オプション) 必要に応じて、[プロトコル] および [ポート] の指定された値を変更します。
7. (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。
8. (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。
9. [変更の保存] を選択します。

を使用してリスナーを更新するには AWS CLI

modify-listener コマンドを使用します。

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 ポリシーの更新

デフォルトの証明書の置き換え

次の手順で TLS リスナーの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」を参照してください。

コンソールを使用して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 (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. [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. [Default SSL certificate (デフォルトの SSL 証明書)] で、次のいずれかを実行します。
   - 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、証明書を選択します。
   - IAM を使用して証明書をアップロードした場合は、[IAM から] を選択し、証明書を選択します。
6. [変更を保存] を選択します。

を使用してデフォルトの証明書を置き換えるには AWS CLI

modify-listener コマンドを使用して、--certificates オプションを指定します。

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。1 つ以上の証明書を追加できます。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オプションで追加できます。詳細については、「証明書リスト」を参照してください。

コンソールを使用して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 (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. [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. リスナーのチェックボックスを選択し、[アクション]、[SNI の SSL 証明書の追加] を選択します。
6. 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。
7. ACM または IAM によって管理されていない証明書がある場合は、[証明書のインポート] を選択し、フォームに記入し、[インポート] を選択します。
8. [保留中の証明書を追加] を選択します。

を使用して証明書リストに証明書を追加するには AWS CLI

add-listener-certificatesコマンドを実行します。

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。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。

コンソールを使用して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 (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. [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. リスナーのチェックボックスを選択し、[アクション]、[SNI の SSL 証明書の追加] を選択します。
6. 証明書のチェックボックスを選択して、[Remove (削除)] を選択します。
7. 確認を求められたら、**`confirm`** と入力し、[削除] を選択します。

を使用して証明書リストから証明書を削除するには AWS CLI

remove-listener-certificatesコマンドを実行します。

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、「セキュリティポリシー」を参照してください。

コンソールを使用してセキュリティポリシーを更新するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[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. [編集] を選択します。
6. [Security policy (セキュリティポリシー)] で、セキュリティポリシーを選択します。
7. [変更を保存] を選択します。

を使用してセキュリティポリシーを更新するには AWS CLI

modify-listener コマンドを使用して、--ssl-policy オプションを指定します。

ALPN ポリシーの更新

次の手順を使用して、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。

コンソールを使用して ALPN ポリシーを更新するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[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. [編集] を選択します。

6. [ALPN ポリシー] の場合は、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. [変更を保存] を選択します。

を使用して ALPN ポリシーを更新するには AWS CLI

modify-listener コマンドを使用して、--alpn-policy オプションを指定します。

Network Load Balancer のリスナーを削除する

リスナーの削除はいつでも行うことができます。

コンソールを使用してリスナーを削除するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[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。

を使用してリスナーを削除するには AWS CLI

delete-listener コマンドを使用します。

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、「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

- ターゲットグループのタグ
- ターゲットグループの削除

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
- 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 Virtual Private Cloud (VPC) 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くても man-in-the-middle 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	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

[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 (データベースなど) でアドレス指定できる リソース。
- AWS 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 Amazon 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、「ターゲットとしての Application Load Balancer」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

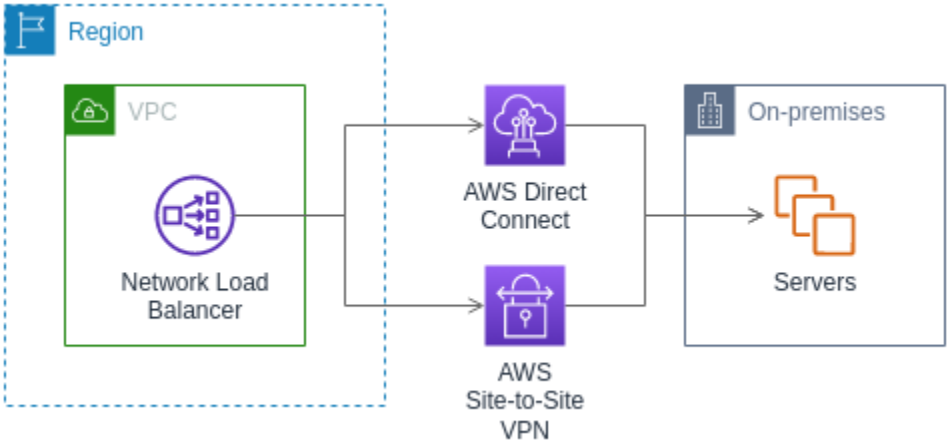
インスタンス ID を使用してターゲットを指定すると、トラフィックはインスタンスのプライマリネットワークインターフェイスで指定されたプライマリプライベート IP アドレスを使用して、インスタンスにルーティングされます。ロードバランサーは、データパケットの宛先 IP アドレスを書き換えてから、ターゲットインスタンスに転送します。

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

AWS 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)

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

の詳細については 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。

Network Load Balancer は、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サポートします。デフォルトで選択されるのは IPv4 です。IPv6 ターゲットグループは、Dualstack Network Load Balancer にのみ関連付けることができます。

考慮事項

- ターゲットグループ内のすべての IP アドレスは、同じ IP アドレスタイプである必要があります。例えば、IPv4 ターゲットを IPv6 ターゲットグループに登録することはできません。
- IPv6 ターゲットグループは、TCP または TLS リスナーを使用した dualstack ロードバランサーのみで使用できます。
- IPv6 ターゲットグループでは、IP およびインスタンスタイプのターゲットがサポートされています。

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。ロードバランサーは、登録プロセスが完了し、設定されたしきい値に関係なく、ターゲットが最初の最初のヘルスチェックに合格するとすぐに、新しく登録され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。 Amazon EC2
- 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 Amazon 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 および TCP_UDP 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

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。

`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。

`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: `use_load_balancer_configuration`。

`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 および TCP_UDP 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。

`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 フェイルアウェイが無効になります。つまり、各ターゲットグループが独立してDNSフェイルオーバーに寄与することになります。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 フェイルアウェイが無効になります。つまり、各ターゲットグループが独立してDNS フェイルオーバーに寄与することになります。デフォルト は 1 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。範囲は 1 からターゲットの最大数です。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: `true`。

`target_health_state.unhealthy.draining_interval_seconds`

異常なターゲットの状態を から に変更するまで Elastic Load Balancing が待機`unhealthy.draining`する時間`unhealthy`。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 の場合にのみ設定できます`false`。

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 の保存を無効にした場合、Network Load Balancer のプライベート IP アドレスは、すべての受信トラフィックのクライアント IP になります。

デフォルトでは、UDP プロトコルと TCP_UDP 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): 有効
- IP タイプのターゲットグループ (TCP、TLS): 無効

要件と考慮事項

- クライアント IP の保存が無効な場合、ターゲットは Network Load Balancer と同じ VPC にある必要があり、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。
- ターゲットが Network Load Balancer と同じ Amazon 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) 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 保存は、 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。AWS PrivateLink トラフィックの送信元 IP 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 ENI または別のNetwork Load Balancer の ENI 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 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。
- 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。
- 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。有効な場合、ターゲットで確認されたソケットの再利用に関連する TCP/IP 接続の制限が発生することがあります。これらの接続制限が発生する可能性があるのは、クライアント、またはクライアントの前面にある NAT デバイスが、複数のロードバランサーノー

ドに同時に接続する際に、同じ送信元 IP アドレスと送信元ポートを使用する場合です。ロードバランサーがこれらの接続を同じターゲットにルーティングする場合、接続は同じ送信元ソケットからの接続のようにターゲットに表示され、それにより接続エラーが発生します。この場合、クライアントは再試行 (接続が失敗した場合)、または再接続 (接続が中断した場合) できます。このタイプの接続エラーは、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すことによって減らすことができます。このタイプの接続エラーは、クライアント IP の保存を無効にするか、クロスゾーン負荷分散を無効にすることで防止できます。

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

コンソールを使用してクライアント IP 保存を設定するには

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. 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. [変更の保存] をクリックします。

を使用してクライアント IP 保存を有効または無効にするには AWS CLI

`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。

たとえば、次のコマンドを使用して、クライアント IP 保存を無効にします。

```
aws elbv2 modify-target-group-attributes --attributes
 Key=preserve_client_ip.enabled,Value=false --target-group-arn ARN
```

出力は次の例のようになります。

```
{
    "Attributes": [
      {
        "Key": "proxy_protocol_v2.enabled",
        "Value": "false"
      },
      {
        "Key": "preserve_client_ip.enabled",
        "Value": "false"
      },
      {
        "Key": "deregistration_delay.timeout_seconds",
        "Value": "300"
      }
    ]
}
```

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。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。

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。

コンソールを使用して登録解除属性を更新するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。

2. ナビゲーションペインの [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. [変更の保存] をクリックします。

を使用して登録解除属性を更新するには AWS CLI

modify-target-group-attributes コマンドを使用します。

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、Network Load Balancer からのものでない可能性があります。

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 および TLS: 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 アドレスが必要な場合は、Proxy Protocol を有効にし、Proxy Protocol ヘッダーからクライアント IP アドレスを取得します。
- UDP および TCP_UDP: 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。

> **Note**
>
> 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。

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。

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)
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。

コンソールを使用してプロキシプロトコル v2 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] を選択します。

を使用してプロキシプロトコル v2 を有効にするには AWS CLI

modify-target-group-attributes コマンドを使用します。

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で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ていません。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 リスナーでサポートされません。

コンソールを使用して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 (https://console.aws.amazon.com/ec2/) を開きます。
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. [変更の保存] をクリックします。

を使用してスティッキーセッションを有効にするには AWS CLI

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

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。

ターゲットグループを作成すると、タグを追加できます。

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、「リスナールール」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、「ターゲットグループへのターゲットの登録」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、「ターゲットグループのヘルスチェック設定を変更する」を参照してください。

コンソールを使用してターゲットグループを作成するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[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. [Choose a target type] (ターゲットタイプの選択) で、インスタンス ID でターゲットを登録する場合は [Instances] (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 addresses] (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。

   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。

   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 アドレスで、プロトコルが TCP または TLS の場合にのみ使用できます。

      IPv6 ターゲットグループをデュアルスタックロードバランサーに関連付ける必要があります。ターゲットグループ内のすべてのターゲットは、同じ IP アドレスタイプである必要があります。ターゲットグループが作成されると、IP アドレスタイプを変更することはできません。

   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、「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。

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] をクリックします。
8. [ターゲットの登録] ページで、次のように 1 つ以上のターゲットを追加します。
   - ターゲットタイプがインスタンスである場合は、インスタンスを選択し、[保留中として以下を含める] を選択します。

     注: IPv6 ターゲットグループに登録する場合、インスタンスにプライマリ IPv6 アドレスが割り当てられている必要があります。
   - ターゲットタイプが IP アドレスの場合は、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。
9. [ターゲットグループの作成] を選択します。

を使用してターゲットグループを作成するには AWS CLI

ターゲットグループを作成するには create-target-group コマンド、ターゲットグループにタグを付けるには add-tags コマンド、ターゲットを追加するには register-targets コマンドを使用します。

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。登録プロセスが完了次第、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。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、「クロスゾーン負荷分散」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。パッシブヘルスチェックは UDP トラフィックではサポートされておらず、維持が有効になっているターゲットグループもサポートされません。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イルオープンします。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、このテストの TCP 接続がロードバランサーから TLS リスナーに登録されているターゲットに送信されることがあります。これらの TCP 接続は、Network Load Balancer の送信元 IP アドレスを持ち、接続にデータパケットが含まれていないため、識別できます。

UDP 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、テストされます。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、UDP 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。UDP 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 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。

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。ヘルスチェックが連続した失敗UnhealthyThreshold数を超えると、ロードバランサーはターゲットをサービス停止にします。ヘルスチェックが連続した成功HealthyThreshold回数を超えると、ロードバランサーはターゲットを稼働状態に戻します。

設定	説明	デフォルト値
HealthCheckプロトコル	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
HealthCheckポート	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。
HealthCheckパス	[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

設定	説明	デフォルト値
		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。 **Important** 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。	30 秒
HealthyThresholdカウント	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
UnhealthyThresholdカウント	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

設定	説明	デフォルト値
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`	ターゲットがヘルスチェックに応答しなかったか、ヘルスチェックに失敗したか、ターゲットが停止状態です。 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。

値	説明
	関連する理由コード : Target.Deregistrat ionInProgress
unhealthy.draining	ターゲットがヘルスチェックに応答しなかったか、ヘルスチェックに失敗して猶予期間に入った。ターゲットは既存の接続をサポートしており、この猶予期間中は新しい接続を受け付けません。 関連する理由コード : Target.FailedHealt hChecks
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、ターゲットグループがリスナールールで使用されていない、またはターゲットが有効化されていないアベイラビリティーゾーンにある。 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

理由コード	説明
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
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

ターゲットのヘルスステータスをチェックする

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。

コンソールを使用してターゲットのヘルスステータスをチェックするには

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. [Details] (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。

を使用してターゲットの状態を確認するには AWS CLI

describe-target-health コマンドを使用します。このコマンドの出力にはターゲットのヘルス状態が含まれます。ステータスの値が `Healthy` 以外の場合は、理由コードも含まれています。

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 Lambda 関数をトリガーし、異常なターゲットに関する詳細を送信します。 step-by-step 手順については、次のブログ記事「ロードバランサーの異常なターゲットの特定」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

ターゲットグループのヘルスチェック設定はいつでも変更できます。

コンソールを使用してターゲットグループのヘルスチェック設定を変更するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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の設定] ページで、必要に応じて設定を変更し、[変更内容の保存] を選択します。

を使用してターゲットグループのヘルスチェック設定を変更するには AWS CLI

modify-target-group コマンドを使用します。

ターゲットグループのクロスゾーンロードバランサー

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。

考慮事項

- Network Load Balancer のクロスゾーン負荷分散を有効にすると、EC2 データ転送料金が適用されます。詳細については、「データエクスポートユーザーガイド」の「データ転送料金について」を参照してください。 AWS
- 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オフの場合は、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オフ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。

ロードバランサーに関連するクロスゾーンロードバランサーを変更する

クロスゾーンロードバランサーは、いつでもロードバランサーレベルでオンまたはオフにすることができます。

コンソールを使用してロードバランサーに関連するクロスゾーンロードバランサーを変更するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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. [変更の保存] をクリックします。

を使用してロードバランサーのクロスゾーン負荷分散を変更するには AWS CLI

`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。

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変更する

ターゲットグループレベルに対する、クロスゾーン負荷分散の設定は、ロードバランサーレベルの設定よりも優先されます。

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オンまたはオフ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。

コンソールを使用して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変更するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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. [変更の保存] をクリックします。

を使用してターゲットグループのクロスゾーン負荷分散を変更するには AWS CLI

`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。

ターゲットグループのヘルス

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。これにより、可用性が向上します。

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 フェイルオーバーでは、ロードバランサーの DNS ホスト名から異常なゾーンの IP アドレスを削除します。ただし、DNS レコードの time-to-live (TTL) の有効期限が切れるまで (60 秒)、ローカルクライアント DNS キャッシュにこれらの 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、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。

ターゲットグループのヘルス設定の変更

ターゲットグループに関連するターゲットグループのヘルス設定は、次のように変更できます。

コンソールを使用してターゲットグループのヘルス設定を変更するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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. クロスゾーンロードバランサーがオンになっているかオフになっているかを確認します。必要に応じてこの設定を更新して、ゾーンに障害が発生した場合に追加のトラフィックを処理するのに十分な容量を確保してください。
6. [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。
7. [Configuration type] (設定タイプ) には、両方のアクションに同じしきい値を設定する [Unified configuration] (統合設定) を選択することをお勧めします。
8. [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。
9. [変更の保存] をクリックします。

を使用してターゲットグループのヘルス設定を変更するには AWS CLI

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 arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
--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=50
```

異常のあるターゲットの接続終了

接続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、異常であると判断された場合、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。接続終了を無効にすると、ターゲットは引き続き異常と見なされ、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に保たれ、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ごとに個別に設定できます。

コンソールを使用して接続終了設定を変更するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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. [変更の保存] をクリックします。

を使用して接続終了設定を変更するには AWS CLI

`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。

異常なドレイニング間隔

> **Important**
> 異常なドレイン間隔を有効にする前に、接続終了を無効にする必要があります。

`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確立された接続を保持します。異常な接続間隔によって、ターゲットが `unhealthy.draining` 状態になるまでに 状態のままになる時間が決まりますunhealthy。異常な接続間隔中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healthy再び になります。登録解除がトリガーされると、ターゲットの状態は になりdraining、登録解除の遅延タイムアウトが開始されます。

異常なドレイニング間隔は、ターゲットグループごとに個別に設定できます。

コンソールを使用して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 (https://console.aws.amazon.com/ec2/) を開きます。
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. 「ターゲットの異常な状態管理」で、ターゲットが異常な状態になったときに接続を終了してください。
6. 異常なドレイニング間隔 の値を入力します。
7. [変更の保存] をクリックします。

を使用して異常なドレイニング間隔を変更するには AWS CLI

`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。

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。

ターゲットの正常性の評価を使用する

- 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` に設定されている場合、`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、Network Load Balancer に対し、そのロードバランサーに関連付けられた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であれば、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。その後、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が使用されている場合、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のいずれかに異常がある場合、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) 。ターゲットの正常性の評価を使用している場合は、フェイルオーバールーティングポリシーが失敗します。
- 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)。ターゲットの正常性の評価を使用している場合は、フェイルオーバールーティングポリシーが失敗します。

詳細については、Amazon Route 53 開発者ガイドの「DNS フェイルオーバーの設定」を参照してください。

ターゲットグループへのターゲットの登録

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、「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。

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: 経由で送信される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。
- PrivateLink 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 では、IPv4 アドレスを IPv6 に変換するときにクライアント IP アドレスが保持されます。ただし、IPv6 アドレスを IPv4 に変換する場合、ソース IP は常に Network Load Balancer のプライベート IP アドレスです。

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 アドレスを検索するには

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、[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

ソース	プロトコル	ポート範囲	[Comment] (コメント)
`###### IP ####`	`####`	`####`	クライアントトラフィックを許可する

			(instance ターゲットタイプ)
VPC CIDR	*####*	*####*	クライアントトラフィックを許可する (ip ターゲットタイプ)
VPC CIDR	*#######*	*#######*	ロードバランサーからのヘルスチェックトラフィックを許可する

Outbound

送信先	プロトコル	ポート範囲	[Comment] (コメント)
###### IP ####	*####*	*####*	クライアントへのレスポンスを許可する (instance ターゲットタイプ)
VPC CIDR	*####*	*####*	クライアントへのレスポンスを許可する (ip ターゲットタイプ)
VPC CIDR	*#######*	1024-65535	ヘルスチェックトラフィックを許可する

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

ソース	プロトコル	ポート範囲	[Comment] (コメント)
###### IP ####	*####*	*####*	クライアントトラフィックを許可する

			(instance ターゲットタイプ)
VPC CIDR	*####*	*####*	クライアントトラフィックを許可する (ip ターゲットタイプ)
VPC CIDR	*#######*	1024-65535	ヘルスチェックトラフィックを許可する
Outbound			
送信先	プロトコル	ポート範囲	[Comment] (コメント)
###### IP ####	*####*	*####*	クライアントへのレスポンスを許可する (instance ターゲットタイプ)
VPC CIDR	*####*	*####*	クライアントへのレスポンスを許可する (ip ターゲットタイプ)
VPC CIDR	*#######*	*#######*	ヘルスチェックトラフィックを許可する
VPC CIDR	*#######*	1024-65535	ヘルスチェックトラフィックを許可する

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

ターゲットの登録または登録解除

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の種類により、ターゲットグループにターゲットを登録する方法が決定されます。詳しくは、「[Target type (ターゲットタイプ)]」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。 Amazon EC2
- 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 Amazon 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。
- UDP および TCP_UDP 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。

目次

インスタンス ID によるターゲットの登録または登録解除

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。

コンソールを使用してターゲットをインスタンス ID で登録または登録解除するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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. インスタンスを登録するには、[ターゲットの登録] を選択します。1 つ以上のインスタンスを選択し、必要に応じてデフォルトのインスタンスポートを入力して、[保留中として以下を含める] を選択します。インスタンスの追加が完了したら、[保留中のターゲットの登録] を選択します。

   [Note:] (メモ:)

   - IPv6 ターゲットグループに登録する場合、インスタンスにプライマリ IPv6 アドレスが割り当てられている必要があります。
   - AWS GovCloud (US) Regionはコンソールでのプライマリ IPv6 アドレスの割り当てをサポートしていません。でプライマリ IPv6 アドレスを割り当てるには、 API を使用する必要があります AWS GovCloud (US) Region。
6. インスタンスを登録解除するには、インスタンスを選択して [登録解除] を選択します。

IP アドレスによるターゲットの登録または登録解除

IPv4 ターゲット

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)

ターゲットグループの作成後に IP アドレスのタイプを変更することはできません。

参加者として共有 Amazon VPC で 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。

IPv6 ターゲット

- 登録する IP アドレスは、VPC CIDR ブロック内、またはピア接続された VPC CIDR ブロック内にある必要があります。
- ターゲットグループの作成後に IP アドレスのタイプを変更することはできません。
- IPv6 ターゲットグループは、TCP または TLS リスナーを使用するデュアルスタックロードバランサーにのみ関連付けることができます。

コンソールを使用してターゲットを IP アドレスで登録または登録解除するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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. IP アドレスを登録するには、[ターゲットの登録] を選択します。IP アドレスごとにネットワーク、アベイラビリティーゾーン、IP アドレス (IPv4 または IPv6)、ポートを選択し、[Include as pending below] (保留中として以下を含める) を選択します。アドレスの指定が終了したら、[保留中のターゲットの登録] を選択します。

6. IP アドレスの登録を解除するには、IP アドレスを選択して [登録解除] を選択します。登録済みの IP アドレスが多い場合は、フィルタを追加したりソート順を変更したりすると便利です。

AWS CLIを使用してターゲットを登録または登録解除する

ターゲットを追加するには register-targets コマンドを使用し、ターゲットを削除するには deregister-targets コマンドを使用します。

ターゲットとしての Application Load Balancer

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

- Application Load Balancer を Network Load Balancer のターゲットとして関連付けるには、同じアカウント内の同じ Amazon VPC に存在する必要があります。
- 1 つの Application Load Balancer は、複数の 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、それぞれの 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あたりのターゲットの最大数が 50 (クロスゾーンロードバランシングが無効の場合) または 100 (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) 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。

ステップ 1: Application Load Balancer を作成する

開始する前に、この Application Load Balancer が使用するターゲットグループを構成します。ターゲットグループに登録するターゲットがある 仮想プライベートクラウド (VPC) があることを確認します。この VPC には、ターゲットが使用する各アベイラビリティーゾーンで少なくとも 1 つのパブリックサブネットが必要です。

コンソールを使用して Application Load Balancer を作成するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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. [Application Load Balancer] で [作成] を選択します。
5. [Create Application Load Balancer] (Application Load Balancer の作成) ページの [Basic configuration] (基本設定) で、[Load balancer name] (ロードバランサー名)、[Scheme] (スキーム)、[IP address type] (IP アドレスタイプ) を指定します。
6. [Listener] (リスナー) セクションでは、HTTP または HTTPS リスナーを任意のポートに作成できます。ただし、このリスナーのポート番号は、この Application Load Balancer が存在するターゲットグループのポートと一致する必要があります。
7. [Availability Zones] (アベイラビリティーゾーン) で次の操作を行います。

   a. [VPC] で、Application Load Balancer のターゲットとして含めたインスタンスまたは IP アドレスを含む Virtual Private Cloud (VPC) を選択します。ステップ 3: Network Load Balancer を作成し、Application Load Balancer をターゲットとして設定する の Network Load Balancer に使用するのと同じ VPC を使用する必要があります。

   b. 2 つ以上のアベイラビリティーゾーンおよび対応するサブネットを選択します。可用性、スケーリング、パフォーマンスを最適化するために、これらのアベイラビリティーゾーンが

Network Load Balancer に対して有効になっているアベイラビリティーゾーンと一致していることを確認します。

8. 新しいセキュリティグループを作成するか、既存のセキュリティグループを選択することによって、ロードバランサーにセキュリティグループを割り当てることができます。

   選択したセキュリティグループは、このロードバランサーのリスナーポートへのトラフィックを許可するルールを含む必要があります。クライアントのコンピューターの CIDR ブロック (IP アドレス範囲) をセキュリティグループのインバウンドルールのトラフィックソースとして使用できます。これにより、クライアントは、この Application Load Balancer を介してトラフィックを送信できます。Network Load Balancer のターゲットとしての Application Load Balancer のセキュリティグループの設定の詳細については、「Application Load Balancer ユーザーガイド」の「Application Load Balancer のセキュリティグループ」を参照してください。

9. [Configure Routing] (ルーティングの設定) で、このApplication Load Balancer に対して設定したターゲットグループを選択します。使用可能なターゲットグループがない場合に新しいターゲットグループを設定するには、Application Load Balancer ユーザーガイドの「ターゲットグループの作成」を参照してください。

10. 設定を確認し、[ロードバランサーの作成] を選択します。

を使用して Application Load Balancer を作成するには AWS CLI

create-load-balancer コマンドを使用します。

ステップ 2: 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する

ターゲットグループを作成すると、新規または既存の Application Load Balancer をターゲットとして登録できます。ターゲットグループごとに追加できる Application Load Balancer は 1 つだけです。最大 2 つの Network Load Balancer のターゲットとして、同じ Application Load Balancer を別のターゲットグループで使用することもできます。

コンソールを使用してターゲットグループを作成し、Application Load Balancer をターゲットとして登録するには

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. リポジトリの [Specify group details] (グループ詳細の指定) ページの [Basic configuration] (基本的設定) で、[Application Load Balancer] を選択します。
5. [Target group name] (ターゲットグループ名) に Application Load Balancer 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。または、このポートと一致するように Application Load Balancer のリスナーポートを追加または編集できます。
7. [VPC] には、ターゲットグループに含める Application Load Balancer を含む [virtual private cloud (VPC)] (仮想プライベートクラウド (VPC)) を選択します。
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) 必要に応じて 1 つまたは複数のタグを追加します。
10. [次へ] をクリックします。
11. [Register targets] (ターゲットの登録) ページで、ターゲットとして登録する Application Load Balancer を選択します。リストから選択する Application Load Balancer には、作成する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、前の手順に戻ってターゲットグループに指定したポートを変更することができます。ターゲットとして追加する Application Load Balancer がわからない場合や、この時点で追加しない場合は、後で Application Load Balancer を追加できます。
12. [ターゲットグループの作成] を選択します。

AWS CLIを使用してターゲットグループを作成し、Application Load Balancer をターゲットとして登録するには

create-target-group コマンドと register-targets コマンドを使用します。

ステップ 3: Network Load Balancer を作成し、Application Load Balancer をターゲットとして設定する

コンソールを使用して Network Load Balancer を作成し、コンソールを使用して Application Load Balancer をターゲットとして設定するには、以下のステップに従います。

コンソールを使用して Network Load Balancer とリスナーを作成するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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

   [基本的な設定] で、[ロードバランサー名]、[スキーム]、および [IP アドレスタイプ] を設定します。
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。[スキーム] で [インターネット向け] を選択した場合は、インターネットゲートウェイを持つ VPC だけを選択できます。

   b. [マッピング] で、1 つまたは複数のアベイラビリティーゾーンと対応するサブネットを選択します。可用性、スケーリング、パフォーマンスを最適化するために、Application Load Balancer ターゲット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。

   (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. 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、トラフィックを転送する Application Load Balancer ターゲットグループを選択します。ターゲットグループがリストに表示されない場合、または (別の Network Load Balancer によってすでに使用されていて) 選択できない場合は、「ステップ 2: 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する」の手順に従って Application Load Balancer ターゲットグループを作成できます。

8. タグ

   (オプション) タグを追加して、ロードバランサーを分類します。詳細については、「タグ」を参照してください。

9. [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。

を使用して Network Load Balancer を作成するには AWS CLI

create-load-balancer コマンドを使用します。

ステップ 4: (オプション) VPC エンドポイントの作成

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、[PC 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。残りの手順については、AWS PrivateLink ガイドの「エンドポイントサービスを作成する」を参照してください。

ターゲットグループのタグ

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 用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。

コンソールを使用してターゲットグループのタグを更新するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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. タグの更新を完了したら、[変更内容の保存] を選択します。

を使用してターゲットグループのタグを更新するには AWS CLI

add-tags コマンドと remove-tags コマンドを使用します。

ターゲットグループの削除

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。

コンソールを使用してターゲットグループを削除するには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインの [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. 確認を求めるメッセージが表示されたら、[はい、削除します] を選択します。

を使用してターゲットグループを削除するには AWS CLI

delete-target-group コマンドを使用します。

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 を使用して CloudWatch 、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を、メトリクス と呼ばれる時系列データの順序付けられた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、「CloudWatch Network Load Balancer の 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。

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

を使用して AWS CloudTrail 、Elastic Load Balancing API に対して行われた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、どの呼び出しが行われたか、呼び出し元の送信元 IP アドレス、呼び出し者、呼び出し日時などを確認できます。詳細については、「AWS CloudTrailを使用した Network Load Balancer での API 呼び出しのログ記録」を参照してください。

CloudWatch Network Load Balancer の メトリクス

Elastic Load Balancing は、ロードバランサーとターゲット CloudWatch のデータポイントを Amazon に発行します。 CloudWatch を使用すると、これらのデータポイントに関する統計を、メトリクス と呼ばれる時系列データの順序付けられた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、指定したメトリクスをモニタリングする CloudWatch アラームを作成し、メトリクスが許容範囲外になった場合にアクション (E メールアドレスへの通知の送信など) を開始できます。

Elastic Load Balancing は、リクエストがロードバランサーを流れる CloudWatch 場合にのみ、メトリクスを 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を持つ 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メトリクスに CloudWatch キャプチャされません。

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

内容

- Network Load Balancer メトリクス

- Network Load Balancer のメトリクスディメンション
- Network Load Balancer メトリクスの統計
- ロードバランサーの CloudWatch メトリクスを表示する

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
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。

メトリクス	説明
	ディメンション • LoadBalancer • AvailabilityZone , LoadBalancer
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
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

メトリクス	説明
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
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

メトリクス	説明
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
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

メトリクス	説明
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: 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 , 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
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

メトリクス	説明
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
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
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: ゼロ以外の値がある。 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
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: 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	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

メトリクス	説明
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
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

メトリクス	説明
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: 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_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

メトリクス	説明
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
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

メトリクス	説明
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
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

メトリクス	説明
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
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

メトリクス	説明
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: 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 , 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。 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

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 は、Elastic Load Balancing によって発行されたメトリクスデータポイントに基づく統計を提供します。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。例えば、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 メトリクスを表示する

Amazon EC2 コンソールを使用して、ロードバランサーの 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。

または、コンソールを使用してロードバランサーの CloudWatchメトリクスを表示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 (https://console.aws.amazon.com/ec2/) を開きます。
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. https://console.aws.amazon.com/cloudwatch/ で CloudWatch コンソールを開きます。
2. ナビゲーションペインで メトリクスを選択します。
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。では、ディメンションの各一意の組み合わせを個別のメトリクスとして CloudWatch 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 で確立された TLS 接続に関する詳細情報をキャプチャ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**
>
> アクセスログは、Network Load Balancer に TLS リスナーがあり、TLS 接続のみに関する情報が含まれている場合にのみ作成されます。

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、それ自体は定期的にローテーションされる KMS キーで暗号化されます。詳細については、Amazon S3 ユーザーガイド」の「Amazon S3 暗号化 (SSE-S3) の指定」および AWS KMS 「 (SSE-KMS) による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。

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)。
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ています。
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)。

フィールド	説明
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。
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_named_group	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。
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。

フィールド	説明
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
```

バケットの要件

アクセスログの作成を有効にするときは、アクセスログの S3 バケットを指定する必要があります。バケットは、ロードバランサーを所有するアカウントとは別のアカウントが所有するものでもかまいません。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。
- 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。ポリシーの例を次に示します。

```
{
    "Version": "2012-10-17",
    "Id": "AWSLogDeliveryWrite",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryAclCheck",
            "Effect": "Allow",
            "Principal": {
                "Service": "delivery.logs.amazonaws.com"
            },
            "Action": "s3:GetBucketAcl",
            "Resource": "arn:aws:s3:::my-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:::my-bucket/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:*"]
                }
            }
        }
    ]
}
```

前のポリシーでは、aws:SourceAccount にはこのバケットにログが配信されるアカウント番号のリストを指定します。aws:SourceArn には、ログを生成するリソースの ARN のリストを arn:aws:logs:*source-region*:*source-account-id*:* の形式で指定します。

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 ユーザーガイド」の「Amazon S3 暗号化 (SSE-S3) の指定」および AWS KMS 「 (SSE-KMS) による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。ポリシーの例を次に示します。

```
{
  "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": "*"
    }
  ]
}
```

アクセスログの作成の有効化

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 バケットを指定する必要があります。このバケットを所有していること、およびこのバケットに必要なバケットポリシーを設定したことを確認します。詳細については、「バケットの要件」を参照してください。

コンソールを使用してアクセスログの作成を有効にするには

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. [Edit load balancer attributes] ページで、以下を実行します。
   a. [モニタリング] で [アクセスログ] をオンにします。
   b. [S3 をブラウズ] を選択し、使用するバケットを選択します。または、プレフィックスを含めて S3 バケットの場所を入力します。
   c. [変更の保存] をクリックします。

を使用してアクセスログ記録を有効にするには AWS CLI

modify-load-balancer-attributes コマンドを使用します。

アクセスログの作成の無効化

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 Simple Storage Service ユーザーガイドでバケットの使用について参照してください。

コンソールを使用してアクセスログの作成を無効にするには

1. Amazon EC2 コンソール (https://console.aws.amazon.com/ec2/) を開きます。
2. ナビゲーションペインで、[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. [変更の保存] をクリックします。

を使用してアクセスログ記録を無効にするには AWS CLI

modify-load-balancer-attributes コマンドを使用します。

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。処理を使用して、このような大量のデータを line-by-line 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

AWS CloudTrailを使用した Network Load Balancer での API 呼び出しのログ記録

Elastic Load Balancing は AWS CloudTrail、Elastic Load Balancing のユーザー、ロール、または によって実行されたアクションを記録するサービスである と統合 AWS のサービス されています。 は、Elastic Load Balancing のすべての API コールをイベントとして CloudTrail キャプチャします。 Elastic Load Balancing キャプチャされた呼び出しには、 からの呼び出し AWS Management Console と、Elastic Load Balancing API オペレーションへのコード呼び出しが含まれます。証跡を作成する場合は、Elastic Load Balancing の CloudTrail イベントなど、Amazon S3 バケットへのイベントの継続的な配信を有効にすることができます。 Amazon S3 証跡を設定しない場合でも、CloudTrail コンソールのイベント履歴 で最新のイベントを表示できます。によって収集された情報を使用して CloudTrail、Elastic Load Balancing に対するリクエスト、リクエスト元の IP アドレス、リクエスト者、リクエスト日時などの詳細を確認できます。

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

の Elastic Load Balancing 情報 CloudTrail

CloudTrail アカウントを作成する AWS アカウント と、 で が有効になります。Elastic Load Balancing でアクティビティが発生すると、そのアクティビティは CloudTrail イベント履歴 の他の AWS のサービス イベントとともにイベントに記録されます。 で最近のイベントを表示、検索、ダウンロードできます AWS アカウント。詳細については、「イベント履歴 を含む CloudTrail イベントの表示」を参照してください。

Elastic Load Balancing のイベントなど AWS アカウント、 のイベントの継続的な記録については、証跡を作成します。証跡により CloudTrail 、 はログファイルを Amazon S3 バケットに配信できます。デフォルトでは、コンソールで証跡を作成すると、証跡はすべての AWS リージョンに適用されます。証跡は、 AWS パーティションのすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、他の を設定 AWS のサービス して、 CloudTrail ログで収集されたイベントデータをさらに分析し、それに基づく対応を行うことができます。詳細については、次を参照してください:

- 「証跡作成の概要」
- CloudTrail がサポートするサービスと統合
- の Amazon SNS 通知の設定 CloudTrail
- 複数のリージョンからの CloudTrail ログファイルの受信と複数のアカウントからの CloudTrail ログファイルの受信

Network Load Balancer のすべての Elastic Load Balancing アクションは によって CloudTrail ログに記録され、Elastic Load Balancing API リファレンスバージョン 2015-12-01 に記載されています。例えば、 および DeleteLoadBalancerアクションを呼び出すCreateLoadBalancerと、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。ID 情報は次の判断に役立ちます。

- リクエストが、ルートとユーザー認証情報のどちらを使用して送信されたか。
- リクエストが、ロールとフェデレーションユーザーの一時的なセキュリティ認証情報のどちらを使用して送信されたか。
- リクエストが、別の AWS のサービスによって送信されたかどうか。

詳細については、CloudTrailuserIdentity 要素」を参照してください。

Elastic Load Balancing ログファイルのエントリの理解

証跡は、指定した Amazon S3 バケットにイベントをログファイルとして配信できるようにする設定です。 CloudTrail ログファイルには 1 つ以上のログエントリが含まれます。イベントは任意ソースからの単一リクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどの情報を含みます。 CloudTrail ログファイルはパブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

ログファイルには、Elastic Load Balancing AWS API コールだけでなく AWS アカウント、 のすべての API コールのイベントが含まれます。値 eventSource を使用して elasticloadbalancing.amazonaws.com 要素を確認することで、Elastic Load Balancing API に対する呼び出しを見つけることができます。CreateLoadBalancer などの特定のアクションのレコードを表示するには、アクション名で eventName 要素を確認します。

Network Load Balancer を作成し、 を使用して削除したユーザーの Elastic Load Balancing の CloudTrail ログレコードの例を次に示します AWS CLI。 Load Balancer userAgent 要素を使用して CLI を特定できます。eventName 要素を使用して、リクエストされた API コールを特定できます。ユーザーに関する情報 (Alice) は userIdentity 要素で確認できます。

Example 例： CreateLoadBalancer

```
{
    "eventVersion": "1.03",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "CreateLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "subnets": ["subnet-8360a9e7","subnet-b7d581c0"],
        "securityGroups": ["sg-5943793c"],
        "name": "my-load-balancer",
        "scheme": "internet-facing",
        "type": "network"
    },
    "responseElements": {
        "loadBalancers":[{
            "type": "network",
            "ipAddressType": "ipv4",
            "loadBalancerName": "my-load-balancer",
            "vpcId": "vpc-3ac0fb5f",
            "securityGroups": ["sg-5943793c"],
            "state": {"code":"provisioning"},
            "availabilityZones": [
               {"subnetId":"subnet-8360a9e7","zoneName":"us-west-2a"},
               {"subnetId":"subnet-b7d581c0","zoneName":"us-west-2b"}
            ],
            "dNSName": "my-load-balancer-1836718677.us-west-2.elb.amazonaws.com",
            "canonicalHostedZoneId": "Z2P70J7HTTTPLU",
            "createdTime": "Apr 11, 2016 5:23:50 PM",
            "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/net/my-load-balancer/ffcddace1759e1d0",
            "scheme": "internet-facing"
        }]
    },
    "requestID": "b9960276-b9b2-11e3-8a13-f1ef1EXAMPLE",
    "eventID": "6f4ab5bd-2daa-4d00-be14-d92efEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-12-01",
```

```
    "recipientAccountId": "123456789012"
}
```

Example 例： DeleteLoadBalancer

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "DeleteLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/net/my-load-balancer/ffcddace1759e1d0"
    },
    "responseElements": null,
    "requestID": "349598b3-000e-11e6-a82b-298133eEXAMPLE",
    "eventID": "75e81c95-4012-421f-a0cf-babdaEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-12-01",
    "recipientAccountId": "123456789012"
}
```

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、「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。

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。

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。

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。

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
- 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。

を介して接続するポート割り当てエラー AWS PrivateLink

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、ロードバランサー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は、PortAllocationErrorCount メトリクスを使用して追跡できます。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。詳細については、「CloudWatch Network Load Balancer の メトリクス」を参照してください。

クライアント IP 保存が有効になっている場合の断続的な接続障害

クライアント IP の保存が有効な場合、ターゲットで確認されたソケットの再利用に関連する TCP/IP 接続の制限が発生することがあります。これらの接続制限が発生する可能性があるのは、クライアント、またはクライアントの前面にある NAT デバイスが、複数のロードバランサーノードに同時に接続する際に、同じ送信元 IP アドレスと送信元ポートを使用する場合です。ロードバランサーがこれらの接続を同じターゲットにルーティングする場合、接続は同じ送信元ソケットからの接続のようにターゲットに表示され、それにより接続エラーが発生します。その場合、クライアントは再試行 (接続が失敗した場合)、または再接続 (接続が中断した場合) できます。このタイプの接続エラーは、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すことによって減らすことができます。このタイプの接続エラーは、クライアント IP の保存を無効にするか、クロスゾーン負荷分散を無効にすることで防止できます。

さらに、クライアント IP の保存が有効な場合、Network Load Balancer に接続しているクライアントがロードバランサーの背後にあるターゲットにも接続されると接続に失敗することがあります。この問題を解決するには、影響を受けるターゲットグループでクライアント IP 保存を無効にすることができます。または、クライアントを Network Load Balancer にのみ接続するか、ターゲットにのみ接続します。ただし、両方に接続することはできません。

TCP 接続の遅延

クロスゾーン負荷分散とクライアント IP の保存の両方が有効になっている場合、同じロードバランサー上の異なる IP に接続しているクライアントが同じターゲットにルーティングされることがあります。クライアントがこれらの接続の両方に同じソースポートを使用する場合、ターゲットでは重複しているかのような接続を受信します。これにより、接続エラーや新しい接続を確立する際の TCP 遅延が発生する可能性があります。このタイプの接続エラーは、クロスゾーン負荷分散を無効にすることで防止できます。詳細については、「クロスゾーン負荷分散」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。

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。

リソースマップを使用した異常なターゲットのトラブルシューティング

Network Load Balancer ターゲットがヘルスチェックに失敗している場合は、リソースマップを使用して異常なターゲットを検索し、失敗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 の 2 つのビューがあります。 概要はデフォルトで選択され、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ターゲットと異常ターゲットの概要が表示されます。これにより、すべてのターゲットがヘルスチェックに失敗しているか、特定のターゲットのみが失敗しているかを判断できます。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失敗している場合は、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、その詳細ページを新しいタブで開きます。

Targets 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と、ヘルスチェックの失敗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失敗する場合は、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、その詳細ページを新しいタブで開きます。

Export 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失敗していることを確認してから、次の問題の失敗理由コードチェックに基づきます。

- 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 に関連付けられている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。
  - ターゲットに、Network Load Balancer からの接続を受け入れるのに十分な容量が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表示できます。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でトラフィックをリッスンしていることを確認します。

TLS リスナーを使用する場合

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されるセキュリティポリシーは、使用中のフロントエンド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。

- TLS リスナーがフロントエンド接続に TLS 1.3 セキュリティポリシーを使用している場合は、バックエンド接続にELBSecurityPolicy-TLS13-1-0-2021-06セキュリティポリシーが使用されます。
- TLS リスナーがフロントエンド接続に TLS 1.3 セキュリティポリシーを使用していない場合、ELBSecurityPolicy-2016-08セキュリティポリシーはバックエンド接続に使用されます。

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。

- 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 によって提供され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 アカウントには、AWS のサービスごとにデフォルトのクォータ (以前は制限と呼ばれていました) があります。特に明記されていない限り、クォータはリージョンごとに存在します。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。クォータが Service Quotas でまだ使用できない場合は、Elastic Load Balancing の制限引き上げフォームを使用します。

ロードバランサー

お客様の AWS アカウントには、Network Load Balancer に関連する以下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	はい
Network Load Balancer あたりのリスナー	50	No
VPC あたりの Network Load Balancer ENI	1,200 [1]	はい
リージョンあたりの Network Load Balancer	50	はい
Network Load Balancer あたりのターゲットグループ (アクションごと)	1	No
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]	はい
Network Load Balancer あたりのターゲット	3,000 [3]	はい

[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]	はい
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	はい
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	No

[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。

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
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 終了	このリリースでは、TLS 接続を終了するときに FIPS 140-3 暗号化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ています。	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、ヘルスチェックに失敗し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、デフォルトで登録解除タイムアウトの終了時に UDP 接続を終了す	2023 年 10 月 12 日

	るサポートが追加されています。	
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 で対処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、ターゲットグループレベルでクロスゾーン負荷分散を設定する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を設定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を設定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、事前定義された 3 つのフォワードシークレットセキュリティポリシー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	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

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。