

---

# Elastic Load Balancing

ユーザーガイド



## Elastic Load Balancing: ユーザーガイド

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは、顧客に混乱を招く可能性がある態様、または Amazon の信用を傷つけたり、失わせたりする態様において、Amazon のものではない製品またはサービスに関連して使用してはなりません。Amazon が所有していない他のすべての商標は、それぞれの所有者の所有物であり、Amazon と提携、接続、または後援されている場合とされていない場合があります。

## Table of Contents

Elastic Load Balancing とは?	1
ロードバランサーの利点	1
Elastic Load Balancing の特徴	1
Elastic Load Balancing へのアクセス	1
関連する のサービス	2
料金	2
Elastic Load Balancing の仕組み	3
アベイラビリティゾーンとロードバランサーノード	3
クロスゾーン負荷分散	3
ゾーンシフト	5
リクエストルーティング	6
ルーティングアルゴリズム	6
HTTP 接続	7
HTTP ヘッダー	7
HTTP ヘッダーの制限	7
ロードバランサーのスキーム	8
ネットワーク MTU	8
開始方法	10
Application Load Balancer の作成	10
Network Load Balancer を作成する	10
ゲートウェイロードバランサーを作成	10
Classic Load Balancer の作成	10
セキュリティ	11
データ保護	11
保管中の暗号化	12
転送時の暗号化	12
Identity and Access Management	12
対象者	13
アイデンティティを使用した認証	13
ポリシーを使用したアクセスの管理	16
Elastic Load Balancing で利用できる IAM 機能	17
API アクセス許可	23
サービスにリンクされたロール	25
AWS マネージドポリシー	27
コンプライアンス検証	29
耐障害性	29
インフラストラクチャセキュリティ	30
ネットワークの隔離	30
ネットワークトラフィックの制御	31
AWS PrivateLink	31
Elastic Load Balancing のインターフェイスエンドポイントを作成する	31
Elastic Load Balancing 用の VPC エンドポイントポリシーを作成する	32
Classic Load Balancer の移行	33
ステップ 1: 新しいロードバランサーを作成する	33
オプション 1: コンソールで移行ウィザードを使用する	34
オプション 2: github からのロードバランサーコピーユーティリティを使用する	35
オプション 3: Application Load Balancer または Network Load Balancer に手動で移行する	35
オプション 4: VPC の Classic Load Balancer に手動で移行する	35
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	36
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	36
ステップ 4: 古いロードバランサーを削除する	37

# 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](#) 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 を含む一連のさまざまな AWS のサービス用のコマンドを提供します。AWS CLI は、Windows、macOS、Linux 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 API を提供し、署名の計算、リクエストの再試行処理、エラー処理など、接続のさまざまな詳細を処理します。詳細については、[AWS SDK](#) をご参照ください。
- クエリ API— HTTPS リクエストを使用して呼び出す低レベル API アクションを提供します。クエリ API の使用は、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、[Linux インスタンス用の Amazon EC2 ユーザーガイド](#)または [Windows インスタンス用の 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監視し、必要に応じてアクションを実行することができます。詳細については、[Amazon CloudWatch ユーザーガイド](#)を参照してください。
- 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へ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、www.example.com を数字で構成される IP アドレス 192.0.2.1 に変換します。AWS は、ロードバランサーなどの URL をリソースに割り当てます。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 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。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ゾーンを有効にすると、Elastic Load Balancing はアベイラビリティ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ゾーンに登録したが、アベイラビリティ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。

アベイラビリティゾーンを無効にしても、そのアベイラビリティ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。

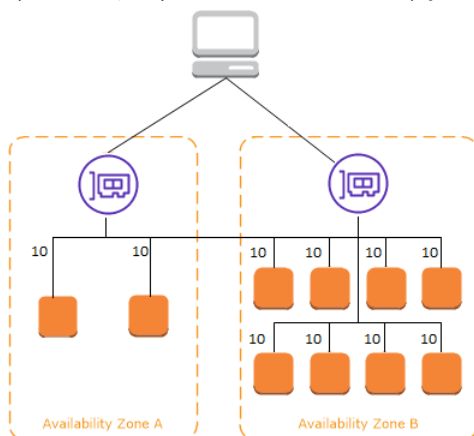
## クロスゾーン負荷分散

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

ロードバランサーノードは、そのアベイラビリティゾーン内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ゾーンが2つあり、アベイラビリティゾーン A には2つのターゲット、アベイラビリティゾーン 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。

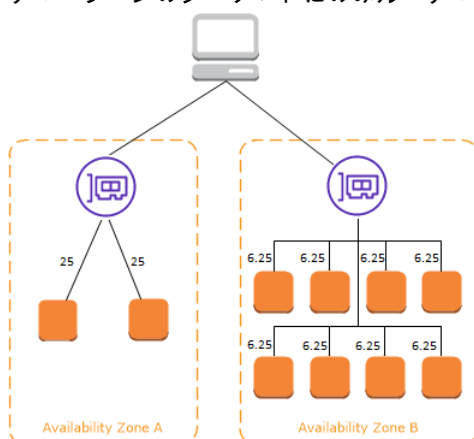
クロスゾーン負荷分散が有効な場合、10個のターゲットのそれぞれがトラフィックの10%を受け取ります。これは、各ロードバランサーノードが、そのクライアントトラフィックの50%を10個のターゲットすべてにルーティングできるためです。



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せん。

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。AWS Management Console を使用する場合、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せん。複数のアベイラビリティ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、ゾーンロードバランサーの IP アドレスを DNS から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ゾーン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 秒の有効期限 (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、「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 は事前にか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 WebSockets への接続アップグレードもサポートします。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および AWS WAF の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、プロトコー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、「[プロトコールバージョン](#)」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ーは、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。その他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。

Application Load Balancer と Classic Load Balancer が Expect ヘッダーを受信すると、コンテンツ長ヘッダーをテストせずに HTTP 100 Continue でクライアントに即座に回答し、Expect ヘッダーを削除して、リクエストをルーティングします。

## 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。Classic Load Balancerを EC2-Classic に作成するときは、インターネット向けロードバランサーにする必要があります。

2022 年 8 月 15 日に、EC2-Classic ネットワークの提供を終了します。Classic Load Balancer は、EC2-Classic ネットワークから VPC に移行することをお勧めします。詳細については、「Amazon EC2 ユーザーガイド」の「[EC2-Classic から VPC へ移行](#)」およびブログ記事「[EC2-Classic ネットワーキングがリタイア — 準備方法](#)」を参照してください。

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

ネットワーク接続の最大送信単位 (MTU) とは、接続を介して渡すことができる最大許容パケットサイズ (バイト単位) です。接続の MTU が大きいほど、より多くのデータを単一のパケットで渡すことができます。イーサネットパケットは、フレーム (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は 1500 MTU に制限されます。パケットが 1500 バイト以上ある場合は、フラグメント化されます。または、Don't Fragment フラグが IP ヘッダーに設定されている場合は削除されます。

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](#) 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、「」を参照してください [Classic Load Balancer の移行 \(p. 33\)](#)

## 目次

- [Application Load Balancer の作成 \(p. 10\)](#)
- [Network Load Balancer を作成する \(p. 10\)](#)
- [ゲートウェイロードバランサーを作成 \(p. 10\)](#)
- [Classic Load Balancer の作成 \(p. 10\)](#)

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

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

- クラウドのセキュリティ - AWS は、AWS クラウドで AWS 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負います。

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 のデータ保護 \(p. 11\)](#)
- [Elastic Load Balancing の Identity and Access Management \(p. 12\)](#)
- [Elastic Load Balancing のコンプライアンス検証 \(p. 29\)](#)
- [Elastic Load Balancing の復元性 \(p. 29\)](#)
- [Elastic Load Balancing のインフラストラクチャセキュリティ \(p. 30\)](#)
- [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 \(AWS PrivateLink\) \(p. 31\)](#)

## Elastic Load Balancing のデータ保護

AWS [責任共有モデル](#) は、Elastic Load Balancing のデータ保護に適用されます。このモデルで説明されているように、AWS は、AWS クラウド のすべてを実行するグローバルインフラストラクチャを保護する責任を負います。顧客は、このインフラストラクチャでホストされているコンテンツに対する管理を維持する責任があります。このコンテンツには、使用される AWS のサービスのセキュリティ構成と管理タスクが含まれます。データプライバシーの詳細については、「[データプライバシーのよくある質問](#)」を参照し



てください。欧州でのデータ保護の詳細については、AWS セキュリティブログに投稿された「[AWS 責任共有モデルおよび GDPR](#)」のブログ記事を参照してください。

データを保護するため、AWS アカウント の認証情報を保護し、AWS IAM Identity Center (successor to AWS Single Sign-On) または AWS Identity and Access Management (IAM) を使用して個々のユーザーをセットアップすることをお勧めします。この方法により、それぞれのジョブを遂行するために必要なアクセス許可のみを各ユーザーに付与できます。また、次の方法でデータを保護することをお勧めします。

- 各アカウントで多要素認証 (MFA) を使用します。
- SSL/TLS を使用して AWS リソースと通信します。TLS 1.2 および TLS 1.3 をお勧めします。
- AWS CloudTrail で API とユーザーアクティビティログをセットアップします。
- AWS のサービス 内でデフォルトである、すべてのセキュリティ管理に加え、AWS の暗号化ソリューションを使用します。
- Amazon Macie などの高度なマネージドセキュリティサービスを使用します。これらは、Amazon S3 に保存されている機密データの検出と保護を支援します。
- コマンドラインインターフェイスまたは API を使用して AWS にアクセスするときに FIPS 140-2 検証済みの暗号化モジュールが必要な場合は、FIPS エンドポイントを使用します。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。

顧客の E メールアドレスなどの機密情報は、タグや Name フィールドなどの自由形式のテキストフィールドに格納しないことを強くお勧めします。これは、コンソール、API、AWS CLI、または AWS SDK で Elastic Load Balancing または他の AWS のサービス を使用する場合も同様です。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 に含めないことを強くお勧めします。

## 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、一意のキーで暗号化されます。この一意のキー自体が、定期的に更新されるマスターキーで更新されます。

## 転送時の暗号化

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) or 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するために役立つ AWS のサービスです。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、追加費用なしで使用できる AWS のサービスです。

内容

- [対象者 \(p. 13\)](#)
- [アイデンティティを使用した認証 \(p. 13\)](#)
- [ポリシーを使用したアクセスの管理 \(p. 16\)](#)
- [Elastic Load Balancing で利用できる IAM 機能 \(p. 17\)](#)
- [Elastic Load Balancing API のアクセス許可 \(p. 23\)](#)
- [Elastic Load Balancing のサービスにリンクされたロール \(p. 25\)](#)
- [Elastic Load Balancing の AWS マネージドポリシー \(p. 27\)](#)

## 対象者

AWS Identity and Access Management (IAM) の用途は、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、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許可を変更する必要があります。このページの情報を確認して、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、アイデンティティ認証情報を使用して AWS にサインインする方法です。ユーザーは、AWS アカウントのルートユーザーとして、または IAM ロールを引き受けることによって、認証済み (AWS にサインイン済み) である必要があります。

ID ソースから提供された認証情報を使用して、フェデレーティッドアイデンティティとして AWS にサインインできます。AWS IAM Identity Center (successor to AWS Single Sign-On) フェデレーティッドアイデンティティの例としては、(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 の認証情報などがあります。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。フェデレーションを使用して AWS にアクセスする場合、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。AWS へのサインインの詳細については、AWS サインイン ユーザーガイドの「[AWS アカウントにサインインする方法](#)」を参照してください。

プログラムを使用して AWS にアクセスする場合、AWS は software development kit (SDK) とコマンドラインインターフェイス (CLI) を提供して、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。自分でリクエストに署名するための推奨される方法については、「AWS 全般のリファレンス」の「[署名バージョン 4 の署名プロセス](#)」を参照してください。

使用する認証方法を問わず、セキュリティ情報の提供を追加でリクエストされる場合もあります。例えば、AWS は、アカウントのセキュリティを強化するために多要素認証 (MFA) を使用することをお勧め



します。詳細については、「AWS IAM Identity Center (successor to AWS Single Sign-On) 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 での多要素認証 \(MFA\) の使用](#)」を参照してください。

## AWS アカウント ルートユーザー

AWS アカウント を作成する場合は、このアカウントのすべての AWS のサービス とリソースに対して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によってアクセスできます。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を保護し、それらを使用してルートユーザーのみが実行できるタスクを実行します。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、「AWS Account Management 全般のリファレンスガイド」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 ID

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対し、ID プロバイダーとのフェデレーションを使用して、一時的な認証情報の使用により、AWS のサービス にアクセスすることを要求します。

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か、または ID ソースから提供された認証情報を使用して AWS のサービス にアクセスするユーザーです。フェデレーティッド ID が AWS アカウント にアクセスすると、ロールが継承され、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 (successor to AWS Single Sign-On) 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か、すべての AWS アカウント とアプリケーションで使用するために、独自の ID ソースで一連のユーザーとグループに接続して同期することもできます。IAM Identity Center の詳細については、「AWS IAM Identity Center (successor to AWS Single Sign-On) 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、1 人のユーザーまたは 1 つのアプリケーションに対して特定の許可を持つ AWS アカウント 内のアイデンティティです。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、一時的な認証情報を使用することをお勧めします。ただし、IAM ユーザーと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許可を指定できます。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許可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許可を与えることができます。

ユーザーは、ロールとは異なります。ユーザーは 1 人の人または 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、特定の許可を持つ、AWS アカウント 内のアイデンティティです。これは IAM ユーザーに似ていますが、特定のユーザーには関連付けられていません。[ロールを切り替える](#)ことによって、AWS

Management Console で IAM ロールを一時的に引き受けることができます。ロールを引き受けるには、AWS CLI または AWS API オペレーションを呼び出すか、カスタム URL を使用します。ロールを使用する方法の詳細については、「IAM ユーザーガイド」の「[IAM ロールの使用](#)」を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます。

- フェデレーテッドユーザーアクセス - フェデレーテッドアイデンティティに許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーテ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許可が付与されます。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「[Creating a role for a third-party Identity Provider](#)」(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、許可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、アクセス許可セットを IAM のロールに関連付けます。アクセス許可セットの詳細については、「AWS IAM Identity Center (successor to AWS Single Sign-On) ユーザーガイド」の「[アクセス許可セット](#)」を参照してください。
- 一時的な IAM ユーザー許可 - 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許可を一時的に 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を許可する主な方法は、ロールを使用することです。ただし、一部の AWS のサービスでは、(ロールをプロキシとして使用する代わりに) リソースにポリシーを直接添付できます。クロスアカウントアクセスにおけるロールとリソーススペ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 ロールとリソーススペースのポリシーとの相違点](#)」を参照してください。
- クロスサービスアクセス - 一部の AWS のサービスでは、他の AWS のサービスの機能を使用します。例えば、サービスで呼び出しを行うと、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許可、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。
- プリンシパル許可 - IAM ユーザーまたはロールを使用して AWS でアクションを実行する場合、そのユーザーはプリンシパルと見なされます。ポリシーによって、プリンシパルに許可が付与されます。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。この場合、両方のアクションを実行するための許可が必要です。ポリシー内で、アクションに追加の依存アクションが必要かどうかを確認するには、「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](#)」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 のサービスにアクセス許可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、AWS のサービスにリンクされたサービスロールの一種です。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは、AWS アカウントに表示され、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールの許可を表示できますが、編集することはできません。
- Amazon EC2 で実行されているアプリケーション - EC2 インスタンスで実行され、AWS CLI または AWS API 要求を行っているアプリケーションの一時的な認証情報を管理するには、IAM ロールを使用できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。AWS ロールを EC2 インスタンスに割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得することができます。詳細については、「IAM ユーザーガイド」の「[Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与する](#)」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

AWS でアクセスをコントロールするには、ポリシーを作成して AWS アイデンティティまたはリソースにアタッチします。ポリシーは AWS のオブジェクトであり、アイデンティティやリソースに関連付けて、これらのアクセス許可を定義します。AWS 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) がリクエストを行うと、これらのポリシーを評価します。ポリシーでの許可により、リクエストが許可されるか拒否されるかが決まります。大半のポリシーは JSON ドキュメントとして AWS に保存されます。JSON ポリシードキュメントの構造と内容の詳細については、「IAM ユーザーガイド」の「[JSON ポリシー概要](#)」を参照してください。

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

デフォルトでは、ユーザーやロールにアクセス許可はありません。IAM 管理者は、リソースに必要なアクションを実行するためのアクセス許可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、アクションの許可を定義します。例えば、iam:GetRole アクションを許可するポリシーがあるとします。このポリシーがあるユーザーは、AWS Management Console、AWS CLI、または AWS API からロールの情報を取得できます。

### 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添付できる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件を制御します。アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「[IAM ポリシーの作成](#)」を参照してください。

アイデンティティベースのポリシーは、さらにインラインポリシーまたは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ます。マネージドポリシーは、AWS アカウント内の複数のユーザー、グループ、およびロールに添付できるスタンドアロンポリシーです。マネージドポリシーには、AWS マネージドポリシーとカスターマネージドポリシーがあります。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。ポリシーが添付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、[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、フェデレーテッドユーザー、または AWS のサービスを含めることができます。

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

### アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、またはロール) がリソースにアクセスするための許可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。

Simple Storage Service (Amazon S3)、AWS WAF、および Amazon VPC は、ACL をサポートするサービスの例です。ACL の詳細については、「Amazon Simple Storage Service デベロッパーガイド」の「[アクセスコントロールリスト \(ACL\) の概要](#)」を参照してください。

## その他のポリシータイプ

AWS では、その他の一般的ではないポリシータイプもサポートしています。これらのポリシータイプでは、より一般的なポリシータイプで付与された最大の許可を設定できます。

- **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) に付与できる許可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アクセス許可は、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、許可は無効になります。アクセス許可の境界の詳細については、「IAM ユーザーガイド」の「[IAM エンティティのアクセス許可の境界](#)」を参照してください。
- **サービスコントロールポリシー (SCP)** - SCP は、AWS Organizations で組織や組織単位 (OU) の最大許可を指定する JSON ポリシーです。AWS Organizations は、顧客のビジネスが所有する複数の AWS アカウントをグループ化し、一元的に管理するサービスです。組織内のすべての機能を有効にすると、サービス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 はメンバーアカウントのエンティティに対するアクセス許可を制限します (各 AWS アカウントのルートユーザーなど)。Organizations と SCP の詳細については、AWS Organizations ユーザーガイドの「[SCP の仕組み](#)」を参照してください。
- **セッションポリシー** - セッションポリシーは、ロールまたはフェデレーティッドユーザーの一時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許可される範囲は、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許可が派生する場合があります。これらのポリシーのいずれかを明示的に拒否した場合、許可は無効になります。詳細については、IAM ユーザーガイドの「[セッションポリシー](#)」を参照してください。

## 複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成される許可を理解するのがさらに難しくなります。複数のポリシータイプが関連するとき、リクエストを許可するかどうかを AWS が決定する方法の詳細については、「IAM ユーザーガイド」の「[ポリシーの評価ロジック](#)」を参照してください。

## 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
<a href="#">アイデンティティベースのポリシー (p. 18)</a>	はい
<a href="#">リソースベースのポリシー (p. 18)</a>	いいえ
<a href="#">ポリシーアクション (p. 18)</a>	はい
<a href="#">ポリシーリソース (p. 19)</a>	はい
<a href="#">ポリシー条件キー (サポート固有) (p. 20)</a>	はい
<a href="#">ACL (p. 21)</a>	いいえ
<a href="#">ABAC (ポリシー内のタグ) (p. 22)</a>	はい



IAM 機能	Elastic Load Balancing サポート
<a href="#">一時的な認証情報 (p. 22)</a>	はい
<a href="#">プリンシパル許可 (p. 22)</a>	はい
<a href="#">サービスロール (p. 23)</a>	いいえ
<a href="#">サービスにリンクされたロール (p. 23)</a>	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のポリシーのサポート	はい
-----------------------	----

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添付できる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件を制御します。アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「[IAM ポリシーの作成](#)」を参照してください。

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、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学ぶには、「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	いいえ
-------------------	-----

リソースベースのポリシーは、リソースにアタッチする JSON ポリシードキュメントです。リソースベースのポリシーには例として、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。ポリシーが添付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、[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、フェデレーテッドユーザー、または AWS のサービスを含めることができます。

クロスアカウントアクセスを有効にするには、全体のアカウント、または別のアカウントの IAM エンティティを、リソースベースのポリシーのプリンシパルとして指定します。リソースベースのポリシーにクロスアカウントのプリンシパルを追加しても、信頼関係は半分しか確立されない点に注意してください。プリンシパルとリソースが異なる AWS アカウントにある場合、信頼できるアカウントの IAM 管理者は、リソースにアクセスするための許可をプリンシパルエンティティ (ユーザーまたはロール) に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許可を付与します。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、ID ベースのポリシーをさらに付与する必要はありません。詳細については、IAM ユーザーガイドの「[IAM ロールとリソースベースのポリシーとの相違点](#)」を参照してください。

## 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、だれが何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの Action 要素には、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、関連する AWS API オペレーションと同じです。一致する API オペレーションのない許可のみのアクションなど、いくつかの例外があります。また、ポリシーに複数のアクションが必要なオペレーションもあります。これらの追加アクションは、依存アクションと呼ばれます。

このアクションは、関連付けられたオペレーションを実行するための許可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 (\*) を使用すると、複数のアクションを指定することができます。たとえば、Describe という単語で始まるすべてのアクションを指定するには、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 \(p. 23\)](#) 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、だれが何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Resource JSON ポリシー要素は、オブジェクトあるい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、リソースレベルの許可と呼ばれる特定のリソースタイプをサポートするアクションに対して実行できます。

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

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

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。複数のリソースを単一のステートメントで指定するには、ARN をカンマで区切ります。

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

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、だれが何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 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 要素を指定する場合、または 1 つの Condition 要素に複数のキーを指定する場合、AWS では AND 論理演算子を使用してそれらを評価します。単一の条件キーに複数の値を指定する場合、AWS では OR 論理演算子を使用して条件を評価します。ステートメントの許可が付与される前にすべての条件が満たされる必要があります。

条件を指定する際にプレースホルダー変数も使用できます。例えば IAM ユーザーに、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許可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ています。すべての AWS グローバル条件キーを確認するには、IAM ユーザーガイドの「[AWS グローバル条件コンテキストキー](#)」を参照してください。

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

## Elastic Load Balancing の ACL

ACL のサポート	いいえ
-----------	-----

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



## Elastic Load Balancing での ABAC

ABAC (ポリシー内のタグ) のサポート	はい
-----------------------	----

属性ベースのアクセスコントロール (ABAC) は、属性に基づいてアクセス許可を定義する認可戦略です。AWS では、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)、および多数の AWS リソースに添付できます。エンティティとリソースのタグ付けは、ABAC の最初のステップです。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 ポリシーをします。

ABAC は、急速に成長している環境でポリシー管理が面倒な状況に役立ちます。

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 つの条件キーすべてをサポートする場合、値は部分的です。

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

一時認証情報のサポート	はい
-------------	----

AWS のサービスには、一時認証情報を使用してサインインしても機能しないものがあります。一時的な認証情報を利用できる AWS のサービスを含めた詳細情報については、IAM ユーザーガイドの「[IAM と連携する AWS のサービス](#)」を参照してください。

ユーザー名とパスワード以外の方法で AWS Management Console にサインインする場合は、一時認証情報を使用していることとなります。例えば、会社の Single Sign-On (SSO) リンクを使用して AWS にアクセスすると、そのプロセスは自動的に一時認証情報を作成します。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。作成後、一時認証情報を使用して AWS にアクセスできるようになります。AWS は、長期的なアクセスキーを使用する代わりに、一時認証情報を動的に生成することをお勧めします。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

プリンシパル許可のサポート	はい
---------------	----

IAM ユーザーまたはロールを使用して AWS でアクションを実行するユーザーは、プリンシパルと見なされます。ポリシーは、プリンシパルにアクセス許可を付与します。一部のサービスを使用する際に、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。この場合、両方のアクションを実行するための許可が必要です。ポリシー内で、アクションに追加の依存アクション

が必要かどうかを確認するには、「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](#)」を参照してください。

## Elastic Load Balancing のサービスロール

サービスロールに対するサポート	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 のサービスにアクセス許可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

サービスにリンクされたロールのサポート	はい
---------------------	----

サービスリンクロールは、AWS のサービスにリンクされているサービスロールの一種です。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは、AWS アカウントに表示され、サービスによって所有されます。IAM 管理者は、サービスにリンクされたロールの許可を表示できますが、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 \(p. 25\)](#)」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

## 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、AWSServiceRoleForElasticLoadBalancing というサービスにリンクされたロールを使用して、お客様の代わりに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、ロールを引き受ける上で elasticloadbalancing.amazonaws.com サービスを信頼します。

## サービスにリンクされたロールの作成

AWSServiceRoleForElasticLoadBalancing 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 日以前にロードバランサーを作成した場合、Elastic Load Balancing は AWSServiceRoleForElasticLoadBalancing を AWS アカウント内に作成します。詳細については、[IAM ユーザーガイドAWSの「アカウントに新しいロールが表示される」](#)を参照してください。

## サービスにリンクされたロールを編集する

IAM を使用して、AWSServiceRoleForElasticLoadBalancing 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、AWSServiceRoleForElasticLoadBalancing を削除することをお勧めします。

このサービスにリンクされたロールを削除するには、AWS アカウントのロードバランサーをすべて削除する必要があります。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。

AWSServiceRoleForElasticLoadBalancing を削除した後、ロードバランサーを作成すると、Elastic Load Balancing によって再度このロールが作成されます。

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

ユーザー、グループ、ロールに許可を追加するには、自分でポリシーを作成するよりも、AWS マネージドポリシーを使用する方が簡単です。チームに必要な許可のみを提供する [IAM カスタマーマネージドポリシーを作成する](#) には、時間と専門知識が必要です。すぐに使用を開始するために、AWS マネージドポリシーを使用できます。これらのポリシーは、一般的なユースケースをターゲット範囲に含めており、AWS アカウントで利用できます。AWS マネージドポリシーの詳細については、「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 のサービスは、AWS マネージドポリシーを維持および更新します。AWS マネージドポリシーの許可を変更することはできません。サービスは、新しい機能をサポートするために、AWS マネージドポリシーに対する許可を追加することができます。この種類の更新は、ポリシーがアタッチされている、すべてのアイデンティティ (ユーザー、グループおよびロール) に影響を与えます。新しい機能が立ち上げられた場合や、新しいオペレーションが使用可能になった場合に、各サービスが AWS マネージドポリシーを更新する可能性が最も高くなります。サービスは、AWS マネージドポリシーから許可を削除しないため、ポリシーの更新によって既存の許可が破棄されることはありません。

さらに、AWS は、複数のサービスにまたがるジョブ機能の特徴に対するマネージドポリシーもサポートしています。例えば、ReadOnlyAccess AWS マネージドポリシーでは、すべての AWS のサービスおよびリソースへの読み取り専用アクセスを許可します。サービスが新しい機能を起動する場合、AWS は、新たなオペレーションとリソース用に、読み取り専用の許可を追加します。職務機能ポリシーのリストとそれらの説明については、IAM ユーザーガイドの「[職務機能の AWS 管理ポリシー](#)」を参照してください。

### AWS 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 AWS のその他サービス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。

このポリシーに対する許可を確認するには、AWS マネジメントコンソールで「[AWSElasticLoadBalancingClassicServiceRolePolicy](#)」を参照してください。

### 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。

このポリシーに対する許可を確認するには、AWS マネジメントコンソールで「[AWSElasticLoadBalancingServiceRolePolicy](#)」を参照してください。

### AWS マネージドポリシー: ElasticLoadBalancingFullAccess

このポリシーは、Elastic Load Balancing サービスに対する完全なアクセス権と、AWS マネジメントコンソール経由でのその他サービスへの限定的なアクセス権を付与します。

このポリシーに対する許可を確認するには、AWS マネジメントコンソールで「[ElasticLoadBalancingFullAccess](#)」を参照してください。



## 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。

このポリシーに対する許可を確認するには、AWS マネジメントコンソールで「[ElasticLoadBalancingReadOnly](#)」を参照してください。

## AWS マネージドポリシーへの Elastic Load Balancing 更新

Elastic Load Balancing がこのサービス向けの AWS マネージドポリシーに対する更新の追跡を開始した日付以降の変更にに関する詳細を確認します。

変更	説明	日付
<a href="#">AWS マネージドポリシー: ElasticLoadBalancingFullAccess</a> (p. 27) – 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
<a href="#">AWS マネージドポリシー: ElasticLoadBalancingReadOnly</a> (p. 27) – 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
<a href="#">AWS マネージドポリシー: AWSElasticLoadBalancingServiceReadOnly</a> (p. 27) – 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
<a href="#">AWS マネージドポリシー: ElasticLoadBalancingFullAccess</a> (p. 27) – 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 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。	2021 年 10 月 11 日
<a href="#">AWS マネージドポリシー: AWSElasticLoadBalancingClassicServiceReadOnly</a> (p. 27) – 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
<a href="#">AWS マネージドポリシー: ElasticLoadBalancingReadOnly</a> (p. 27) – 既存ポリシーへの更新。	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

変更	説明	日付
Elastic Load Balancing が変更の追跡を開始	Elastic Load Balancing がその AWS マネージドポリシーに対する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対象であるかどうかを確認するには、「[コンプライアンスプログラムによる対象範囲内の AWS のサービスのサービス](#)」をご覧ください。関心のあるコンプライアンスプログラムを選択してください。一般的な情報については、[AWS コンプライアンスプログラム](#)を参照してください。

AWS Artifact を使用して、サードパーティーの監査レポートをダウンロードできます。詳細については、「[AWS Artifact におけるダウンロードレポート](#)」を参照してください。

AWS のサービスを使用する際のユーザーのコンプライアンス責任は、ユーザーのデータの機密性や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。AWS では、コンプライアンスに役立つ次のリソースを提供しています。

- [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を AWS にデプロイするためのステップを示します。
- 「[アマゾン ウェブ サービスでの HIPAA のセキュリティとコンプライアンスのためのアーキテクチャ](#)」 – このホワイトペーパーは、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を説明しています。

### Note

すべての AWS のサービスが HIPAA 適格であ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 コンプライアンスのリソース](#) – このワークブックおよびガイドのコレクションは、顧客の業界と拠点に適用されるものである場合があります。
- AWS Config デベロッパーガイドの[ルールでのリソースの評価](#) – AWS Config サービスでは、自社のプラクティス、業界ガイドライン、および規制に対するリソースの設定の準拠状態を評価します。
- [AWS Security Hub](#) – この AWS のサービスは、AWS 内のセキュリティ状態の包括的なビューを提供します。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。
- [AWS Audit Manager](#) – この AWS のサービスは AWS の使用状況を継続的に監査し、リスクの管理方法やコンプライアンスを業界スタンダードへの準拠を簡素化するために役立ちます。

## 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ゾーンがあります。アベイラビリティゾーンでは、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 セキュリティサービスと AWS がインフラストラクチャを保護する方法については、「[AWS クラウドセキュリティ](#)」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、「セキュリティの柱 - AWS Well-Architected Framework」の「[インフラストラクチャ保護](#)」を参照してください。

AWS が公開している 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 および TLS 1.3 をお勧めします。
- 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です。これらのモードは、Java 7 以降など、最近のほとんどのシステムでサポートされています。

また、リクエストは、アクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、[AWS Security Token Service](#) (AWS STS) を使用して、一時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 \(AWS PrivateLink\) \(p. 31\)](#)」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 Guide] (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) を参照してください。

# Classic Load Balancer の移行

EC2-Classic ネットワークを廃止します。Classic Load Balancer は、EC2-Classic ネットワークから VPC に移行することをお勧めします。詳しくは、「Amazon EC2 ユーザーガイド」の「[EC2-Classic から VPC へ移行](#)」およびブログ記事「[EC2-Classic ネットワーキングの廃止 — 準備方法](#)」を参照してください。

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。

EC2-Classic ネットワークに Classic Load Balancer がある場合は、VPC に移行する必要があります。AWS は 2022 年 8 月 15 日をもって EC2-Classic ネットワークを終了する予定です。この日付以降は、Classic Load Balancer は VPC でのみ使用できます。中断を防ぐために、2022 年 8 月 15 日までに EC2-Classic にある Classic Load Balancer を VPC に移行する必要があります。詳細については、「[VPC の Classic Load Balancer に移行する \(p. 35\)](#)」を参照してください。

また、VPC にある Classic Load Balancer を、Application Load Balancer または Network Load Balancer に置き換えることもできます。詳細については、「[Application Load Balancer または Network Load Balancer に移行する \(p. 33\)](#)」を参照してください。

## 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 \(p. 33\)](#)
- [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 \(p. 36\)](#)
- [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 \(p. 36\)](#)
- [ステップ 4: 古いロードバランサーを削除する \(p. 37\)](#)

## 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。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。

Application Load Balancer または Network Load Balancer を作成して VPC 内の Classic Load Balancer を置き換えるには、次のいずれかのオプションを使用します。

- [オプション 1: コンソールで移行ウィザードを使用する \(p. 34\)](#)
- [オプション 2: github からのロードバランサーコピーユーティリティを使用する \(p. 35\)](#)
- [オプション 3: Application Load Balancer または Network Load Balancer に手動で移行する \(p. 35\)](#)

VPC で Classic Load Balancer を作成して EC2-Classic の Classic Load Balancer を置き換えるには、次のオプションを使用します。

- [オプション 4: VPC の Classic Load Balancer に手動で移行する \(p. 35\)](#)

## オプション 1: コンソールで移行ウィザードを使用する

移行ウィザードは、VPC 内の Classic Load Balancer の設定に基づいて、Application Load Balancer または Network Load Balancer を作成します。作成されるロードバランサーのタイプは、Classic Load Balancer の設定によって異なります。

### 移行ウィザードのリリースノート

- Classic Load Balancer は VPC 内に存在する必要があります。
- Classic Load Balancer に HTTP または HTTPS リスナーがある場合、ウィザードは Application Load Balancer を作成します。Classic Load Balancer に TCP リスナーがある場合、ウィザードは Network Load Balancer を作成します。
- Classic Load Balancer の名前が既存の Application Load Balancer または Network Load Balancer の名前と一致する場合は、ウィザードで移行時に別の名前を指定する必要があります。
- Classic Load Balancer に 1 つのサブネットがある場合、ウィザードでは、Application Load Balancer の作成時に 2 番目のサブネットを指定する必要があります。
- Classic Load Balancer で EC2-Classic に登録されたインスタンスがある場合、新しいロードバランサーのターゲットグループには登録されません。
- Classic Load Balancer に次のタイプの登録されたインスタンスがある場合は、Network Load Balancer のターゲットグループには登録されません:  
C1、CC1、CC2、CG1、CG2、CR1、CS1、G1、G2、HI1、HS1、M1、M2、M3、および T1。
- Classic Load Balancer に HTTP/HTTPS リスナーがあるが、TCP ヘルスチェックを使用する場合、ウィザードはそれらを HTTP ヘルスチェックに変更します。次に、Application Load Balancer の作成時にデフォルトでパスが「/」に設定されます。
- Classic Load Balancer が Network Load Balancer に移行されると、Network Load Balancer の要件を満たすようにヘルスチェックの設定が変更されます。
- Classic Load Balancer に複数の HTTPS リスナーがある場合、ウィザードはその 1 つを選択し、証明書とポリシーを使用します。ポート 443 に HTTPS リスナーがある場合、ウィザードはこのリスナーを選択します。選択されたリスナーでカスタムポリシーまたは Application Load Balancer でサポートされていないポリシーを使用する場合、ウィザードはデフォルトのセキュリティポリシーに変更されます。
- Classic Load Balancer にセキュアな TCP リスナーがある場合、Network Load Balancer は TCP リスナーを使用します。ただし、証明書またはセキュリティポリシーは使用しません。
- Classic Load Balancer に複数のリスナーがある場合、ウィザードは最も低い値を持つリスナーポートを、ターゲットグループポートとして使用します。これらのリスナーに登録された各インスタンスは、すべてのリスナーのリスナーポートでターゲットグループに登録されます。
- Classic Load Balancer で、タグ名に aws というプレフィックスの付いたタグがある場合、そのタグは新しいロードバランサーに追加されません。

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 [LOAD BALANCING] で [ロードバランサー] を選択します。
3. Classic Load Balancer を選択します。
4. [Migration] タブで、[Launch ALB Migration Wizard] または [Launch NLB Migration Wizard] をクリックします。表示されるボタンは、Classic Load Balancer を調べた後でウィザードによって選択されたロードバランサータイプによって異なります。
5. [Review] ページで、ウィザードによって選択された設定オプションを確認します。オプションを変更するには、[Edit] を選択します。

6. 新しいロードバランサーの設定が終了したら、[Create] を選択します。

## オプション 2: github からのロードバランサーコピーユーティリティを使用する

このロードバランサーコピーユーティリティは GitHub にあります。詳細については、[Elastic Load Balancing のツール](#)を参照してください。

## オプション 3: Application Load Balancer または Network Load Balancer に手動で移行する

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 を使用して移行を実行できます。詳細については、「」を参照してください[Elastic Load Balancing の使用開始 \(p. 10\)](#)

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。

## オプション 4: VPC の Classic Load Balancer に手動で移行する

以下の情報は、EC2-Classic の Classic Load Balancer に基づいて、VPC で新しい Classic 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 を使用して移行を実行できます。詳細については、Classic Load Balancer ユーザーガイドの[Tutorial: Create a Classic Load Balancer](#)を参照してください。

1. EC2 リソース (インスタンスやセキュリティグループなど) を EC2-Classic から VPC に移行します。詳細については、Amazon EC2 ユーザーガイドの[Migrate your resources to a VPC](#)を参照してください。
2. VPC で新しい Classic Load Balancer を作成します。
3. ロードバランサーを作成する際は、EC2-Classic からインスタンスを移行したときに使用した VPC を選択します。新しいロードバランサーに登録するインスタンスを含む各アベイラビリティゾーンから 1 つのサブネットを選択します。
4. プロンプトが表示されたら、ロードバランサーに登録するインスタンスを選択します。
5. 古い Classic Load Balancer にタグがある場合は、タグを確認し、関連するタグを新しい Classic Load Balancer 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 レコードの有効期限 (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。
- AWS/ELB 名前空間の CloudWatch メトリクスを使用するプロセスを更新して、AWS/ApplicationELB または AWS/NetworkELB 名前空間のメトリクスを使用します。
- `aws elb` AWS CLI コマンドを使用するスクリプトを更新して、`aws elbv2` AWS CLI コマンドを使用します。
- `AWS::ElasticLoadBalancing::LoadBalancer` リソースを使用する AWS CloudFormation テンプレートを更新して、`AWS::ElasticLoadBalancingV2` リソースを使用します。
- 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 \(p. 12\)](#)
- 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。