ユーザーガイド

Elastic Load Balancing

Elastic Load Balancing: ユーザーガイド

Table of Contents

Elastic Load Balancing 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、着信トラフィックの変化に応じて、自動的にロードバランサーの容量を拡張します。

ロードバランサーの利点

ロードバランサーは、ワークロードを仮想サーバーなど複数のコンピューティングリソース間に分散させます。ロードバランサーを使用すると、アプリケーションの可用性と耐障害性が向上します。

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コンピューティングリソースの追加と削除を行うことができます。

ロードバランサーが正常なものにのみリクエストを送信するように、コンピューティングリソースのヘルス状態をモニタリングするヘルスチェックを設定できます。コンピューティングリソースがメインワークに集中できるように、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もできます。

Elastic Load Balancing の特徴

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。詳細については、製品の比較の を参照してください。

各ロードバランサーの使用方法については、以下のドキュメントを参照してください。

- Application Load Balancer のユーザーガイド
- Network Load Balancer のユーザーガイド
- Gateway Load Balancer のユーザーガイド
- Classic Load Balancer のユーザーガイド

Elastic Load Balancing へのアクセス

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

- AWS Management Console— Elastic Load Balancing にアクセスするために使用できるウェブインターフェイスを提供します。
- AWS コマンドラインインターフェイス (AWS CLI) — Elastic Load Balancing を含む幅広い AWS サービスのコマンドを提供します。 AWS CLI は、Windows、macOSでサポートされています。詳細については、「AWS Command Line Interface」を参照してください。
- AWS SDKs — 言語固有の APIs を提供し、署名の計算、リクエストの再試行処理、エラー処理など、接続に関する多くの詳細を処理します。詳細については、AWS SDK をご参照ください。
- クエリ API— HTTPS リクエストを使用して呼び出す低レベル API アクションを提供します。クエリ API の使用は、Elastic Load Balancing にアクセスする最も直接的な方法です。ただし、クエリ API では、リクエストに署名するハッシュの生成やエラー処理など、低レベルの詳細な作業をアプリケーションで処理する必要があります。詳細については、以下を参照してください。
  - Application Load Balancer および Network Load Balancer — API バージョン 2015-12-01
  - Classic Load Balancer — API バージョン 2012-06-01

関連する のサービス

Elastic Load Balancing は、アプリケーションの可用性とスケーラビリティを高める以下のサービスを使用します。

- Amazon EC2 — クラウドでアプリケーションを実行する仮想サーバーです。EC2 インスタンスへのトラフィックをルーティングするように、ロードバランサーを設定できます。詳細については、Amazon EC2 ユーザーガイド」を参照してください。
- Amazon EC2 Auto Scaling — インスタンスに障害が発生した場合でも、必要な数のインスタンスを実行していることを確認します。Amazon EC2 Auto Scaling を使用すると、インスタンスに対する需要の変化に応じて、インスタンスの数を自動的に増減することもできます。Elastic Load Balancing で Auto Scaling を有効にすると、Auto Scaling によって起動されるインスタンスは自動的にロードバランサーに登録されます。同様に、Auto Scaling によって終了されたインスタンスは、ロードバランサー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」を参照してください。

- AWS Certificate Manager — HTTPS リスナーを作成するには、ACM で提供された証明書を指定できます。ロードバランサーは、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。
- Amazon CloudWatch — ロードバランサーをモニタリングし、必要に応じてアクションを実行できます。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。
- Amazon ECS — EC2 インスタンスのクラスター上で Docker コンテナを実行、停止、管理することができます。コンテナにトラフィックをルーティングするように、ロードバランサーを設定できます。詳細については、Amazon Elastic Container Service デベロッパーガイドを参照してください。
- AWS Global Accelerator — アプリケーションの可用性とパフォーマンスが向上します。アクセラレーターを使用して、1 つ以上の AWS リージョンの複数のロードバランサーにトラフィックを分散します。詳細については、「AWS Global Accelerator デベロッパーガイド」を参照してください。
- Route 53 — ドメイン名を、コンピュータが相互の接続に使用する数字の IP アドレスに変換することで、閲覧者をウェブサイトにルーティングするための信頼性が高く、コスト効率のよい方法を提供します。例えば、 は数値 IP アドレス `www.example.com` に変換されます`192.0.2.1`。は、ロードバランサーなどのリソースに URLsを AWS 割り当てます。ただし、ユーザーが覚えやすい URL を使用することもできます。たとえば、ドメイン名をお客様のロードバランサーにマッピングすることができます。詳細については、Amazon Route 53 デベロッパーガイドを参照してください。
- AWS WAF — Application Load Balancer AWS WAF で を使用して、ウェブアクセスコントロールリスト (ウェブ ACL) のルールに基づいてリクエストを許可またはブロックできます。詳細については、AWS WAF デベロッパーガイドを参照してください。

料金

ロードバランサーについては、お客様が利用された分のみのお支払いとなります。詳細については、Elastic Load Balancing の料金表を参照してください。

Elastic Load Balancing の仕組み

ロードバランサーは、クライアントからの受信トラフィックを受け入れ、リクエストを 1 つ以上のアベイラビリティーゾーンにある登録済みのターゲット (EC2 インスタンスなど) にルーティングします。また、ロードバランサーは登録されているターゲットの状態を監視して、トラフィックが正常なターゲットにのみルーティングされるようにします。ロードバランサーは、異常なターゲットを検出すると、そのターゲットへのトラフィックのルーティングを中止します。その後、ターゲットが再び正常になったことを検出すると、そのターゲットへのトラフィックのルーティングを再開します。

1 つ以上のリスナーを指定することで、受信トラフィックを受け入れるようにロードバランサーを設定します。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番号を使用して設定します。同様に、ロードバランサーからターゲットへの接続用のプロトコルとポート番号を使用して設定します。

Elastic Load Balancing では、以下のタイプのロードバランサーをサポートしています。

- Application Load Balancer
- Network Load Balancer
- ゲートウェイロードバランサー
- クラシックロードバランサー

ロードバランサーの設定方法は、種類によって大きく異なります。Application Load Balancer、Network Load Balancer、ゲートウェイロードバランサーでは、ターゲットをターゲットグループに登録し、トラフィックをターゲットグループにルーティングします。Classic Load Balancer では、ロードバランサーにインスタンスを登録します。

アベイラビリティーゾーンとロードバランサーノード

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。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トラフィックを受信しません。有効な各アベイラビリティーゾーンに 1 つ以上の登録済みターゲットが含まれるようにすると、ロードバランサーは最も効果的に機能します。

すべてのロードバランサーに対して複数のアベイラビリティーゾーンを有効にすることをお勧めします。ただし、Application Load Balancer では、少なくとも 2 つ以上のアベイラビリティーゾーンを有

効にする必要があります。この設定により、ロードバランサーが引き続きトラフィックをルーティングできるようになります。1 つのアベイラビリティーゾーンが利用できなくなるか、正常なターゲットがなくなった場合、ロードバランサーは別のアベイラビリティーゾーンの正常なターゲットにトラフィックをルーティングできます。

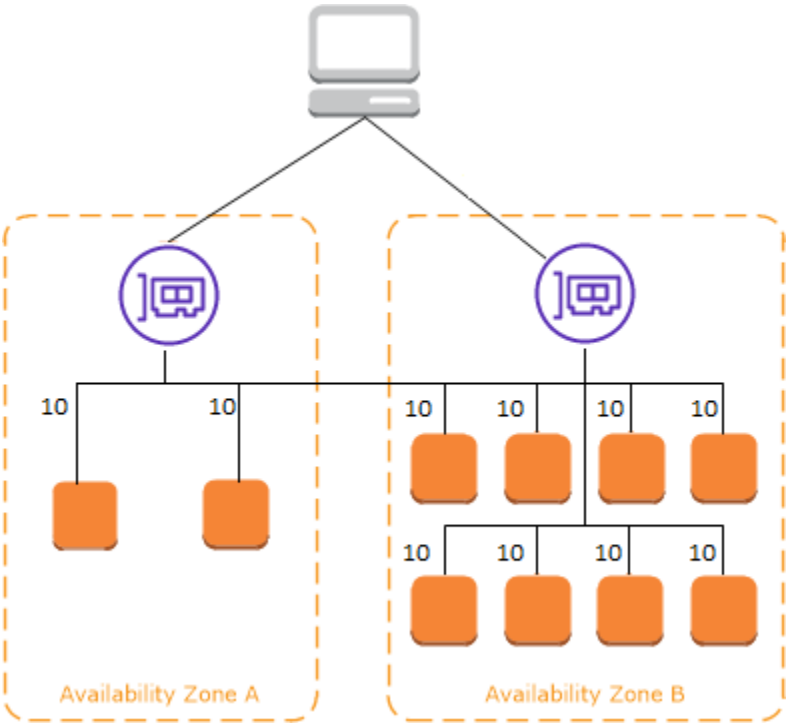
アベイラビリティーゾーンを無効にしても、そのアベイラビリティーゾーンのターゲットはロードバランサーに登録されたままになります。ただし、登録されたままであっても、ロードバランサーはトラフィックをルーティングしません。

クロスゾーン負荷分散

ロードバランサーのノードは、クライアントからのリクエストを登録済みターゲットに分散させます。クロスゾーン負荷分散が有効な場合、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。クロスゾーン負荷分散が無効な場合、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にのみトラフィックを分散します。

次の図はラウンドロビンをデフォルトのルーティングアルゴリズムとして使用したクロスゾーンロードバランシングの効果について示しています。有効なアベイラビリティーゾーンが 2 つがあり、アベイラビリティーゾーン A には 2 つのターゲット、アベイラビリティーゾーン B には 8 つのターゲットがあります。クライアントがリクエストを送信すると、Amazon Route 53 はロードバランサーノードのいずれか 1 つの IP アドレスを使用して各リクエストに応答します。ラウンドロビンルーティングアルゴリズムに基づいて、各ロードバランサーノードがクライアントからのトラフィックの 50% を受け取るようにトラフィックが分散されます。各ロードバランサーノードは、範囲内の登録済みターゲット間で配分されたトラフィックを分散します。

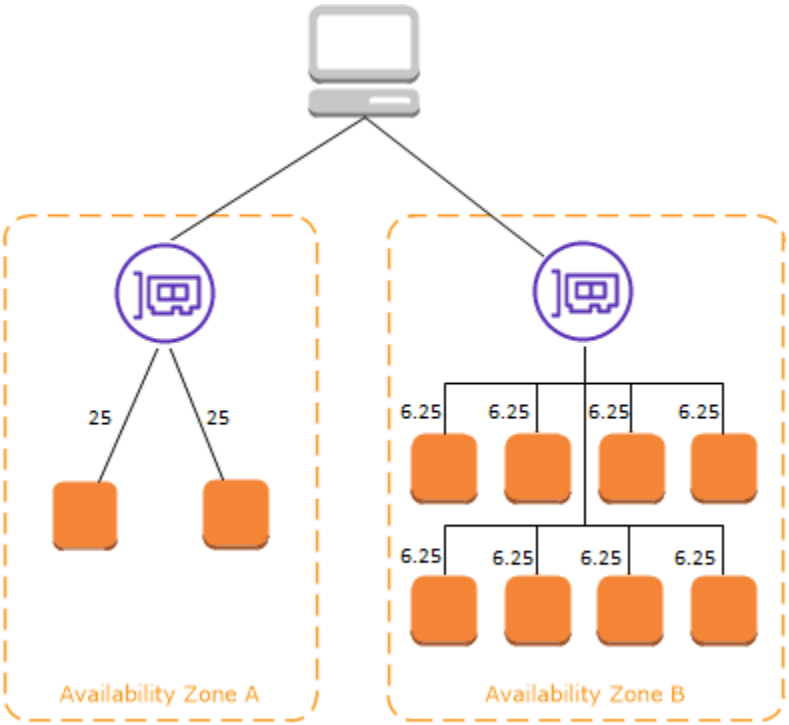
クロスゾーン負荷分散が有効な場合、10 個のターゲットのそれぞれがトラフィックの 10% を受け取ります。これは、各ロードバランサーノードが、そのクライアントトラフィックの 50% を 10 個のターゲットすべてにルーティングできるためです。



クロスゾーン負荷分散が無効な場合:

- アベイラビリティーゾーン A の 2 つのターゲットはそれぞれ、トラフィックの 25% を受け取ります。
- アベイラビリティーゾーン B の 8 つのターゲットはそれぞれ、トラフィックの 6.25% を受け取ります。

これは、各ロードバランサーノードが、そのクライアントトラフィックの 50% を自身のアベイラビリティーゾーンのターゲットにのみルーティングできるためです。



Application Load Balancer では、クロスゾーン負荷分散がロードバランサーレベルで常に有効になっています。ターゲットグループレベルでは、クロスゾーン負荷分散を無効化できます。詳細については、「Application Load Balancer ユーザーガイド」の「Turn off cross-zone load balancing」(クロスゾーン負荷分散をオフにする) を参照してください。

Network Load Balancer と Gateway Load Balancer では、クロスゾーン負荷分散はデフォルトで無効になっています。ロードバランサーの作成後は、いつでもクロスゾーン負荷分散を有効または無効にできます。

Classic Load Balancer の作成時における、クロスゾーン負荷分散のデフォルト状態は、ロードバランサーの作成方法により異なります。API または CLI を使用する場合、クロスゾーン負荷分散はデフォルトで無効化されます。では AWS Management Console、クロスゾーン負荷分散を有効にするオプションがデフォルトで選択されています。クロスゾーン負荷分散は、Classic Load Balancer の作成後に、いつでも有効化または無効化できます。詳細については、Classic Load Balancer ユーザーガイドの Enable cross-zone load balancing を参照してください。

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 サービスに影響する場合、DNS からゾーンロードバランサーの IP アドレスをプロアクティブに削除します。ゾーンシフトを開始する前に、現在のアベイラビリティーゾーンの容量を必ず確認してください。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。

リクエストルーティング

クライアントがリクエストをロードバランサーに送信する前に、ドメインネームシステム (DNS) サーバーを使用してロードバランサーのドメイン名を解決します。ロードバランサーは `amazonaws.com` ドメインにあるため、DNS エントリは Amazon によって制御されます。Amazon DNS サーバーがクライアントに 1 つ以上の IP アドレスを返します。これらは、ロードバランサーのロードバランサーノードの IP アドレスです。Network Load Balancer を使用すると、Elastic Load Balancing は、有効にする各アベイラビリティーゾーンのネットワークインターフェイスを作成し、それを使用して静的 IP アドレスを取得します。Network Load Balancer の作成時に、必要に応じて 1 つの Elastic IP アドレスを各ネットワークインターフェイスに関連付けることができます。

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と、Elastic Load Balancing はロードバランサーをスケーリングして DNS エントリを更新します。DNS エントリは、60 秒の time-to-live (TTL) も指定します。これにより、トラフィックの変化に応じて IP アドレスを迅速に再マッピングできます。

クライアントは、ロードバランサーにリクエストを送信するために使用する IP アドレスを決定します。リクエストを受信したロードバランサーノードは、正常な登録済みターゲットを選択し、そのプライベート IP アドレスを使用してターゲットにリクエストを送信します。

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。

ルーティングアルゴリズム

Application Load Balancer では、リクエストを受信するロードバランサーノードは、次のプロセスを使用します。

1. リスナールールを優先度順に評価して、適用するルールを決定します。
2. ターゲットグループに設定されたルーティングアルゴリズムを使用して、ルールアクションのターゲットグループからターゲットを選択します。デフォルトのルーティングアルゴリズムはラウンドロビンです。それぞれのターゲットグループでルーティングは個別に実行され、複数のターゲットグループに登録されているターゲットの場合も同じです。

Network Load Balancer では、接続を受信するロードバランサーノードは、次のプロセスを使用します。

1. フローハッシュアルゴリズムを使用して、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。アルゴリズムは以下に基づきます。

- プロトコル
- 送信元 IP アドレスと送信元ポート
- 送信先 IP アドレスと送信先ポート
- TCP シーケンス番号

2. 接続中、各 TCP 接続を単一のターゲットにルーティングします。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。

Classic Load Balancer では、リクエストを受信するロードバランサーノードは、次のように登録済みインスタンスを選択します。

- TCP リスナーのラウンドロビンルーティングアルゴリズムを使用する
- HTTP リスナーと HTTPS リスナーの最小未処理リクエストルーティングアルゴリズムを使用する

HTTP 接続

Classic Load Balancer は事前に開かれた接続を使用しますが、Application Load Balancer は使用しません。Classic Load Balancer と Application Load Balancer はどちらも接続の多重化を使用します。つまり、複数のフロントエンド接続の複数のクライアントからのリクエストは、1 つのバックエンド接続を介して指定のターゲットにルーティングできます。接続の多重化により、レイテンシーが改善され、アプリケーションの負荷が低下します。接続の多重化を回避するには、HTTP レスポンスの `Connection: close` ヘッダーを設定して、HTTP `keep-alive` ヘッダーを無効にします。

Application Load Balancer と Classic Load Balancer は、フロントエンド接続でパイプライン化された HTTP をサポートします。バックエンド接続ではパイプライン化された HTTP をサポートしていません。

Application Load Balancer は、GET、HEAD、POST、PUT、DELETE、OPTIONS、PATCH の HTTP リクエストメソッドをサポートしています。

Application Load Balancer はフロントエンド接続の次のプロトコールをサポートします: HTTP/0.9、HTTP/1.0、HTTP/1.1、HTTP/2。HTTPS/2 は HTTPS リスナーにのみ使用でき、HTTP/2 接続を使用して最大 128 のリクエストを並行して送信できます。Application Load Balancer は、HTTP から への接続アップグレードもサポートしています WebSockets。ただし、接続のアップグレードがある場合、Application Load Balancer リスナールーティングルールと AWS WAF 統合は適用されません。

デフォルトでは、Application Load Balancer はバックエンド接続 (登録されたターゲットへのロードバランサー) で HTTP/1.1 を使用します。ただし、プロトコルバージョンを使用して、HTTP/2 または gRPC を使用するターゲットに要求を送信することができます。詳細については、「プロトコルバージョン」を参照してください。`keep-alive` ヘッダーは、デフォルトでは、バックエンド接続でサポートされています。ホストヘッダーを満たさないクライアントからの HTTP/1.0 リクエストの場合、ロードバランサーによりバックエンド接続で送信されたリクエストに対してHTTP/1.1ホストヘッダーを生成します。ホストヘッダーにはロードバランサーの DNS 名が含まれます。

Classic Load Balancer はフロントエンド接続 (ロードバランサーのクライアント) の次のプロトコールをサポートします: HTTP/0.9、HTTP/1.0、HTTP/1.1。デフォルトでは、 バックエンド接続 (登録されたターゲットへのロードバランサー) で HTTP/1.1 を使用します。`keep-alive` ヘッダーは、デフォルトでは、バックエンド接続でサポートされています。ホストヘッダーを満たさないクライアントからの HTTP/1.0 リクエストの場合、ロードバランサーによりバックエンド接続で送信されたリクエストに対してHTTP/1.1ホストヘッダーを生成します。ホストヘッダーにはロードバランサーノードの IP アドレスが含まれています。

HTTP ヘッダー

Application Load Balancer と Classic Load Balancer は、X-Forwarded-For、X-Forwarded-Proto、および X-Forwarded-Port ヘッダーを自動的にリクエストに追加します。

アプリケーションロードバランサは、HTTP ホストヘッダーのホスト名を小文字に変換してから、ターゲットに送信します。

HTTP/2 を使用するフロントエンド接続の場合は、ヘッダー名は小文字です。リクエストが HTTP/1.1 を使用してターゲットに送信される前に、以下のヘッダー名は、大小混合文字に変換されます: X-Forwarded-For、X-Forwarded-Proto、X-Forwarded-Port、Host、X-Amzn-Trace-Id、Upgrade、および Connection。そのほかのヘッダー名はすべて小文字です。

Application Load Balancer および Classic Load Balancer は、応答をクライアントにプロキシした後、着信クライアント要求からの接続ヘッダーを尊重します。

HTTP/1.1 を使用している Application Load Balancer と Classic Load Balancer は、Expect: 100-Continue ヘッダーを受け取ると、コンテンツ長ヘッダーをテストすることなく、直ちに HTTP/1.1 100 Continue で応答します。Expect: 100-Continue リクエストヘッダーはターゲットに転送されません。

HTTP/2 を使用する場合、Application Load Balancer はクライアントリクエストの Expect: 100-Continue ヘッダーをサポートしません。Application Load Balancer は HTTP/2 100 Continue で応答したり、このヘッダーをターゲットに転送したりしません。

HTTP ヘッダーの制限

Application Load Balancer の以下のサイズ制限は、変更できないハードリミットです。

- リクエストライン: 16 K
- 単一ヘッダー: 16 K
- レスポンスのヘッダー全体: 32 K
- リクエストのヘッダー全体: 64 K

ロードバランサーのスキーム

ロードバランサーを作成するとき、ロードバランサーを内部向けにするかインターネット向けにするか選択する必要があります。

インターネット向けロードバランサーのノードにはパブリック IP アドレスが必要です。インターネット向けロードバランサーの DNS 名は、ノードのパブリック IP アドレスにパブリックに解決可能です。したがって、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ルーティングできます。

内部ロードバランサーのノードはプライベート IP アドレスのみを持ちます。内部ロードバランサーの DNS 名は、ノードのプライベート IP アドレスにパブリックに解決可能です。そのため、内部向けロードバランサーは、ロードバランサー用に VPC へのアクセス権を持つクライアントからのみ、リクエストをルーティングできます。

インターネット向けロードバランサーと内部向けロードバランサーは、どちらもプライベート IP アドレス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。したがって、ターゲットは、内部またはインターネット向けロードバランサーからリクエストを受信するためのパブリック IP アドレスを必要としません。

アプリケーションに複数の層がある場合は、内部向けロードバランサーとインターネット向けロードバランサーを併用するアーキテクチャを設計できます。これはたとえば、アプリケーションで、インターネットに接続する必要があるウェブサーバーと、ウェブサーバーにのみ接続するアプリケーションサーバーを使用する場合に該当します。インターネット接続ロードバランサーを作成し、そこにウェブサーバーを登録します。内部ロードバランサーを作成し、そこにアプリケーションサーバーを登録します。ウェブサーバーは、インターネット接続ロードバランサーからリクエストを受け取り、アプリケーションサーバー用のリクエストを内部ロードバランサーに送信します。アプリケーションサーバーは、内部ロードバランサーからリクエストを受け取ります。

ロードバランサーのネットワーク MTU

最大送信単位 (MTU) は、ネットワーク上で送信できる最大のパケットサイズ (バイト単位) を決定します。接続の MTU が大きいほど、より多くのデータを単一のパケットで渡すことができます。イーサネットフレームは、パケット (送信している実際のデータ) とそれを囲むネットワークオーバーヘッド情報で構成されています。インターネットゲートウェイを介して送信されるトラフィックの MTU は 1500 です。つまり、パケットが 1500 バイトを超えている場合は、断片化されて複数のフレームを使用して送信されるか、`Don't Fragment` が IP ヘッダーに設定されていればドロップされます。

ロードバランサーノードの MTU サイズは設定できません。Application Load Balancer、Network Load Balancer、Classic Load Balancer のロードバランサーノード全体で、ジャンボフレーム (9001 MTU) は、標準になっています。Gateway Load Balancer は 8500 MTU をサポートします。詳細については、「Gateway Load Balancer のユーザーガイド」の「最大送信単位 (MTU)」を参照してください。。

パス MTU は、送信側ホストと受信側ホスト間のパスでサポートされている最大のパケットサイズです。2 つのデバイス間のパス MTU を判断するために、パス MTU 検出 (PMTUD) が使用されます。パス MTU 検出は、クライアントまたはターゲットがジャンボフレームをサポートしていない場合に特に重要です。

ホストがパスに沿って送信するパケットが、受信側ホストの MTU、あるいはデバイスの MTU よりも大きな場合、受信側ホストまたはデバイスはそのパケットを削除し、次のような ICMP メッセージ `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Type 3, Code 4)` を返します。このメッセージは送信側ホストに対し、ペイロードを複数の小さなパケットに分割し再送信することを指示します。

クライアントまたはターゲットインターフェイスの MTU サイズより大きいパケットが引き続き削除される場合、Path MTU 検出 (PMTUD) が機能していない可能性があります。これを回避するには、Path MTU 検出がエンドツーエンドで動作していること、およびクライアントおよびターゲットでジャンボフレームを有効にしていることを確認します。パス MTU 検出とジャンボフレームの有効化の詳細については、Amazon EC2 ユーザーガイド の [パス MTU 検出]を参照してください。

Elastic Load Balancing の使用開始

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。詳細については、製品の比較の を参照してください。

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。

既存の Classic Load Balancer がある場合は、Application Load Balancer または Network Load Balancer に移行できます。詳細については、「Classic Load Balancer の移行」を参照してください。

目次

Application Load Balancer の作成

AWS Management Console を使用して Application Load Balancer を作成するには、Application Load Balancer ユーザーガイド の Getting started with Application Load Balancers を参照してください。

AWS CLI を使用して Application Load Balancer を作成するには、Application Load Balancer ユーザーガイド の Create an Application Load Balancer using the AWS CLI を参照してください。

Network Load Balancer を作成する

AWS Management Console を使用して Network Load Balancer を作成するには、Network Load Balancer ユーザーガイド の Getting started with Network Load Balancers を参照してください。

AWS CLI を使用して Network Load Balancer を作成するには、Network Load Balancer ユーザーガイド の Create a Network Load Balancer using the AWS CLI を参照してください。

ゲートウェイロードバランサーを作成

AWS Management Console を使用してゲートウェイロードバランサーを作成するには、ゲートウェイロードバランサーのユーザーガイドの「ゲートウェイロードバランサーの使用開始」を参照してください。

AWS CLI を使用してゲートウェイロードバランサーを作成するには、ゲートウェイロードバランサーのユーザーガイドの「AWS CLI を使用したゲートウェイロードバランサーの使用開始」を参照してください。

Classic Load Balancer の作成

AWS Management Console を使用して Classic Load Balancer を作成するには、Classic Load Balancer ユーザーガイド の Create a Classic Load Balancer を参照してください。

Elastic Load Balancing のセキュリティ

AWS では、クラウドセキュリティが最優先事項です。AWS のお客様は、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャから利点を得られます。

セキュリティは、AWS とお客様の間の共有責任です。責任共有モデルでは、これをクラウドのセキュリティおよびクラウド内のセキュリティと説明しています。

- クラウドのセキュリティ - AWS は、AWS クラウドで AWS のサービスを実行するインフラストラクチャを保護する責任を負います。また、AWS は、使用するサービスを安全に提供します。AWS コンプライアンスプログラムの一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。Elastic Load Balancing に適用するコンプライアンスプログラムの詳細については、AWSコンプライアンスプログラムによる対象範囲内のサービス、を参照してください。
- クラウド内のセキュリティ – お客様の責任は使用する AWS のサービスによって決まります。また、お客様は、お客様のデータの機密性、企業の要件、および適用可能な法律および規制などの他の要因についても責任を担います。

このドキュメントは、Elastic Load Balancing を使用する際の責任共有モデルの適用方法を理解するのに役立ちます。ここでは、セキュリティとコンプライアンスの目標を満たすように Elastic Load Balancing を設定する方法について説明します。また、Elastic Load Balancing リソースのモニタリングや保護に役立つ他の AWS のサービスの使用方法についても説明します。

ゲートウェイロードバランサー では、アプライアンスベンダーのソフトウェアを選択して適格化する必要があります。ロードバランサーからのトラフィックを検査または変更するには、アプライアンスソフトウェアを信頼する必要があります。ロードバランサーは、OSI (Open Systems Interconnection) モデルのレイヤー 3 (ネットワークレイヤー) で動作します。Elastic Load Balancing パートナーとして一覧されているアプライアンスベンダーは、アプライアンスソフトウェアを AWS と統合し、認定を受けています。このリストのベンダーのアプライアンスソフトウェアには、より高いレベルの信頼を置くことができます。ただし、AWS は、これらのベンダーのソフトウェアのセキュリティまたは信頼性を保証するものではありません。

目次

- Elastic Load Balancing のコンプライアンス検証
- Elastic Load Balancing の復元性
- Elastic Load Balancing のインフラストラクチャセキュリティ
- インターフェイスエンドポイントを使用して Elastic Load Balancing にアクセスする (AWS PrivateLink)

Elastic Load Balancing のデータ保護

AWS https://aws.amazon.com/compliance/shared-responsibility-model/このモデルで説明したように、 AWS は、 AWS クラウドすべてを稼働させるグローバルインフラストラクチャを保護する責任があります。お客様は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。また、使用する AWS のサービス のセキュリティ設定と管理タスクもユーザーの責任となります。データプライバシーの詳細については、「データプライバシーのよくある質問」を参照してください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿された記事「AWS 責任共有モデルおよび GDPR」を参照してください。

データ保護のため、 AWS アカウント 認証情報を保護し、 AWS IAM Identity Center または AWS Identity and Access Management (IAM) を使用して個々のユーザーを設定することをお勧めします。この方法により、それぞれのジョブを遂行するために必要な権限のみが各ユーザーに付与されます。また、次の方法でデータを保護することもお勧めします:

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用してリソースと通信します。 AWS TLS 1.2 は必須であり TLS 1.3 がお勧めです。
- を使用して API とユーザーアクティビティのロギングを設定します。 AWS CloudTrail
- AWS 暗号化ソリューションと、 AWS のサービスその中に含まれるデフォルトのセキュリティコントロールをすべて使用してください。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API AWS を介してアクセスするときに FIPS 140-2 で検証された暗号モジュールが必要な場合は、FIPS エンドポイントを使用してください。利用可能な FIPS エンドポイントの詳細については、「連邦情報処理規格 (FIPS) 140-2」を参照してください。

お客様の E メールアドレスなどの極秘または機密情報は、タグ、または名前フィールドなどの自由形式のテキストフィールドに配置しないことを強くお勧めします。これには、Elastic Load

Balancing を使用する場合や、コンソール、API AWS CLI、または AWS SDK AWS のサービス を使用して作業する場合も含まれます。名前に使用する自由記述のテキストフィールドやタグに入力したデータは、課金や診断ログに使用される場合があります。外部サーバーへの URL を提供する場合は、そのサーバーへのリクエストを検証するための認証情報を URL に含めないように強くお勧めします。

保管中の暗号化

Elastic Load Balancing アクセスログの S3 バケットに対して、Amazon S3 管理の暗号化キーによるサーバー側の暗号化 (SSE-S3) を有効にした場合、Elastic Load Balancing は、S3 バケットに保存される前にアクセスログファイルを自動的に暗号化し、また、Elastic Load Balancing は、アクセスログファイルにアクセスするときに復号化を行います。各ログファイルは固有のキーで暗号化され、そのキー自体も定期的に更新される KMS キーで暗号化されます。

転送中の暗号化

Elastic Load Balancing は、ロードバランサーでクライアントからの HTTPS および TLS トラフィックを終了することで、セキュアなウェブアプリケーションの構築プロセスを簡素化します。ロードバランサーは、各 EC2 インスタンスに TLS ターミネーションの処理を要求する代わりに、トラフィックの暗号化と復号化の作業を実行します。セキュアリスナーを設定するときは、アプリケーションでサポートされている暗号スイートとプロトコルバージョン、およびロードバランサーにインストールするサーバー証明書を指定します。 AWS Certificate Manager (ACM) または AWS Identity and Access Management (IAM) を使用してサーバー証明書を管理できます。Application Load Balancer は HTTPS リスナーをサポートします。Network Load Balancer は TLS リスナーをサポートします。Classic Load Balancer は、HTTPS リスナーと TLS リスナーの両方をサポートします。

Elastic Load Balancing の Identity and Access Management

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御 AWS のサービス するのに役立つ です。IAM 管理者は、Elastic Load Balancing リソースの使用をユーザーに許可するために、認証 (サインイン) および承認 (アクセス許可を付与する) の制御を行います。IAM は、追加料金なしで AWS のサービス 使用できる です。

内容

- Elastic Load Balancing で利用できる IAM 機能
- Elastic Load Balancing API のアクセス許可
- リソース作成時にタグ付けするElastic Load Balancing API のアクセス許可
- Elastic Load Balancing のサービスにリンクされたロール
- AWS Elastic Load Balancing の マネージドポリシー

対象者

AWS Identity and Access Management (IAM) の使用方法は、Elastic Load Balancing で行う作業によって異なります。

Service user (サービスユーザー) – Elastic Load Balancing サービスを使用してジョブを実行する場合は、必要な認証情報とアクセス許可を管理者が用意します。さらに多くの Elastic Load Balancing 機能を使用して作業を行う場合は、追加のアクセス許可が必要になることがあります。アクセスの管理方法を理解しておくと、管理者に適切な許可をリクエストするうえで役立ちます。

Service administrator (サービス管理者) – 社内の Elastic Load Balancing リソースを担当している場合は、通常、Elastic Load Balancing へのフルアクセスがあります。サービスのユーザーがどの Elastic Load Balancing 機能やリソースにアクセスするかを決めるのは管理者の仕事です。その後、IAM 管理者にリクエストを送信して、サービスユーザーの権限を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してください。

IAM administrator (IAM 管理者) – IAM 管理者は、Elastic Load Balancing へのアクセスを管理するポリシーの作成方法の詳細について確認する場合があります。

アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。として、IAM ユーザーとして AWS アカウントのルートユーザー、または IAM ロールを引き受けて認証（にサインイン AWS) される必要があります。

ID ソースを介して提供された認証情報を使用して、フェデレーティッド ID AWS として にサインインできます。 AWS IAM Identity Center（IAM Identity Center) ユーザー、会社のシングルサインオン認証、Google または Facebook の認証情報は、フェデレーション ID の例です。フェデレーティッドアイデンティティとしてサインインする場合、IAM ロールを使用して、前もって管理者により ID フェデレーションが設定されています。フェデレーション AWS を使用して にアクセスすると、間接的にロールを引き受けることになります。

ユーザーのタイプに応じて、 AWS Management Console または AWS アクセスポータルにサインインできます。へのサインインの詳細については AWS、「 ユーザーガイド」の「 にサインインする方法 AWS アカウントAWS サインイン 」を参照してください。

AWS プログラムで にアクセスする場合、 は Software Development Kit (SDK) とコマンドラインインターフェイス (CLI) AWS を提供し、認証情報を使用してリクエストに暗号で署名します。 AWS ツールを使用しない場合は、リクエストに自分で署名する必要があります。推奨される方法を使用してリクエストを自分で署名する方法の詳細については、IAM ユーザーガイドの API AWS リクエストの署名を参照してください。

使用する認証方法を問わず、追加セキュリティ情報の提供をリクエストされる場合もあります。例えば、 AWS では、多要素認証 (MFA) を使用してアカウントのセキュリティを強化することをお勧めします。詳細については、『AWS IAM Identity Center ユーザーガイド』の「Multi-factor authentication」(多要素認証) および『IAM ユーザーガイド』の「AWSにおける多要素認証 (MFA) の使用」を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、アカウント内のすべての およびリソースへの AWS のサービス 完全なアクセス権を持つ 1 つのサインインアイデンティティから始めます。この ID は AWS アカウント ルートユーザーと呼ばれ、アカウントの作成に使用した E メールアドレスとパスワードでサインインすることでアクセスできます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザーの認証情報は保護し、ルートユーザーでしか実行できないタスクを実行するときに使用します。ルートユーザーとしてサインインする必要があるタスクの完全なリストについては、『IAM ユーザーガイド』の「ルートユーザー認証情報が必要なタスク」を参照してください。

フェデレーティッドアイデンティティ

ベストプラクティスとして、管理者アクセスを必要とするユーザーを含む人間のユーザーに、一時的な認証情報を使用して にアクセスするための ID プロバイダーとのフェデレーションの使用を要求 AWS のサービス します。

フェデレーティッド ID は、エンタープライズユーザーディレクトリ、ウェブ ID プロバイダー、AWS Directory Service、Identity Center ディレクトリのユーザー、または ID ソースを通じて提供された認証情報 AWS のサービス を使用して にアクセスするユーザーです。フェデレーティッド ID がにアクセスすると AWS アカウント、ロールを引き受け、ロールは一時的な認証情報を提供します。

アクセスを一元管理する場合は、 AWS IAM Identity Centerを使用することをお勧めします。IAM Identity Center でユーザーとグループを作成することも、独自の ID ソース内のユーザーとグルー

プのセットに接続して同期して、すべての AWS アカウント とアプリケーションで使用することもできます。IAM Identity Center の詳細については、『AWS IAM Identity Center ユーザーガイド』の「What is IAM Identity Center?」(IAM Identity Center とは) を参照してください。

IAM ユーザーとグループ

IAM ユーザーは、単一のユーザーまたはアプリケーションに対して特定のアクセス許可 AWS アカウント を持つ 内のアイデンティティです。可能であれば、パスワードやアクセスキーなどの長期的な認証情報を保有する IAM ユーザーを作成する代わりに、一時認証情報を使用することをお勧めします。ただし、IAM ユーザーでの長期的な認証情報が必要な特定のユースケースがある場合は、アクセスキーをローテーションすることをお勧めします。詳細については、「IAM ユーザーガイド」の「長期的な認証情報を必要とするユースケースのためにアクセスキーを定期的にローテーションする」を参照してください。

IAM グループは、IAM ユーザーの集団を指定するアイデンティティです。グループとしてサインインすることはできません。グループを使用して、複数のユーザーに対して一度に権限を指定できます。多数のユーザーグループがある場合、グループを使用することで権限の管理が容易になります。例えば、IAMAdmins という名前のグループを設定して、そのグループに IAM リソースを管理する権限を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 1 つのアプリケーションに一意に関連付けられますが、ロールはそれを必要とする任意の人が引き受けるようになっています。ユーザーには永続的な長期の認証情報がありますが、ロールでは一時的な認証情報が提供されます。詳細については、『IAM ユーザーガイド』の「IAM ユーザー (ロールではなく) の作成が適している場合」を参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可 AWS アカウント を持つ 内のアイデンティティです。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。ロール を切り替える AWS Management Console ことで、 で IAM ロールを一時的に引き受けることができます。ロールを引き受けるには、 または AWS API AWS CLI オペレーションを呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAM ユーザーガイド」の「IAM ロールの使用」を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます:

- フェデレーションユーザーアクセス – フェデレーティッドアイデンティティに権限を割り当てるには、ロールを作成してそのロールの権限を定義します。フェデレーティッドアイデンティティが認証されると、そのアイデンティティはロールに関連付けられ、ロールで定義されている権限

が付与されます。フェデレーションの詳細については、『IAM ユーザーガイド』の「サードパーティーアイデンティティプロバイダー向けロールの作成」 を参照してください。IAM アイデンティティセンターを使用する場合、権限セットを設定します。アイデンティティが認証後にアクセスできるものを制御するため、IAM Identity Center は、権限セットを IAM のロールに関連付けます。権限セットの詳細については、『AWS IAM Identity Center ユーザーガイド』の「権限セット」を参照してください。

- 一時的な IAM ユーザー権限 - IAM ユーザーまたはロールは、特定のタスクに対して複数の異なる権限を一時的に IAM ロールで引き受けることができます。
- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、 (ロールをプロキシとして使用する代わりに) ポリシーをリソースに直接アタッチできます。クロスアカウントアクセスにおけるロールとリソースベースのポリシーの違いについては、『IAM ユーザーガイド』の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。
- クロスサービスアクセス — 一部の は、他の の機能 AWS のサービス を使用します AWS のサービス。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの権限、サービスロール、またはサービスにリンクされたロールを使用してこれを行う場合があります。
  - 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用して でアクションを実行する場合 AWS、ユーザーはプリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストのリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。
  - サービスロール - サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービスに権限を委任するロールの作成」を参照してください。
  - サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行する

ロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールの権限を表示できますが、編集することはできません。

- Amazon EC2 で実行されているアプリケーション – IAM ロールを使用して、EC2 インスタンスで実行され、 AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。 AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、『IAM ユーザーガイド』の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して権限を付与する」を参照してください。

IAM ロールと IAM ユーザーのどちらを使用するかについては、『IAM ユーザーガイド』の「(IAM ユーザーではなく) IAM ロールをいつ作成したら良いのか?」を参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、 AWS ID またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられているときにアクセス許可を定義する のオブジェクトです。 は、プリンシパル (ユーザー、ルートユーザー、またはロールセッション) AWS がリクエストを行うときに、これらのポリシー AWS を評価します。ポリシーでの権限により、リクエストが許可されるか拒否されるかが決まります。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「IAM ユーザーガイド」の「JSON ポリシー概要」を参照してください。

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

デフォルトでは、ユーザーやロールに権限はありません。IAM 管理者は、リソースで必要なアクションを実行するための権限をユーザーに付与する IAM ポリシーを作成できます。その後、管理者はロールに IAM ポリシーを追加し、ユーザーはロールを引き継ぐことができます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの権限を定義します。例えば、`iam:GetRole` アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、 AWS Management Console、、 AWS CLIまたは AWS API からロール情報を取得できます。

アイデンティティベースのポリシー

アイデンティティベースポリシーは、IAM ユーザー、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 権限ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「IAM ポリシーの作成」を参照してください。

アイデンティティベースポリシーは、さらにインラインポリシーまたはマネージドポリシーに分類できます。インラインポリシーは、単一のユーザー、グループ、またはロールに直接埋め込まれています。管理ポリシーは、 内の複数のユーザー、グループ、ロールにアタッチできるスタンドアロンポリシーです AWS アカウント。管理ポリシーには、 AWS 管理ポリシーとカスタマー管理ポリシーが含まれます。マネージドポリシーまたはインラインポリシーのいずれかを選択する方法については、『IAM ユーザーガイド』の「マネージドポリシーとインラインポリシーの比較」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための権限を持つかをコントロールします。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、 AWS WAF、および Amazon VPC は、ACLs。ACL の詳細については、『Amazon Simple Storage Service デベロッパーガイド』の「アクセスコントロールリスト (ACL) の概要」を参照してください。

その他のポリシータイプ

AWS は、一般的ではない追加のポリシータイプをサポートします。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の権限を設定できます。

- アクセス許可の境界 - アクセス許可の境界は、アイデンティティベースのポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。エンティティにアクセス許可の境界を設定できます。結果として得られる権限は、エンティティのアイデンティティベースポリシーとそのアクセス許可の境界の共通部分になります。`Principal` フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、アクセス許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。許可の境界の詳細については、「IAM ユーザーガイド」の「IAM エンティティの許可の境界」を参照してください。
- サービスコントロールポリシー (SCPs) – SCPs は、 の組織または組織単位 (OU) に対する最大アクセス許可を指定する JSON ポリシーです AWS Organizations。 AWS Organizations は、 AWS アカウント ビジネスが所有する複数の をグループ化して一元管理するサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP は、各 を含むメンバーアカウントのエンティティのアクセス許可を制限します AWS アカウントのルートユーザー。Organizations と SCP の詳細については、『AWS Organizations ユーザーガイド』の「SCP の仕組み」を参照してください。
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「IAM ユーザーガイド」の「セッションポリシー」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される権限を理解するのがさらに難しくなります。複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、IAM ユーザーガイドの「ポリシー評価ロジック」を参照してください。

Elastic Load Balancing で利用できる IAM 機能

IAM を使用して Elastic Load Balancing へのアクセスを管理する前に、Elastic Load Balancing で使用できる IAM 機能について学びます。

Elastic Load Balancing で使用できる IAM 機能

IAM 機能	Elastic Load Balancing サポート
アイデンティティベースのポリシー	Yes
リソースベースのポリシー	No
ポリシーアクション	Yes
ポリシーリソース	はい
ポリシー条件キー (サービス固有)	はい
ACL	No
ABAC (ポリシー内のタグ)	はい
一時的な認証情報	Yes
プリンシパル権限	Yes
サービスロール	いいえ
サービスリンクロール	はい

Elastic Load Balancing のアイデンティティベースのポリシー

アイデンティティベースポリシーをサポートする	Yes

アイデンティティベースポリシーは、IAM ユーザー、ユーザーグループ、ロールなど、アイデンティティにアタッチできる JSON 権限ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティ

ベースのポリシーを作成する方法については、『IAM ユーザーガイド』の「IAM ポリシーの作成」を参照してください。

IAM アイデンティティベースのポリシーでは、許可または拒否するアクションとリソース、およびアクションを許可または拒否する条件を指定できます。プリンシパルは、それが添付されているユーザーまたはロールに適用されるため、アイデンティティベースのポリシーでは指定できません。JSON ポリシーで使用できるすべての要素については、「IAM ユーザーガイド」の「IAM JSON ポリシーの要素のリファレンス」を参照してください。

Elastic Load Balancing 内のリソースベースのポリシー

リソースベースのポリシーのサポート	No

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

クロスアカウントアクセスを有効にするには、アカウント全体、または別のアカウントの IAM エンティティをリソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる にある場合 AWS アカウント、信頼されたアカウントの IAM 管理者は、プリンシパルエンティティ (ユーザーまたはロール) にリソースへのアクセス許可も付与する必要があります。IAM 管理者は、アイデンティティベースのポリシーをエンティティにアタッチすることで権限を付与します。ただし、リソースベースのポリシーで、同じアカウントのプリンシパルへのアクセス権が付与されている場合は、アイデンティティベースのポリシーを追加する必要はありません。詳細については、『IAM ユーザーガイド』の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。

Elastic Load Balancing のポリシーアクション

ポリシーアクションに対するサポート	はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

JSON ポリシーのAction要素には、ポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。ポリシーアクションの名前は通常、関連付けられた AWS API オペレーションと同じです。一致する API オペレーションのない権限のみのアクションなど、いくつかの例外があります。また、ポリシーに複数アクションが必要なオペレーションもあります。これらの追加アクションは、 依存アクション と呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための権限を付与するポリシーで使用されます。

Elastic Load Balancing アクションのリストを表示するには、「サービス認可リファレンス」の「Elastic Load Balancing のアクション、リソース、および条件キー」を参照してください。

Elastic Load Balancing のポリシーアクションは、アクションの前に以下のプレフィックス を使用します。

```
elasticloadbalancing
```

単一のステートメントで複数のアクションを指定するには、アクションをカンマで区切ります。

```
"Action": [
    "elasticloadbalancing:action1",
    "elasticloadbalancing:action2"
]
```

ワイルドカード (*) を使用して複数アクションを指定できます。例えば、Describe という単語で始まるすべてのアクションを指定するには、次のアクションを含めます。

```
"Action": "elasticloadbalancing:Describe*"
```

Elastic Load Balancing の API アクションの詳細なリストについては、次のドキュメントを参照してください。

- Application Load Balancer、Network Load Balancer、および Gateway Load Balancer — API リファレンスバージョン 2015-12-01
- Classic Load Balancer — API リファレンスバージョン 2012-06-01

Elastic Load Balancing の各アクションに必要なアクセス許可の詳細については、Elastic Load Balancing API のアクセス許可 を参照してください。

Elastic Load Balancing のポリシーリソース

ポリシーリソースに対するサポート	はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースにどのような条件でアクションを実行できるかということです。

`Resource` JSON ポリシー要素は、アクションが適用されるオブジェクトを指定します。ステートメントには、`Resource` または `NotResource`要素を含める必要があります。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。これは、リソースレベルの権限と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

オペレーションのリスト化など、リソースレベルの権限をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

複数のリソースをサポートする Elastic Load Balancing API アクションがあります。複数リソースを単一ステートメントで指定するには、ARN をカンマで区切ります。

```
"Resource": [
    "resource1",
    "resource2"
]
```

Elastic Load Balancing リソースタイプとその ARN のリストを確認するには、「サービス認可リファレンス」の「Resources defined by Elastic Load Balancing」(Elastic Load Balancing で定義されるリソースタイプ) を参照してください。　各リソースの ARN を指定できるアクションについては、「Elastic Load Balancing のアクション、リソース、および条件キー」を参照してください。

Elastic Load Balancing のポリシー条件キー

サービス固有のポリシー条件キーのサポート	はい

管理者は AWS JSON ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどんなリソースにどんな条件でアクションを実行できるかということです。

Condition 要素 (または Condition ブロック) を使用すると、ステートメントが有効な条件を指定できます。Condition 要素はオプションです。イコールや未満などの 条件演算子 を使用して条件式を作成することで、ポリシーの条件とリクエスト内の値を一致させることができます。

1 つのステートメントに複数の Condition 要素を指定するか、1 つの Condition 要素に複数のキーを指定すると、 AWS は AND 論理演算子を使用してそれらを評価します。1 つの条件キーに複数の値を指定すると、 は論理ORオペレーションを使用して条件 AWS を評価します。ステートメントの権限が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、IAM ユーザー名がタグ付けされている場合のみリソースにアクセスできる権限を付与することができます。詳細については、『IAM ユーザーガイド』の「IAM ポリシーの要素: 変数およびタグ」を参照してください。

AWS は、グローバル条件キーとサービス固有の条件キーをサポートします。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「 グローバル条件コンテキストキー」を参照してください。

Elastic Load Balancing の条件キーのリストを確認するには、「サービス認証リファレンス」の「Elastic Load Balancing のアクション、リソース、および条件キー」の条件キーを参照してください。条件キーを使用できるアクションおよびリソースについては、「Elastic Load Balancing のアクション、リソース、および条件キー」を参照してください。

elasticloadbalancing:ResourceTag 条件キー

elasticloadbalancing:ResourceTag/## 条件キーは Elastic Load Balancing 固有です。以下のアクションでこの条件キーがサポートされています。

API バージョン 2015-12-01

- AddTags
- CreateListener
- CreateLoadBalancer
- DeleteLoadBalancer
- DeleteTargetGroup

- DeregisterTargets
- ModifyLoadBalancerAttributes
- ModifyTargetGroup
- ModifyTargetGroupAttributes
- RegisterTargets
- RemoveTags
- SetIpAddressType
- SetSecurityGroups
- SetSubnets

API バージョン 2012-06-01

- AddTags
- ApplySecurityGroupsToLoadBalancer
- AttachLoadBalancersToSubnets
- ConfigureHealthCheck
- CreateAppCookieStickinessPolicy
- CreateLBCookieStickinessPolicy
- CreateLoadBalancer
- CreateLoadBalancerListeners
- CreateLoadBalancerPolicy
- DeleteLoadBalancer
- DeleteLoadBalancerListeners
- DeleteLoadBalancerPolicy
- DeregisterInstancesFromLoadBalancer
- DetachLoadBalancersFromSubnets
- DisableAvailabilityZonesForLoadBalancer
- EnableAvailabilityZonesForLoadBalancer
- ModifyLoadBalancerAttributes

- `RegisterInstancesWithLoadBalancer`
- `RemoveTags`
- `SetLoadBalancerListenerSSLCertificate`
- `SetLoadBalancerPoliciesForBackendServer`
- `SetLoadBalancerPoliciesOfListener`

`elasticloadbalancing:ListenerProtocol` 条件キー

`elasticloadbalancing:ListenerProtocol` 条件キーは、作成して使用できるリスナーのタイプを定義する条件に使用できます。以下のアクションでこの条件キーがサポートされています。

API バージョン 2015-12-01

- `CreateListener`
- `ModifyListener`

API バージョン 2012-06-01

- `CreateLoadBalancer`
- `CreateLoadBalancerListeners`

このポリシーは、Application Load Balancer、Network Load Balancer、および Classic Load Balancer で使用できます。以下は、リスナーに指定されたプロトコルの 1 つのみをユーザーが選択できるようにするポリシーの例です。

サポートされるプロトコル:

- HTTPS
- HTTP
- TCP
- SSL
- TLS
- UDP
- TCP_UDP

```
"Version": "2015-12-01",
    "Statement": {"Effect": "Allow",
        "Action": [
            "elasticloadbalancing:CreateListener",
            "elasticloadbalancing:ModifyListener"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals":{
                    "elasticloadbalancing:ListenerProtocol": [
                    "HTTPS",
                    "TLS"
                    ]
            },
        }
```

`elasticloadbalancing:SecurityPolicy` 条件キー

`elasticloadbalancing:SecurityPolicy` 条件キーは、ロードバランサーに特定のセキュリティポリシーを定義して適用する条件に使用できます。以下のアクションでこの条件キーがサポートされています。

API バージョン 2015-12-01

- `CreateListener`
- `ModifyListener`

API バージョン 2012-06-01

- `CreateLoadBalancerPolicy`
- `SetLoadBalancerPoliciesOfListener`

このポリシーは、Application Load Balancer、Network Load Balancer、Classic Load Balancer で使用できます。以下は、ユーザーがロードバランサーに指定されたセキュリティポリシーの 1 つだけを選択できるようにするポリシーの例です。

```
"Resource": [
"Version": "2015-12-01",
    "Statement": {"Effect": "Allow",
        "Action": [
```

```
            "elasticloadbalancing:CreateListener",
            "elasticloadbalancing:ModifyListener"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEquals":{
                "elasticloadbalancing:SecurityPolicy": [
                    "ELBSecurityPolicy-TLS13-1-2-2021-06",
                    "ELBSecurityPolicy-TLS13-1-2-Res-2021-06",
                    "ELBSEcurityPolicy-TLS13-1-1-2021-06"
                ]
            },
        }
]
```

`elasticloadbalancing:Scheme` 条件キー

`elasticloadbalancing:Scheme` 条件キーは、ロードバランサーの作成時に選択できるスキームを定義する条件に使用できます。以下のアクションでこの条件キーがサポートされています。

API バージョン 2015-12-01

- `CreateLoadBalancer`

API バージョン 2012-06-01

- `CreateLoadBalancer`

このポリシーは、Application Load Balancer、Network Load Balancer、および Classic Load Balancer で使用できます。以下は、ユーザーがロードバランサーに指定されたスキームの 1 つを選択することのみを許可するポリシーの例です。

```
"Version": "2015-12-01",
    "Statement": {"Effect": "Allow",
        "Action":"elasticloadbalancing:CreateLoadBalancer",
        "Resource": "*",
        "Condition": {
            "StringEquals":{
                "elasticloadbalancing:Scheme": "internal"
            },
        }
```

`elasticloadbalancing:Subnet` 条件キー

> **Important**
>
> Elastic Load Balancing は、サブネット IDs。ただし、大文字と小文字を区別しない適切な条件演算子を使用してください。例: `StringEqualsIgnoreCase`。

`elasticloadbalancing:Subnet` 条件キーは、作成してロードバランサーにアタッチできるサブネットを定義する条件に使用できます。以下のアクションでこの条件キーがサポートされています。

API バージョン 2015-12-01

- `CreateLoadBalancer`
- `SetSubnets`

API バージョン 2012-06-01

- `CreateLoadBalancer`
- `AttachLoadBalancerToSubnets`

このポリシーは、Application Load Balancer、Network Load Balancer、Gateway Load Balancer、Classic Load Balancer で使用できます。以下は、ユーザーがロードバランサーに指定されたサブネットの 1 つだけを選択できるようにするポリシーの例です。

```
"Version": "2015-12-01",
    "Statement": {"Effect": "Allow",
        "Action": [
            "elasticloadbalancing:CreateLoadBalancer",
            "elasticloadbalancing:SetSubnets"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEqualsIgnoreCase":{
                "elasticloadbalancing:Subnet": [
                    "subnet-01234567890abcdef",
                    "subnet-01234567890abcdeg "
                ]
            },
```

```
        }
```

elasticloadbalancing:SecurityGroup 条件キー

> **Important**
>
> Elastic Load Balancing は、 SecurityGroup IDs。ただし、大文字と小文字を区別しない適切な条件演算子を使用してください。例: `StringEqualsIgnoreCase`。

`elasticloadbalancing:SecurityGroup` 条件キーは、ロードバランサーに適用できるセキュリティグループを定義する条件に使用できます。以下のアクションでこの条件キーがサポートされています。

API バージョン 2015-12-01

- `CreateLoadBalancer`
- `SetSecurityGroups`

API バージョン 2012-06-01

- `CreateLoadBalancer`
- `ApplySecurityGroupsToLoadBalancer`

このポリシーは、Application Load Balancer、Network Load Balancer、Classic Load Balancer で使用できます。以下は、ユーザーがロードバランサーに指定されたセキュリティグループの 1 つを選択できるようにするポリシーの例です。

```
"Version": "2015-12-01",
    "Statement": {"Effect": "Allow",
        "Action": [
            "elasticloadbalancing:CreateLoadBalancer",
            "elasticloadbalancing:SetSecurityGroup"
        ],
        "Resource": "*",
        "Condition": {
            "ForAnyValue:StringEqualsIgnoreCase":{
                "elasticloadbalancing:SecurityGroup": [
                    "sg-51530134",
```

```
                    "sg-51530144",
                    "sg-51530139"
                ]
            },
        }
```

Elastic Load Balancing の ACL

ACL のサポート	No

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための権限を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Elastic Load Balancing での ABAC

ABAC のサポート (ポリシー内のタグ)	はい

属性ベースのアクセス制御 (ABAC) は、属性に基づいてアクセス許可を定義するアクセス許可戦略です。では AWS、これらの属性はタグ と呼ばれます。タグは、IAM エンティティ (ユーザーまたはロール) および多くの AWS リソースにアタッチできます。エンティティとリソースのタグ付けは、ABAC の最初の手順です。その後、プリンシパルのタグがアクセスしようとしているリソースのタグと一致した場合に操作を許可するように ABAC ポリシーを設計します。

ABAC は、急成長する環境やポリシー管理が煩雑になる状況で役立ちます。

タグに基づいてアクセスを管理するには、`aws:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

サービスがすべてのリソースタイプに対して 3 つの条件キーすべてをサポートする場合、そのサービスの値ははいです。サービスが一部のリソースタイプに対してのみ 3 つの条件キーのすべてをサポートする場合、値は「部分的」になります。

ABAC の詳細については、『IAM ユーザーガイド』の「ABAC とは?」を参照してください。ABAC をセットアップするステップを説明するチュートリアルについては、「IAM ユーザーガイド」の「属性に基づくアクセスコントロール (ABAC) を使用する」を参照してください。

Elastic Load Balancing での一時的な認証情報の使用

一時的な認証情報のサポート	はい

一部の は、一時的な認証情報を使用してサインインすると機能 AWS のサービス しません。一時的な認証情報 AWS のサービス を使用する などの詳細については、IAM ユーザーガイドのAWS のサービス 「IAM と連携する 」を参照してください。

ユーザー名とパスワード以外の AWS Management Console 方法で にサインインする場合、一時的な認証情報を使用します。例えば、会社の Single Sign-On (SSO) リンク AWS を使用して にアクセスすると、そのプロセスによって一時的な認証情報が自動的に作成されます。また、ユーザーとしてコンソールにサインインしてからロールを切り替える場合も、一時的な認証情報が自動的に作成されます。ロールの切り替えに関する詳細については、「IAM ユーザーガイド」の「ロールへの切り替え (コンソール)」を参照してください。

一時的な認証情報は、 AWS CLI または AWS API を使用して手動で作成できます。その後、これらの一時的な認証情報を使用して . AWS recommends にアクセスできます AWS。これは、長期的なアクセスキーを使用する代わりに、一時的な認証情報を動的に生成することを推奨しています。詳細については、「IAM の一時的セキュリティ認証情報」を参照してください。

Elastic Load Balancing のクロスサービスプリンシパル許可

フォワードアクセスセッション (FAS) をサポート	はい

IAM ユーザーまたはロールを使用して でアクションを実行すると AWS、プリンシパルと見なされます。一部のサービスを使用する際に、アクションを実行することで、別のサービスの別のアクションがトリガーされることがあります。FAS は、 を呼び出すプリンシパルのアクセス許可を AWS のサービス、ダウンストリームサービス AWS のサービス へのリクエストのリクエストと組み合わせて使用します。FAS リクエストは、サービスが他の AWS のサービス またはリソースとのやり取りを完了する必要があるリクエストを受け取った場合にのみ行われます。この場合、両方のアクションを実行するためのアクセス許可が必要です。FAS リクエストを行う際のポリシーの詳細については、「転送アクセスセッション」を参照してください。

Elastic Load Balancing のサービスロール

サービスロールのサポート	いいえ

サービスロールとは、サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、「IAM ユーザーガイド」の「AWS のサービスに権限を委任するロールの作成」を参照してください。

Elastic Load Balancing のサービスリンクロール

サービスリンクロールのサポート	はい

サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールの権限を表示できますが、編集することはできません。

Elastic Load Balancing サービスにリンクされたロールの作成または管理の詳細については、「Elastic Load Balancing のサービスにリンクされたロール」を参照してください。

Elastic Load Balancing API のアクセス許可

必要な Elastic Load Balancing API アクションを呼び出すアクセス許可を ユーザーに付与する必要があります。また、一部の Elastic Load Balancing アクションでは、Amazon EC2 API から特定のアクションを呼び出すアクセス許可をユーザーに付与する必要があります。

2015-12-01 API に必要なアクセス許可

2015-12-01 API から次のアクションを呼び出す場合は、指定されたアクションを呼び出すアクセス許可をユーザーに付与する必要があります。

`CreateLoadBalancer`

- `elasticloadbalancing:CreateLoadBalancer`
- `ec2:DescribeAccountAttributes`
- `ec2:DescribeAddresses`

- ec2:DescribeInternetGateways
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeVpcs
- iam:CreateServiceLinkedRole

CreateTargetGroup

- elasticloadbalancing:CreateTargetGroup
- ec2:DescribeInternetGateways
- ec2:DescribeVpcs

RegisterTargets

- elasticloadbalancing:RegisterTargets
- ec2:DescribeInstances
- ec2:DescribeInternetGateways
- ec2:DescribeSubnets
- ec2:DescribeVpcs

SetIpAddressType

- elasticloadbalancing:SetIpAddressType
- ec2:DescribeSubnets

SetSubnets

- elasticloadbalancing:SetSubnets
- ec2:DescribeSubnets

2012-06-01 API に必要なアクセス許可

2012-06-01 API から次のアクションを呼び出す場合は、指定されたアクションを呼び出すアクセス許可をユーザーに付与する必要があります。

ApplySecurityGroupsToLoadBalancer

- elasticloadbalancing:ApplySecurityGroupsToLoadBalancer
- ec2:DescribeAccountAttributes
- ec2:DescribeSecurityGroups

AttachLoadBalancerToSubnets

- elasticloadbalancing:AttachLoadBalancerToSubnets
- ec2:DescribeSubnets

CreateLoadBalancer

- elasticloadbalancing:CreateLoadBalancer
- ec2:CreateSecurityGroup
- ec2:DescribeAccountAttributes
- ec2:DescribeInternetGateways
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeVpcs
- iam:CreateServiceLinkedRole

DeregisterInstancesFromLoadBalancer

- elasticloadbalancing:DeregisterInstancesFromLoadBalancer
- ec2:DescribeClassicLinkInstances
- ec2:DescribeInstances

DescribeInstanceHealth

- elasticloadbalancing:DescribeInstanceHealth
- ec2:DescribeClassicLinkInstances
- ec2:DescribeInstances

DescribeLoadBalancers

- elasticloadbalancing:DescribeLoadBalancers
- ec2:DescribeSecurityGroups

DisableAvailabilityZonesForLoadBalancer

- elasticloadbalancing:DisableAvailabilityZonesForLoadBalancer
- ec2:DescribeAccountAttributes
- ec2:DescribeInternetGateways
- ec2:DescribeVpcs

EnableAvailabilityZonesForLoadBalancer

- elasticloadbalancing:EnableAvailabilityZonesForLoadBalancer

- ec2:DescribeAccountAttributes
- ec2:DescribeInternetGateways
- ec2:DescribeSubnets
- ec2:DescribeVpcs

RegisterInstancesWithLoadBalancer

- elasticloadbalancing:RegisterInstancesWithLoadBalancer
- ec2:DescribeAccountAttributes
- ec2:DescribeClassicLinkInstances
- ec2:DescribeInstances
- ec2:DescribeVpcClassicLink

リソース作成時にタグ付けするElastic Load Balancing API のアクセス許可

ユーザーが作成時にタグを付けるには、elasticloadbalancing:CreateLoadBalancer または elasticloadbalancing:CreateTargetGroup などのリソースを作成するアクションを使用するためのアクセス許可が必要です。リソース作成アクションでタグが指定されている場合、作成されたリソースにタグを適用するための認可をユーザーが持っているかどうか確認するため、elasticloadbalancing:AddTags アクションで追加の承認が必要です したがって、ユーザーは elasticloadbalancing:AddTags アクションを使用するための明示的な許可も持っている必要があります。

elasticloadbalancing:AddTags アクションの IAM ポリシー定義で、Condition 要素をelasticloadbalancing:CreateAction 条件キーと使用して、リソース作成アクションにタグ付けのアクセス許可を付与します。

次の例では、ユーザーがターゲットグループを作成し、作成中に任意のタグを適用できるポリシーを示しています。ユーザーには、既存のリソースへのタグ付けが許可されません (elasticloadbalancing:AddTags アクションを直接呼び出すことはできません)。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:CreateTargetGroup"
      ],
```

```
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:AddTags"
      ],
      "Resource": "*",
      "Condition": {
         "StringEquals": {
             "elasticloadbalancing:CreateAction" : "CreateTargetGroup"
          }
       }
    }
  ]
}
```

同様に、次のポリシーでは、ユーザーはロードバランサーの作成と作成時のタグの適用が可能になります。ユーザーには、既存のリソースへのタグ付けが許可されません (`elasticloadbalancing:AddTags` アクションを直接呼び出すことはできません)。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:CreateLoadBalancer"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "elasticloadbalancing:AddTags"
      ],
      "Resource": "*",
      "Condition": {
         "StringEquals": {
             "elasticloadbalancing:CreateAction" : "CreateLoadBalancer"
          }
       }
```

```
    }
  ]
}
```

elasticloadbalancing:AddTags アクションは、タグがリソース作成アクション時に適用された場合のみ評価されます。したがって、リクエストでタグが指定されていない場合、リソースを作成するアクセス許可を持っているユーザー (タグ付け条件がないと仮定) には、elasticloadbalancing:AddTags アクションを実行するアクセス許可は必要ありません。ただし、ユーザーがタグを使用してリソースを作成しようとした場合、ユーザーが elasticloadbalancing:AddTags アクションを使用するアクセス許可を持っていない場合はリクエストに失敗します。

Elastic Load Balancing のサービスにリンクされたロール

Elastic Load Balancing は、Elastic Load Balancing がユーザーに代わって他の AWS サービスを呼び出すために必要なアクセス許可を持つサービスにリンクされたロールを使用します。詳細については、IAM ユーザーガイドの「サービスにリンクされたロールの使用」を参照してください。

サービスにリンクされたロールによって付与されるアクセス許可

Elastic Load Balancing は、 という名前のサービスにリンクされたロールAWSServiceRoleForElasticLoadBalancingを使用して、ユーザーに代わって次のアクションを呼び出します。

- ec2:AssignIpv6Addresses
- ec2:AssignPrivateIpAddresses
- ec2:AssociateAddress
- ec2:AttachNetworkInterface
- ec2:AuthorizeSecurityGroupIngress
- ec2:CreateNetworkInterface
- ec2:CreateSecurityGroup
- ec2:DeleteNetworkInterface
- ec2:DescribeAccountAttributes
- ec2:DescribeAddresses
- ec2:DescribeClassicLinkInstances
- ec2:DescribeCoipPools

- `ec2:DescribeInstances`
- `ec2:DescribeInternetGateways`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcClassicLink`
- `ec2:DescribeVpcPeeringConnections`
- `ec2:DescribeVpcs`
- `ec2:DetachNetworkInterface`
- `ec2:DisassociateAddress`
- `ec2:GetCoipPoolUsage`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:ReleaseAddress`
- `ec2:UnassignIpv6Addresses`
- `logs:CreateLogDelivery`
- `logs:DeleteLogDelivery`
- `logs:GetLogDelivery`
- `logs:ListLogDeliveries`
- `logs:UpdateLogDelivery`
- `outposts:GetOutpostInstanceTypes`

AWSServiceRoleForElasticLoadBalancing は、 `elasticloadbalancing.amazonaws.com`サービスを信頼してロールを引き受けます。

サービスにリンクされたロールの作成

AWSServiceRoleForElasticLoadBalancing ロールを手動で作成する必要はありません。このロールは、ロードバランサーまたはターゲットグループの作成時に Elastic Load Balancing によって作成されます。

Elastic Load Balancing がユーザーに代わってサービスにリンクされたロールを作成するには、必要なアクセス許可がユーザーに付与されていなければなりません。詳細については、IAM ユーザーガイド の「サービスにリンクされたロールのアクセス許可」を参照してください。

2018 年 1 月 11 日より前にロードバランサーを作成した場合、AWSServiceRoleForElasticLoadBalancingアカウントに AWS Elastic Load Balancing が作成されます。詳細については、「IAM ユーザーガイド」の AWS 「アカウントに新しいロールが表示されました」を参照してください。

サービスにリンクされたロールを編集する

IAM AWSServiceRoleForElasticLoadBalancingを使用して の説明を編集できます。詳細については、IAM ユーザーガイドの「サービスにリンクされたロールの編集」を参照してください。

サービスにリンクされたロールを削除する

Elastic Load Balancing を使用する必要がなくなった場合は、 を削除することをお勧めしますAWSServiceRoleForElasticLoadBalancing。

このサービスにリンクされたロールは、 AWS アカウント内のすべてのロードバランサーを削除した後にのみ削除できます。これにより、ロードバランサーへのアクセス許可を誤って削除することがなくなります。詳細については、Application Load Balancer の削除、Network Load Balancer の削除、およびClassic Load Balancer の削除を参照してください。

サービスにリンクされたロールは、IAM コンソール、IAM CLI、または IAM API を使用して削除することができます。詳細については、IAM ユーザーガイド の「サービスにリンクされたロールの削除」を参照してください。

を削除するとAWSServiceRoleForElasticLoadBalancing、ロードバランサーを作成すると、Elastic Load Balancing によってロールが再度作成されます。

AWS Elastic Load Balancing の マネージドポリシー

AWS 管理ポリシーは、 によって作成および管理されるスタンドアロンポリシーです AWS。 AWS 管理ポリシーは、多くの一般的なユースケースにアクセス許可を付与するように設計されているため、ユーザー、グループ、ロールにアクセス許可の割り当てを開始できます。

AWS 管理ポリシーは、すべての AWS お客様が使用できるため、特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることに注意してください。ユースケース別にカスタマーマネージドポリシーを定義して、マネージドポリシーを絞り込むことをお勧めします。

AWS 管理ポリシーで定義されているアクセス許可は変更できません。が AWS 管理ポリシーで定義されたアクセス許可 AWS を更新すると、ポリシーがアタッチされているすべてのプリンシパル ID (ユーザー、グループ、ロール) が更新されます。 は、新しい AWS のサービス が起動されたとき、

または既存のサービスで新しい API AWS オペレーションが使用可能になったときに、 AWS 管理ポリシーを更新する可能性が最も高くなります。

詳細については、「IAM ユーザーガイド」の「AWS マネージドポリシー」を参照してください。

AWS マネージドポリシー： AWSElasticLoadBalancingClassicServiceRolePolicy

このポリシーには、Elastic Load Balancing (Classic Load Balancer) がユーザーに代わって他の AWS サービスを呼び出すために必要なすべてのアクセス許可が含まれています。サービスリンクロールは事前定義されています。事前定義されたロールを使用すると、Elastic Load Balancing がユーザーに代わってアクションを完了するために必要とする許可を手動で追加する必要がなくなります。このポリシーをアタッチ、デタッチ、変更、または削除することはできません。

このポリシーのアクセス許可を確認するには、「 マネージドポリシーリファレンスAWSElasticLoadBalancingClassicServiceRolePolicy」の「」を参照してください。 AWS

AWS マネージドポリシー: AWSElasticLoadBalancingServiceRolePolicy

このポリシーには、Elastic Load Balancing がユーザーに代わって AWS のその他サービスを呼び出すために必要とするすべての許可が含まれています。サービスリンクロールは事前定義されています。事前定義されたロールを使用すると、Elastic Load Balancing がユーザーに代わってアクションを完了するために必要とする許可を手動で追加する必要がなくなります。このポリシーをアタッチ、デタッチ、変更、または削除することはできません。

このポリシーのアクセス許可を確認するには、「 マネージドポリシーリファレンスAWSElasticLoadBalancingServiceRolePolicy」の「」を参照してください。 AWS

AWS マネージドポリシー: ElasticLoadBalancingFullAccess

このポリシーは、Elastic Load Balancing サービスへのフルアクセスと、 AWS マネジメントコンソールを介した他のサービスへの制限付きアクセスを許可します。

このポリシーのアクセス許可を確認するには、「 マネージドポリシーリファレンスElasticLoadBalancingFullAccess」の「」を参照してください。 AWS

AWS マネージドポリシー: ElasticLoadBalancingReadOnly

このポリシーは、Elastic Load Balancing および依存サービスに対する読み取り専用アクセス権を付与します。

このポリシーのアクセス許可を確認するには、「 マネージドポリシーリファレンスElasticLoadBalancingReadOnly」の「」を参照してください。 AWS

Elastic Load Balancing の AWS マネージドポリシーの更新

このサービスがこれらの変更の追跡を開始してからの Elastic Load Balancing の AWS マネージドポリシーの更新に関する詳細を表示します。

変更	説明	日付
AWS マネージドポリシー: ElasticLoadBalancingFullAccess – 既存ポリシーへの更新。	Elastic Load Balancing に、ゾーンシフトを使用するためのアクセス許可を付与する新しいアクションが追加されました。このアクションは、Elastic Load Balancing フルアクセスポリシーに追加されました。これは、`arc-zonal-shift:*` API オペレーションに関連付けられています。	2022 年 11 月 28 日
AWS マネージドポリシー: ElasticLoadBalancingReadOnly – 既存ポリシーへの更新。	Elastic Load Balancing に、ゾーンシフトを使用するためのアクセス許可を付与する新しいアクションが追加されました。このアクションは、Elastic Load Balancing 読み取り専用ポリシーに追加されました。これは、`arc-zonal-shift:GetManagedResource` 、`arc-zonal-shift:ListManagedResources` 、および `arc-zonal-shift:ListZonalShifts` API 操作に関連付けられています。	2022 年 11 月 28 日
AWS マネージドポリシー: AWSElasticLoadBalancingServiceRolePolicy – 既存ポリシーへの更新。	Elastic Load Balancing が、ピアリング接続を使用するための許可を付与する新しいアクションを追加しました。このアクションは、Elastic Load Balancing コントロールプレーンのために、サービスリンクロールポリシーに追加されました。これは、`ec2:DescribeVpcPeeringConnections` API 操作に関連付けられています。	2021 年 10 月 11 日
AWS マネージドポリシー: ElasticLoadBalancingFullAccess – 既存ポリシーへの更新。	Elastic Load Balancing が、ピアリング接続を使用するための許可を付与する新しいアクションを追加しました。このアクションは、Elastic	2021 年 10 月 11 日

変更	説明	日付
	Load Balancing フルアクセスポリシーに追加されました。これは、`ec2:DescribeVpcPeeringConnections` API 操作に関連付けられています。	
AWS マネージドポリシー：AWSElasticLoadBalancingClassicServiceRolePolicy – 既存ポリシーへの更新。	Elastic Load Balancing が、Classic Load Balancer 用のサービスリンクロールポリシー (コントロールプレーン向け) を追加しました。これは、バージョン 2 (デフォルト) に対する更新です。	2019 年 10 月 7 日
AWS マネージドポリシー: ElasticLoadBalancingReadOnly	Elastic Load Balancing および依存サービスへの読み取り専用アクセス権を付与。これはバージョン 1 (デフォルト) です。	2018 年 9 月 20 日
Elastic Load Balancing が変更の追跡を開始	Elastic Load Balancing が AWS マネージドポリシーの変更の追跡を開始しました。	2021 年 7 月 23 日

Elastic Load Balancing のコンプライアンス検証

AWS のサービス が特定のコンプライアンスプログラムの範囲内にあるかどうかを確認するには、コンプライアンスプログラムAWS のサービス による対象範囲内のコンプライアンスプログラムを参照し、関心のあるコンプライアンスプログラムを選択します。一般的な情報については、AWS 「コンプライアンスプログラム」を参照してください。

を使用して、サードパーティーの監査レポートをダウンロードできます AWS Artifact。詳細については、「 でのレポートのダウンロード AWS Artifact」の」を参照してください。

を使用する際のお客様のコンプライアンス責任 AWS のサービス は、お客様のデータの機密性、貴社のコンプライアンス目的、適用される法律および規制によって決まります。 では、コンプライアンスに役立つ以下のリソース AWS を提供しています。

- セキュリティとコンプライアンスのクイックスタートガイド – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境 AWS を にデプロイする手順について説明します。

- アマゾン ウェブ サービスにおける HIPAA セキュリティとコンプライアンスのアーキテクチャ — このホワイトペーパーでは、企業が AWS を使用して HIPAA 対象アプリケーションを作成する方法について説明します。

> **Note**
>
> すべて AWS のサービス HIPAA の対象となるわけではありません。詳細については、「HIPAA 対応サービスのリファレンス」を参照してください。

- AWS コンプライアンスリソース – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- AWS カスタマーコンプライアンスガイド — コンプライアンスの観点から責任共有モデルを理解します。このガイドでは、ガイダンスを保護し AWS のサービス 、複数のフレームワーク (米国国立標準技術研究所 (NIST)、Payment Card Industry Security Standards Council (PCI)、国際標準化機構 (ISO) を含む) のセキュリティコントロールにマッピングするためのベストプラクティスをまとめています。
- 「 デベロッパーガイド」の「 ルールによるリソースの評価」 – この AWS Config サービスは、リソース設定が社内プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。 AWS Config
- AWS Security Hub – これにより AWS のサービス 、 内のセキュリティ状態を包括的に確認できます AWS。Security Hub では、セキュリティコントロールを使用して AWS リソースを評価し、セキュリティ業界標準とベストプラクティスに対するコンプライアンスをチェックします。サポートされているサービスとコントロールのリストについては、「Security Hub のコントロールリファレンス」を参照してください。
- Amazon GuardDuty – これにより AWS アカウント、疑わしいアクティビティや悪意のあるアクティビティがないか環境を監視することで、、、ワークロード、コンテナ、データに対する潜在的な脅威 AWS のサービス を検出します。 GuardDuty は、特定のコンプライアンスフレームワークで義務付けられている侵入検知要件を満たすことで、PCI DSS などのさまざまなコンプライアンス要件への対応に役立ちます。
- AWS Audit Manager – これにより AWS のサービス 、 AWS 使用状況を継続的に監査し、リスクの管理方法と規制や業界標準への準拠を簡素化できます。

Elastic Load Balancing の復元性

AWS のグローバルインフラストラクチャは AWS リージョンとアベイラビリティーゾーンを中心として構築されます。リージョンには、低レイテンシー、高いスループット、そして高度の冗長ネッ

トワークで接続されている複数の物理的に独立および隔離されたアベイラビリティーゾーンがあります。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェイルオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性、耐障害性、および拡張性が優れています。

AWS リージョンとアベイラビリティーゾーンの詳細については、「AWS グローバルインフラストラクチャ」を参照してください。

AWS グローバルインフラストラクチャに加え、Elastic Load Balancing が、データ耐障害性をサポートする以下の機能を提供します。

- 1 つまたは複数のアベイラビリティーゾーンにある複数のインスタンスの間で受信トラフィックを分散する機能
- AWS Global Accelerator を Application Load Balancer と一緒に使用すると、1 つ以上の AWS リージョンの複数のロードバランサーに着信トラフィックを分散できます。詳細については、「AWS Global Accelerator デベロッパーガイド」を参照してください。
- Amazon ECS により、EC2 インスタンスのクラスターで Docker コンテナを実行、停止、管理できます。ロードバランサーを使用して、クラスター内のサービス全体に受信トラフィックを分散させるように、Amazon ECS サービスを設定できます。詳細については、Amazon Elastic Container Service デベロッパーガイドを参照してください。

Elastic Load Balancing のインフラストラクチャセキュリティ

マネージドサービスである Elastic Load Balancing は AWS グローバルネットワークセキュリティで保護されています。AWS のセキュリティサービスと、AWS がインフラストラクチャをどのように保護するかについては、「AWS クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「セキュリティの柱 - AWS Well-Architected フレームワーク」の「インフラストラクチャ保護」を参照してください。

AWS が公開している API コールを使用して、ネットワーク経由で Elastic Load Balancing にアクセスします。クライアントは以下をサポートする必要があります:

- Transport Layer Security (TLS)。TLS 1.2 が必須です。TLS 1.3 が推奨されます。
- DHE (Ephemeral Diffie-Hellman) や ECDHE (Elliptic Curve Ephemeral Diffie-Hellman) などの Perfect Forward Secrecy (PFS) を使用した暗号スイート。これらのモードは、Java 7 以降など、ほとんどの最新システムでサポートされています。

また、リクエストには、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、AWS Security Token Service (AWS STS) を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

ネットワークの隔離

仮想プライベートクラウド (VPC) は、AWS クラウド内の論理的に隔離された領域にある仮想ネットワークです。サブネットは、ある範囲の IP アドレスが示す VPC 内の領域です。ロードバランサーを作成するときに、ロードバランサーノードに 1 つ以上のサブネットを指定できます。VPC のサブネットに EC2 インスタンスをデプロイし、ロードバランサーに登録できます。VPC およびサブネットの詳細については、Amazon VPC ユーザーガイドを参照してください。

VPC 内にロードバランサーを作成する場合、インターネット向けまたは内部のいずれかを選択できます。内部向けロードバランサーは、ロードバランサー用に VPC へのアクセス権を持つクライアントからのリクエストのみをルーティングできます。

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、登録されたターゲットにリクエストを送信します。したがって、ロードバランサーからのリクエストを受信するために、ターゲットにパブリック IP アドレスは必要ありません。

プライベート IP アドレスを使用して VPC から Elastic Load Balancing API を呼び出すには、AWS PrivateLink を使用します。詳細については、「インターフェイスエンドポイントを使用して Elastic Load Balancing にアクセスする (AWS PrivateLink)」を参照してください。

ネットワークトラフィックの制御

ロードバランサーを使用する場合、ネットワークトラフィックをセキュリティで保護するには、次のオプションを検討してください。

- セキュアなリスナーを使用して、クライアントとロードバランサーの間で暗号化された通信をサポートします。Application Load Balancer は HTTPS リスナーをサポートします。Network Load Balancer は TLS リスナーをサポートします。Classic Load Balancer は、HTTPS リスナーと TLS リスナーの両方をサポートします。ロードバランサーの事前定義されたセキュリティポリシーから選択して、アプリケーションでサポートされている暗号スイートとプロトコルバージョンを指定できます。AWS Certificate Manager (ACM) または AWS Identity and Access Management (IAM) を使用して、ロードバランサーにインストールされたサーバー証明書を管理できます。Server Name Indication (SNI) プロトコルを使用して、単一のセキュアなリスナーを使用して複数の安全なウェ

ブサイトを提供できます。複数のサーバー証明書をセキュアリスナーに関連付けると、ロードバランサーで SNI が自動的に有効になります。

- 特定のクライアントからのトラフィックのみを受け入れるように、Application Load Balancer と Classic Load Balancer のセキュリティグループを設定します。これらのセキュリティグループは、リスナーポート上のクライアントからのインバウンドトラフィックと、クライアントへのアウトバウンドトラフィックを許可する必要があります。
- ロードバランサーからのトラフィックのみを受け入れるように、Amazon EC2 インスタンスのセキュリティグループを設定します。これらのセキュリティグループは、リスナーポートとヘルスチェックポートでロードバランサーからのインバウンドトラフィックを許可する必要があります。
- ID プロバイダーまたは社内認証を使用してユーザーを安全に認証するように Application Load Balancer を設定します。詳細については、Application Load Balancer を使用してユーザーを認証するを参照してください。
- Application Load Balancer で AWS WAF を使用して、ウェブアクセスコントロールリスト (ウェブ ACL) のルールに基づいてリクエストを許可またはブロックできます。

インターフェイスエンドポイントを使用して Elastic Load Balancing にアクセスする (AWS PrivateLink)

インターフェイス VPC エンドポイントを作成することで、Virtual Private Cloud (VPC) と Elastic Load Balancing API の間にプライベート接続を確立できます。この接続を使用すると、インターネットゲートウェイ、NAT インスタンス、または VPN 接続を VPC にアタッチする必要なく、VPC から Elastic Load Balancing API を呼び出すことができます。エンドポイントは、ロードバランサーの作成と管理に使用する Elastic Load Balancing API バージョン 2015-12-01 および 2012-06-01 への信頼性の高いスケーラブルな接続を提供します。

インターフェイス VPC エンドポイントは、プライベート IP アドレスを使用してアプリケーションと AWS のサービス間の通信を可能にする機能である AWS PrivateLink を利用しています。詳細については、「AWS PrivateLink」を参照してください。

制限

AWS PrivateLink は、リスナー数が 50 を超える Network Load Balancer をサポートしません。

Elastic Load Balancing のインターフェイスエンドポイントを作成する

以下のサービス名を使用して、Elastic Load Balancing のエンドポイントを作成します。

```
com.amazonaws.region.elasticloadbalancing
```

詳細については、「AWS PrivateLink ガイド」の「インターフェイスエンドポイントの作成」を参照してください。

Elastic Load Balancing 用の VPC エンドポイントポリシーを作成する

Elastic Load Balancing API へのアクセスをコントロールするために VPC エンドポイントにポリシーをアタッチすることができます。このポリシーでは以下の内容を指定します。

- アクションを実行できるプリンシパル。
- 実行可能なアクション。
- このアクションを実行できるリソース。

次の例は、エンドポイントを介してロードバランサーを作成するアクセス許可を全員に対して拒否する VPC エンドポイントポリシーを示しています。このポリシー例では、他のすべてのアクションを実行するアクセス許可も全員に付与しています。

```
{
   "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "elasticloadbalancing:CreateLoadBalancer",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

詳細については、「AWS PrivateLink ガイド」の「Control access to services using endpoint policies (エンドポイントポリシーを使用してサービスへのアクセスをコントロールする)」を参照してください。

Classic Load Balancer の移行

Elastic Load Balancing は、Application Load Balancer、Network Load Balancer、Gateway Load Balancer、Classic Load Balancer などのロードバランサーをサポートします。各ロードバランサータイプの機能の違いの詳細については、Elastic Load Balancing 製品の比較を参照してください。

VPC 内の既存の Classic Load Balancer をApplication Load Balancer またはNetwork Load Balancer に移行することもできます。

Classic Load Balancer からの移行のメリット

各タイプのロードバランサーには、それぞれ独自の特徴、機能、構成があります。各ロードバランサーの利点を確認して、どのロードバランサーが最適かを判断してください。

Application Load Balancer

クラシックロードバランサーの代わりにApplication Load Balancer を使用すると、次のようなメリットがあります。

Support 対象:

- パス条件、ホスト条件、HTTP ヘッダー条件。
- ある URL から別の URL にリクエストをリダイレクトし、1 つの EC2 インスタンス上の複数のアプリケーションにリクエストをルーティングする。
- カスタム HTTP レスポンスを返す。
- IP アドレスでターゲットを登録し、Lambda 関数をターゲットとして登録します。ロードバランサーの VPC 外のターゲットを含みます。
- 企業 ID またはソーシャル ID によるユーザーの認証。
- Amazon Elastic Container Service (Amazon ECS) のコンテナ化されたアプリケーション。
- 各サービスの状態を個別に監視します。

アクセスログには追加情報が含まれており、圧縮形式で保存されます。

ロードバランサーのパフォーマンスが全体的に向上しました。

Network Load Balancer

クラシックロードバランサーの代わりにNetwork Load Balancer を使用すると、次のようなメリットがあります。

Support 対象:

- 静的 IP アドレス。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ができます。
- ロードバランサーの VPC 外のターゲットを含む IP アドレスによるターゲットの登録。
- 1 つの EC2 インスタンス上の複数のアプリケーションにリクエストをルーティングする。
- Amazon Elastic Container Service (Amazon ECS) のコンテナ化されたアプリケーション。
- 各サービスの状態を個別に監視します。

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。

移行ウィザードを使用して移行する

移行ウィザードは、Classic Load Balancer の設定を使用して、同等のApplication Load Balancer またはNetwork Load Balancer を作成します。これにより、Classic Load Balancer の移行に必要な時間と労力が他の方法に比べて削減されます。

> **Note**
>
> ウィザードは新しいロードバランサーを作成します。ウィザードは、既存のクラシックロードバランサーをApplication Load Balancer またはNetwork Load Balancer に変換しません。新しく作成したロードバランサーにトラフィックを手動でリダイレクトする必要があります。

制限事項

- 新しいロードバランサーの名前は、同じリージョン内の同じタイプの既存のロードバランサーと同じにすることはできません。
- Classic Load Balancer `aws:` のキーにプレフィックスを含むタグがある場合、それらのタグは移行されません。

Application Load Balancer に移行する場合

- Classic Load Balancer にサブネットが 1 つしかない場合は、2 つ目のサブネットを指定する必要があります。
- Classic Load Balancer に TCP ヘルスチェックを使用する HTTP/HTTPS リスナーがある場合、ヘルスチェックプロトコルは HTTP に更新され、パスは「/」に設定されます。
- Classic Load Balancer に、カスタムまたはサポートされていないセキュリティポリシーを使用する HTTPS リスナーがある場合、移行ウィザードは新しいロードバランサータイプのデフォルトのセキュリティポリシーを使用します。

Network Load Balancer に移行する場合

- 次のインスタンスタイプは新しいターゲットグループに登録されません:C1、CC1、CC2、CG1、CG2、CR1、CS1、G1、G2、HI1、HS1、M1、M2、M3、T1
- Classic Load Balancer の特定のヘルスチェック設定は、新しいターゲットグループに転送できない場合があります。このような場合は、移行ウィザードのサマリーセクションに変更として表示されます。
- Classic Load Balancer に SSL リスナーがある場合、移行ウィザードは SSL リスナーの証明書とセキュリティポリシーを使用して TLS リスナーを作成します。

移行ウィザードのプロセス

移行ウィザードを使用してClassic Load Balancer を移行するには

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. 移行するClassic Load Balancer を選択します。
4. ロードバランサーの [詳細] セクションで、[移行ウィザードの起動] を選択します。
5. [Application Load Balancer に移行] または [Network Load Balancer に移行] を選択して、移行ウィザードを開きます。
6. [新しいロードバランサーに名前を付ける] の [ロードバランサー名] に、新しいロードバランサーの名前を入力します。
7. [新しいターゲットグループに名前を付けてターゲットを確認する] で、[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。

8. (オプション) [ターゲット] で、新しいターゲットグループに登録されるターゲットインスタンスを確認できます。
9. (オプション)「レビュータグ」では、新しいロードバランサーに適用されるタグを確認できます。
10. 「Application Load Balancer の概要」または「Network Load Balancer の概要」で、移行ウィザードによって割り当てられた構成オプションを確認して確認します。
11. 設定の概要を確認したら、[Application Load Balancer の作成] または [Network Load Balancer の作成] を選択して移行を開始します。

ロードバランサーのコピーユーティリティを使用して移行します。

ロードバランサーのコピーユーティリティは、 AWS GitHub ページの Elastic Load Balancing Tools リポジトリ内にあります。

リソース

- Elastic Load Balancing ツール
- Classic Load Balancer からApplication Load Balancer へのコピー・ ユーティリティ
- Classic Load Balancer からNetwork Load Balancer へのコピー・ ユーティリティ

ロードバランサーを手動で移行します。

以下の情報は、VPC 内の既存の Classic Load Balancer に基づいて、新しい Application Load Balancer または Network Load Balancer を手動で作成するための一般的な手順を示しています。 AWS Management Console、 AWS CLI、または AWS SDK を使用して移行できます。詳細については、「Elastic Load Balancing の使用開始」を参照してください。

移行プロセスを完了すると、新しいロードバランサーの機能を利用できます。

手動移行プロセス

ステップ 1: 新しいロードバランサーを作成する

移行する Classic Load Balancer と同等の設定でロードバランサーを作成します。

1. 新しいロードバランサーを、Classic Load Balancer と同じスキーム (インターネット向けまたは内部向け)、サブネット、セキュリティグループを設定して作成します。

2. ロードバランサーの 1 つのターゲットグループを、Classic Load Balancerと同じヘルスチェック設定で作成します。
3. 次のいずれかを行ってください。
   - Classic Load Balancer が Auto Scaling グループに接続されている場合は、ターゲットグループを Auto Scaling グループに接続します。これにより、Auto Scaling インスタンスがターゲットグループに登録されます。
   - EC2 インスタンスをターゲットグループに登録します。
4. 1 つ以上のリスナーを作成し、各リスナーに、リクエストをターゲットグループに転送するデフォルトのルールを設定します。HTTPS リスナーを作成する場合は、Classic Load Balancer 用に指定したのと同じ証明書を指定できます。デフォルトのセキュリティポリシーを使用することをお勧めします。
5. Classic Load Balancer がタグ付けされている場合は、それらを見直して、関連性のあるタグを新しいロードバランサーに追加します。

ステップ 2: トラフィックを新しいロードバランサーに段階的にリダイレクトする

インスタンスを新しいロードバランサーに登録したら、古いロードバランサーから新しいロードバランサーにトラフィックをリダイレクトするプロセスを開始できます。これにより、アプリケーションの可用性に与えるリスクを最小限に抑えながら、新しいロードバランサーをテストできます。

トラフィックを新しいロードバランサーに段階的にリダイレクトするには

1. 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、新しいロードバランサーの DNS 名を貼り付けます。すべて適切な場合は、ブラウザにアプリケーションのデフォルトページが表示されます。
2. ドメイン名を新しいロードバランサーに関連付ける新しい DNS レコードを作成します。DNS サービスが重み付けをサポートしている場合は、新しい DNS レコードに重み 1 を、古いロードバランサーの既存の DNS レコードに重み 9 を指定します。これで、トラフィックの 10% が新しいロードバランサーに、90% が古いロードバランサーにリダイレクトされます。
3. 新しいロードバランサーをモニタリングして、トラフィックが受信され、リクエストがインスタンスにルーティングされていることを確認します。

> **Important**
>
> DNS レコードの time-to-live (TTL) は 60 秒です。つまり、ドメイン名を解決する DNS サーバーは、変更が反映される間、レコード情報を 60 秒間キャッシュに保持します。

したがって、これらの DNS サーバーは、前のステップを完了してから最大 60 秒間、トラフィックを引き続き古いロードバランサーにルーティングできます。伝達の実行中、トラフィックは両方のロードバランサーにリダイレクトされる可能性があります。

4. すべてのトラフィックが新しいロードバランサーにリダイレクトされるまで、DNS レコードの重みの更新を繰り返します。完了したら、古いロードバランサーの DNS レコードを削除できます。

ステップ 3: ポリシー、スクリプト、およびコードを更新する

Classic Load Balancer を Application Load Balancer または Network Load Balancer に移行した場合は、必ず以下のことを行ってください。

- API バージョン 2012-06-01 を使用する IAM ポリシーを更新して、バージョン 2015-12-01 を使用します。
- CloudWatch 名前空間のメトリクスを使用するプロセスを、`AWS/ELB``AWS/ApplicationELB``AWS/NetworkELB`または名前空間のメトリクスを使用するように更新します。
- aws elb AWS CLI aws elbv2 AWS CLI コマンドを使用するスクリプトをコマンドを使用するように更新します。
- AWS CloudFormation `AWS::ElasticLoadBalancing::LoadBalancer`リソースを使用するテンプレートを更新してリソースを使用する。`AWS::ElasticLoadBalancingV2`
- Elastic Load Balancing API バージョン 2012-06-01 を使用するコードを更新して、バージョン 2015-12-01 を使用します。

リソース

- AWS CLI コマンドリファレンス の elbv2
- Elastic Load Balancing API リファレンスバージョン 2015-12-01
- Elastic Load Balancing の Identity and Access Management
- Application Load Balancer ユーザーガイドの Application Load Balancer metrics
- Network Load Balancer ユーザーガイドの Network Load Balancer metrics
- 「AWS CloudFormation ユーザーガイド」の「AWS::ElasticLoadBalancingV2::LoadBalancer」

ステップ 4: 古いロードバランサーを削除する

古い Classic Load Balancer は、以下を行った後に削除できます。

- すべてのトラフィックを古いロードバランサーから新しいロードバランサーにリダイレクトしました。
- 古いロードバランサーにルーティングされたすべての既存のリクエストが完了しました。

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。