

---

# Elastic Load Balancing

ユーザーガイド



## Elastic Load Balancing: ユーザーガイド

Copyright © 2019 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## Table of Contents

Elastic Load Balancing とは .....	1
ロードバランサーの利点 .....	1
Elastic Load Balancing の機能 .....	1
Elastic Load Balancing へのアクセス .....	1
関連サービス .....	2
料金 .....	2
Elastic Load Balancing の詳細 .....	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
リクエストルーティング .....	4
ルーティングアルゴリズム .....	5
HTTP 接続 .....	5
HTTP ヘッダー .....	6
HTTP ヘッダーの制限 .....	6
ロードバランサーのスキーム .....	6
開始方法 .....	8
Application Load Balancer を作成する .....	8
ネットワークロードバランサー の作成 .....	8
Classic Load Balancer の作成 .....	8
認証とアクセスコントロール .....	9
IAM ポリシーを使用したアクセス権限の付与 .....	9
Elastic Load Balancing の API アクション .....	10
Elastic Load Balancing リソース .....	10
Elastic Load Balancing のリソースレベル権限 .....	12
Elastic Load Balancing の条件キー .....	13
事前定義の AWS 管理ポリシー .....	15
API アクセス許可 .....	15
2015-12-01 API に必要なアクセス許可 .....	15
2012-06-01 API に必要なアクセス許可 .....	16
サービスにリンクされたロール .....	17
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 .....	17
サービスにリンクされたロールを作成する .....	18
サービスにリンクされたロールを編集する .....	18
サービスにリンクされたロールを削除する .....	18
移行 .....	19
ステップ 1: 新しいロードバランサーを作成する .....	19
オプション 1: 移行ウィザードを使用した移行 .....	19
オプション 2: ロードバランサーコピーユーティリティを使用した移行 .....	20
オプション 3: 手動の移行 .....	20
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 .....	21
ステップ 3: Classic Load Balancer への参照を更新する .....	21
ステップ 4: Classic Load Balancer を削除する .....	22

# Elastic Load Balancing とは

Elastic Load Balancing は受信したアプリケーションまたはネットワークトラフィックを、Amazon EC2 インスタンス、コンテナ、IP アドレス、複数のアベイラビリティゾーンなど、複数のターゲットに分散させます。Elastic Load Balancing はアプリケーションへのトラフィックが時間の経過とともに変化するのに応じてロードバランサーをスケーリングし、大半のワークロードに合わせて自動的にスケーリングできます。

## 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。

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ために使用されるヘルスチェックを設定することで、ロードバランサーは正常なもののみリクエストを送信できます。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 の機能

Elastic Load Balancing は、Application Load Balancer、Network Load Balancer、および クラシックロードバランサー の 3 種類のロードバランサーをサポートしています。アプリケーションのニーズに合わせて、ロードバランサーを選択できます。詳細については、「[Elastic Load Balancing 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の詳細については、[Application Load Balancer 用ユーザーガイド](#)、[Network Load Balancer 用ユーザーガイド](#)、および [クラシックロードバランサー 用ユーザーガイド](#)を参照してください。

## 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 マネジメントコンソール — Elastic Load Balancing へのアクセスに使用するウェブインターフェイスを提供します。
- AWS コマンドラインインターフェース (AWS CLI) — Elastic Load Balancing を含むさまざまな AWS サービス用のコマンドが用意されており、Windows、Mac、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 SDK — 言語固有の API を提供し、署名の計算、リクエストの再試行処理、エラー処理など、接続のさまざまな詳細を処理します。詳細については、AWS SDK を参照してください。
- 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 の最も直接的なアクセス方法ですが、リクエストに署名するハッシュの生成やエラー処理など、低レベルの詳細な作業をアプリケーションで処理する必要があります。詳細については、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](#)
- クラシックロードバランサー — [API バージョン 2012-06-01](#)

## 関連サービス

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 Amazon EC2 ユーザーガイド](#) または [Windows インスタンスの 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インスタンスの実行数を保証し、需要の変化に応じて自動的にインスタンス数を増減できるようにします。Elastic Load Balancing を使用して Auto Scaling を有効にした場合、Auto Scaling によって起動されたインスタンスは自動的にロードバランサーに登録され、Auto Scaling によって終了されたインスタンスは自動的にロードバランサーから登録解除されます。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せん。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せん。
- [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 Developer Guide](#)」を参照してください。
- [Route 53] — ドメイン名 (www.example.com など) 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 (192.0.2.1 など) 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。ロードバランサーなどのリソースには、AWS により URL が割り当てられます。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 開発者ガイド](#)」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 開発者ガイド](#)」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 料金表](#)」を参照してください。

# Elastic Load Balancing の詳細

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、不具合のある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停止し、ターゲットが正常な状態に戻ったことを検出すると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。リスナーの設定には、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、および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ます。

Elastic Load Balancing は、Application Load Balancer、Network Load Balancer、およびクラシックロードバランサーの 3 種類のロードバランサーをサポートしています。これらのロードバランサーの設定方法には重要な相違点があります。Application Load Balancer および Network Load Balancer 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。クラシックロードバランサーでは、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る場合、登録したターゲットはトラフィックを受信しません。ロードバランサーが最も効果的なのは、有効な各アベイラビリティゾーンに少なくとも 1 つの登録済みターゲットがある場合です。

マルチ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。(Application Load Balancer では、マルチアベイラビリティゾーンを有効にする必要があります。) この設定では、1 つのアベイラビリティゾーンが利用できなくなった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トラフィックを別のアベイラビリティゾーンの正常な登録済みターゲットにルーティングできます。

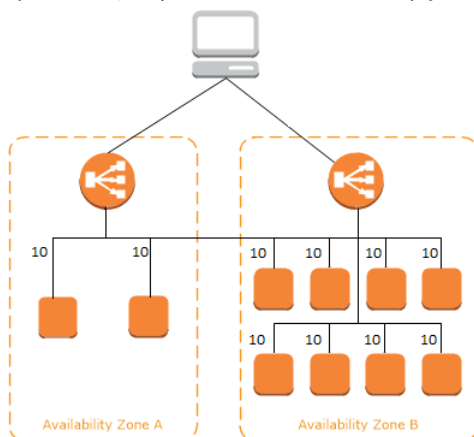
アベイラビリティゾーンを無効にすると、そのアベイラビリティゾーン内のターゲットはロードバランサーに登録されたままですが、ロードバランサーはトラフィックをターゲットにルーティングしなくなります。

## クロスゾーン負荷分散

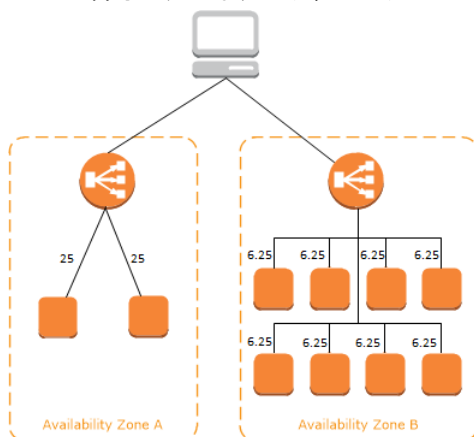
ロードバランサーのノードは、クライアントからのリクエストを登録済みターゲットに分散させます。クロスゾーン負荷分散が有効な場合、各ロードバランサーノードは、有効なすべてのアベイラビリティゾーンの登録済みターゲットにトラフィックを分散します。クロスゾーン負荷分散が無効な場合、各ロードバランサーノードは、そのアベイラビリティゾーンの登録済みターゲットにのみトラフィックを分散します。

次の図はクロスゾーン負荷分散の効果を示しています。有効なアベイラビリティゾーンが 2 つがあり、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。これにより、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 つのターゲットそれぞれがトラフィックの 25% を受け取り、アベイラビリティゾーン B の 8 つのターゲットそれぞれがトラフィックの 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常に有効になっています。

Network Load Balancer 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。ネットワーク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。詳細については、Network Load Balancer 用ユーザーガイドの「[クロスゾーン負荷分散](#)」を参照してください。

Classic Load Balancer を作成する際に、クロスゾーン負荷分散のデフォルト設定は、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 マネジメントコンソールを使用する場合、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。Classic Load Balancer 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。詳細については、クラシックロードバランサー 用ユーザーガイドの「[クロスゾーン負荷分散の有効化](#)」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

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 サーバーは、1 つ以上の IP アドレス (ロードバランサー用のロードバランサーノードの IP アドレス) をクライアントに返します。Network Load Balancer では、Elastic Load Balancing は有効にした各アベイラビリティゾーンにネットワークインターフェイスを作成します。アベイラビリティゾーンの各ロードバランサーノードは、このネットワークインターフェイスを使用して静的 IP アドレスを取得します。ロードバランサーの作成時に、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 エントリでは、有効期限 (TTL) も 60 秒に指定されているため、トラフィックの変化に応じて IP アドレスが迅速に再マップされる点に注意してください。

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。

## ルーティングアルゴリズム

Application Load Balancer を使用すると、リクエストを受信したロードバランサーノードは、優先度順にリスナールールを評価して適用するルールを決定し、ラウンドロビン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。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、接続を受信したロードバランサーノードは、プロトコル、送信元 IP アドレス、送信元ポート、宛先 IP アドレス、宛先ポート、および TCP シーケンス番号に基づいて、フローハッシュアルゴリズムを使用して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。クライアントからの TCP 接続のソースポートとシーケンス番号は異なり、別のターゲットにルーティングできます。各 TCP 接続は、接続中は単一のターゲットにルーティングされます。

クラシックロードバランサー では、リクエストを受信したロードバランサーノードは、TCP リスナーにはラウンドロビンルーティングアルゴリズムを使用して、HTTP および HTTPS リスナーには最小の未処理リクエストルーティングアルゴリズムを使用して、登録されたインスタンスを選択します。

## HTTP 接続

クラシックロードバランサー は事前にかかれた接続を使用しますが、Application Load Balancer はこの接続を使用しません。クラシックロードバランサー と Application Load Balancer の両方で、接続の多重化が使用されます。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 キープアライブを無効にします。

クラシックロードバランサー 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) の次のプロトコルをサポートします: HTTP/0.9、HTTP/1.0、HTTP/1.1。

Application Load Balancer はフロントエンド接続の次のプロトコルをサポートします: HTTP/0.9、HTTP/1.0、HTTP/1.1、HTTP/2。HTTP/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します。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。

Application Load Balancer および クラシックロードバランサー はどちらもバックエンド接続で HTTP/1.1 を使用します (登録されたターゲットへのロードバランサー)。キープアライブによりバックエンド接続はデフォルトでサポートされます。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。Application Load Balancer においては、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 においては、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。

Application Load Balancer と クラシックロードバランサー の両方でアイドルタイムアウト値を設定できます。デフォルト値は 60 秒です。Application Load Balancer では、アイドルタイムアウト値はフロントエンド接続のみに適用されます。Classic Load Balancer では、フロントエンド接続とバックエンド接続がアイドルタイムアウト値を超えたアイドル状態になった場合には、接続が切断されてクライアントはエラーレスポンスを受け取ります。登録されたターゲットは、切断できる状態になるまでキープアライブタイムアウトを使用してバックエンド接続を確保できます。

Application Load Balancer および クラシックロードバランサー 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。

## HTTP ヘッダー

Application Load Balancer と クラシックロードバランサー は [X-Forwarded-For]、[X-Forwarded-Proto]、[X-Forwarded-Port] ヘッダーをサポートします。

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 クラシックロードバランサー は、クライアントに返信する応答のプロキシの後のクライアントの入力リクエストからの接続ヘッダーを優先します。

## HTTP ヘッダーの制限

Application Load Balancer の HTTP/1.x ヘッダーには次のようなサイズ制限があります。

- リクエスト行: 16K
- 単一ヘッダー: 16K
- 総ヘッダー: 64K

Application Load Balancer の HTTP/2 ヘッダーには次のようなサイズ制限があります。

- リクエスト行: 8K
- 単一ヘッダー: 8K
- 総ヘッダー: 64K

## 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。

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レイヤーがある場合 (インターネットに接続する必要があるウェブサーバーや、ウェブサーバーにのみ接続されているデータベースサーバーなど)、内部ロードバランサーとインターネット接続ロードバランサーの両方を使用するアーキテクチャーを設計できます。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、そこにデータベース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、データベース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。データベースサーバーは、内部ロードバランサーからリクエストを受け取ります。

# Elastic Load Balancing の開始方法

ロードバランサーには、Application Load Balancer、Network Load Balancer、およびクラシックロードバランサーの3種類があります。アプリケーションのニーズに合わせて、ロードバランサーを選択できます。詳細については、「[Elastic Load Balancing 製品の比較](#)」を参照してください。

すでに Classic Load Balancer がある場合は、Application Load Balancer または ネットワークロードバランサー に移行できます。詳細については、「[Classic Load Balancer の移行 \(p. 19\)](#)」を参照してください。

## Application Load Balancer を作成する

AWS マネジメントコンソール を使用して Application Load Balancer を作成するには、Application Load Balancer 用ユーザーガイドの「[Application Load Balancer の開始方法](#)」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 用ユーザーガイドの「[AWS CLI を使用した Application Load Balancer の作成](#)」を参照してください。

## ネットワークロードバランサー の作成

AWS マネジメントコンソール を使用して ネットワークロードバランサー を作成するには、Network Load Balancer 用ユーザーガイドの「[Network Load Balancer の開始方法](#)」を参照してください。

AWS CLI を使用して ネットワークロードバランサー を作成するには、Network Load Balancer 用ユーザーガイドの「[AWS CLI を使用した ネットワークロードバランサー の作成](#)」を参照してください。

## Classic Load Balancer の作成

AWS マネジメントコンソール を使用して Classic Load Balancer を作成するには、クラシックロードバランサー 用ユーザーガイドの「[Classic Load Balancer の作成](#)」を参照してください。

# ロードバランサーの認証とアクセス コントロール

AWS ではセキュリティ認証情報を使用して、ユーザーを識別し、AWS リソースへのアクセスを付与します。AWS Identity and Access Management (IAM) の機能を使用して、他のユーザー、サービス、およびアプリケーションが完全にまたは制限付きでお客様の AWS リソースを使用できるようにします。その際、お客様のセキュリティ認証情報は共有されません。

デフォルトでは、IAM ユーザーには、AWS リソースを作成、表示、変更するためのアクセス権限はありません。IAM ユーザーがロードバランサーなどのリソースにアクセスし、タスクを実行できるようにするには、特定のリソースや必要となる API アクションを使用するためのアクセス許可を IAM ユーザーに付与する IAM ポリシーを作成してから、そのポリシーを IAM ユーザーまたは IAM ユーザーが属するグループにアタッチする必要があります。ポリシーをユーザーまたはユーザーのグループにアタッチする場合、ポリシーによって特定リソースの特定タスクを実行するユーザーの権限が許可または拒否されます。

たとえば、IAM を使用して AWS アカウントにユーザーとグループを作成することができます (IAM ユーザーは、人、システム、またはアプリケーションのいずれかです)。その後、ユーザーとグループにアクセス権限を付与すると、IAM ポリシーを使用して指定したリソースに対する特定のアクションを実行できます。

## IAM ポリシーを使用したアクセス権限の付与

ポリシーをユーザーまたはユーザーのグループにアタッチする場合、ポリシーによって特定リソースの特定タスクを実行するユーザーの権限が許可または拒否されます。

IAM ポリシーは 1 つ以上のステートメントで構成される JSON ドキュメントです。各ステートメントは次のように構成されます。

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "effect",
    "Action": "action",
    "Resource": "resource-arn",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  }]
}
```

- [Effect (効果)] — effect は、Allow または Deny にすることができます。デフォルトでは、IAM ユーザーはリソースおよび API アクションを使用するアクセス許可がないため、リクエストはすべて拒否されます。明示的な許可はデフォルトに優先します。明示的な拒否はすべての許可に優先します。
- [Action (アクション)] — action は、アクセス許可を付与または拒否する対象とする、特定の API アクションです。アクション条件を指定する方法については、「[Elastic Load Balancing の API アクション \(p. 10\)](#)」を参照してください。
- [Resource (リソース)] — アクションによって影響を及ぼされるリソースです。多くの Elastic Load Balancing API アクションがある場合、このステートメントで Amazon リソースネーム (ARN) を指定す

ることで、特定のロードバランサーに許可または拒否される権限を制限できます。それ以外の場合、すべてのロードバランサーを指定するにはワイルドカード (\*) を使用できます。詳細については、「[Elastic Load Balancing リソース \(p. 10\)](#)」を参照してください。

- [Condition (条件)] — ポリシーが有効になるタイミングを制御する条件を必要に応じて使用できます。詳細については、「[Elastic Load Balancing の条件キー \(p. 13\)](#)」を参照してください。

詳細については、『[IAM ユーザーガイド](#)』を参照してください。

## Elastic Load Balancing の API アクション

IAM ポリシーステートメントの [Action (アクション)] 要素では、Elastic Load Balancing に用意された API アクションを指定できます。次の例に示すように、アクション名の前に小文字の文字列 `elasticloadbalancing:` を指定する必要があります。

```
"Action": "elasticloadbalancing:DescribeLoadBalancers"
```

1 つのステートメントで複数のアクションを指定するには、次のようにアクションをカンマで区切って全体を角括弧で囲みます。

```
"Action": [  
  "elasticloadbalancing:DescribeLoadBalancers",  
  "elasticloadbalancing>DeleteLoadBalancer"  
]
```

ワイルドカード (\*) を使用して複数のアクションを指定することもできます。次の例では、Describe で始まる Elastic Load Balancing のすべての API アクション名を指定します。

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

Elastic Load Balancing のすべての API アクションを指定するには、次の例のように、ワイルドカード (\*) を使用します。

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

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 および Network Load Balancer — [API リファレンスバージョン 2015-12-01](#)
- クラシックロードバランサー — [API リファレンスバージョン 2012-06-01](#)

## Elastic Load Balancing リソース

リソースレベルのアクセス許可とは、ユーザーがアクションを実行可能なリソースを指定できることを意味します。Elastic Load Balancing では、リソースレベルのアクセス許可が部分的にサポートされます。リソースレベルのアクセス許可をサポートする API アクションの場合、そのアクションでユーザーが使用できるリソースを制御できます。ポリシーステートメント内でリソースを指定するには、Amazon リソースネーム (ARN) を使用する必要があります。ARN の指定時、正確なロードバランサー名を指定したくない場合などに、パスに \* ワイルドカードを使用できます。

Application Load Balancer の ARN の形式を次に示します。

```
arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/app/load-balancer-name/load-balancer-id
```

ネットワークロードバランサー の ARN の形式を次に示します。

```
arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/net/load-balancer-name/load-balancer-id
```

Classic Load Balancer の ARN の形式を次に示します。

```
arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/load-balancer-name
```

リスナーおよび Application Load Balancer のリスナーールの ARN の形式を次に示します。

```
arn:aws:elasticloadbalancing:region-code:account-id:listener/app/load-balancer-name/load-balancer-id/listener-id  
arn:aws:elasticloadbalancing:region-code:account-id:listener-rule/app/load-balancer-name/load-balancer-id/listener-id/rule-id
```

ネットワークロードバランサー のリスナーの ARN の形式を次に示します。

```
arn:aws:elasticloadbalancing:region-code:account-id:listener/net/load-balancer-name/load-balancer-id/listener-id
```

ターゲットグループの ARN の形式を以下に示します。

```
arn:aws:elasticloadbalancing:region-code:account-id:targetgroup/target-group-name/target-group-id
```

リソースレベルの権限をサポートしない API アクション

次の Elastic Load Balancing アクションは、リソースレベルのアクセス許可をサポートしていません。

- API バージョン 2015-12-01:
  - DescribeAccountLimits
  - DescribeListenerCertificates
  - DescribeListeners
  - DescribeLoadBalancerAttributes
  - DescribeLoadBalancers
  - DescribeRules
  - DescribeSSLPolicies
  - DescribeTags
  - DescribeTargetGroupAttributes
  - DescribeTargetGroups
  - DescribeTargetHealth
- API バージョン 2012-06-01:
  - DescribeInstanceHealth
  - DescribeLoadBalancerAttributes
  - DescribeLoadBalancerPolicyTypes
  - DescribeLoadBalancers
  - DescribeLoadBalancerPolicies

- DescribeTags

リソースレベルの権限をサポートしていない API アクションの場合、次の Resource ステートメントを指定する必要があります。

```
"Resource": "*"

```

## Elastic Load Balancing のリソースレベル権限

次の表では、リソースレベルのアクセス許可をサポートしている Elastic Load Balancing アクションと、各アクションでサポートされるリソースについて説明しています。

API バージョン 2015-12-01

API アクション	リソース ARN
AddListenerCertificates	リスナー
AddTags	ロードバランサー、ターゲットグループ
CreateListener	ロードバランサー
CreateLoadBalancer	ロードバランサー
CreateRule	リスナー
CreateTargetGroup	ターゲットグループ
DeleteListener	リスナー
DeleteLoadBalancer	ロードバランサー
DeleteRule	リスナールール
DeleteTargetGroup	ターゲットグループ
DeregisterTargets	ターゲットグループ
ModifyListener	リスナー
ModifyLoadBalancerAttributes	ロードバランサー
ModifyRule	リスナールール
ModifyTargetGroup	ターゲットグループ
ModifyTargetGroupAttributes	ターゲットグループ
RegisterTargets	ターゲットグループ
RemoveListenerCertificates	リスナー
RemoveTags	ロードバランサー、ターゲットグループ
SetIpAddressType	ロードバランサー
SetRulePriorities	リスナールール
SetSecurityGroups	ロードバランサー

API アクション	リソース ARN
SetSubnets	ロードバランサー

#### API バージョン 2012-06-01

API アクション	リソース ARN
AddTags	ロードバランサー
ApplySecurityGroupsToLoadBalancer	ロードバランサー
AttachLoadBalancerToSubnets	ロードバランサー
ConfigureHealthCheck	ロードバランサー
CreateAppCookieStickinessPolicy	ロードバランサー
CreateLBCookieStickinessPolicy	ロードバランサー
CreateLoadBalancer	ロードバランサー
CreateLoadBalancerListeners	ロードバランサー
CreateLoadBalancerPolicy	ロードバランサー
DeleteLoadBalancer	ロードバランサー
DeleteLoadBalancerListeners	ロードバランサー
DeleteLoadBalancerPolicy	ロードバランサー
DeregisterInstancesFromLoadBalancer	ロードバランサー
DetachLoadBalancerFromSubnets	ロードバランサー
DisableAvailabilityZonesForLoadBalancer	ロードバランサー
EnableAvailabilityZonesForLoadBalancer	ロードバランサー
ModifyLoadBalancerAttributes	ロードバランサー
RegisterInstancesWithLoadBalancer	ロードバランサー
RemoveTags	ロードバランサー
SetLoadBalancerListenerSSLCertificate	ロードバランサー
SetLoadBalancerPoliciesForBackendServer	ロードバランサー
SetLoadBalancerPoliciesOfListener	ロードバランサー

## Elastic Load Balancing の条件キー

ポリシーを作成するときは、ポリシーをいつ有効にするか制御する条件を指定できます。各条件には 1 つ以上のキーと値のペアが含まれます。グローバル条件キーとサービス固有の条件キーがあります。

The `elasticloadbalancing:ResourceTag/key` 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

グローバル条件キーの詳細については、『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。

以下のアクションでは `aws:RequestTag/##` および `aws:TagKeys` 条件キーがサポートされています。

- AddTags

- `CreateLoadBalancer`
- `RemoveTags`

## 事前定義の AWS 管理ポリシー

AWS によって作成された管理ポリシーは、一般的ユースケースに必要なアクセス権限を付与します。これらのポリシーを、Elastic Load Balancing に対して必要なアクセス権に基づいて IAM ユーザーにアタッチできます。

- `[ElasticLoadBalancingFullAccess]` — Grants full access required to use Elastic Load Balancing の機能を使用するために必要なフルアクセスを付与します。
- `[ElasticLoadBalancingReadOnly]` — Elastic Load Balancing の機能に対する読み取り専用アクセスを付与します。

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 アクセス許可 \(p. 15\)](#)」を参照してください。

## Elastic Load Balancing API 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 IAM ユーザーに付与する必要があります。詳しくは、「[Elastic Load Balancing の API アクション \(p. 10\)](#)」を参照してください。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 IAM 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 IAM 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 IAM 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 サービスにリンクされたロール

Elastic Load Balancing は、&ELB; 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:DescribeAddresses
- ec2:DescribeInstances
- ec2:DescribeSubnets
- ec2:DescribeSecurityGroups
- ec2:DescribeVpcs
- ec2:DescribeInternetGateways
- ec2:DescribeAccountAttributes
- ec2:DescribeClassicLinkInstances
- ec2:DescribeVpcClassicLink
- ec2:CreateSecurityGroup
- ec2:CreateNetworkInterface
- ec2>DeleteNetworkInterface
- ec2:ModifyNetworkInterface
- ec2:ModifyNetworkInterfaceAttribute
- ec2:AuthorizeSecurityGroupIngress
- ec2:AssociateAddress
- ec2:DisassociateAddress
- ec2:AttachNetworkInterface
- ec2:DetachNetworkInterface

- `ec2:AssignPrivateIpAddresses`
- `ec2:AssignIpv6Addresses`
- `ec2:UnassignIpv6Addresses`
- `logs:CreateLogDelivery`
- `logs:GetLogDelivery`
- `logs:UpdateLogDelivery`
- `logs>DeleteLogDelivery`
- `logs:ListLogDeliveries`

[`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を作成する

[`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の作成時に Elastic Load Balancing によって作成されます。

Elastic Load Balancing がお客様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お客様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』の「[AWS アカウントに新しいロールが表示される](#)」を参照してください。

## 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](#)」、「[ネットワークロードバランサーの削除](#)」、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。

# Classic Load Balancer の移行

VPC に既存の Classic Load Balancer があるが、Application Load Balancer または ネットワークロードバランサーの方がニーズを満たす可能性があるとは判断した場合は、Classic Load Balancer を移行できます。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。詳細については、「[Elastic Load Balancing 製品の比較](#)」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 \(p. 19\)](#)
- [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 \(p. 21\)](#)
- [ステップ 3: Classic Load Balancer への参照を更新する \(p. 21\)](#)
- [ステップ 4: Classic Load Balancer を削除する \(p. 22\)](#)

## ステップ 1: 新しいロードバランサーを作成する

Classic Load Balancer と同じ設定で Application Load Balancer または ネットワークロードバランサーを作成します。

次のいずれかの方法を使用して、ロードバランサーとターゲットグループを作成できます。

- [コンソールの移行ウィザード \(p. 19\)](#)
- [ロードバランサーコピーユーティリティ \(p. 20\)](#)
- [手動 \(p. 20\)](#)

## オプション 1: 移行ウィザードを使用した移行

移行ウィザードにより、Classic Load Balancer の設定に基づいて Application Load Balancer または ネットワークロードバランサーが作成されます。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 内にある必要があります。
- 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 を作成できます。Classic Load Balancer に TCP リスナーがある場合、ウィザードは ネットワークロードバランサーを作成できます。
- Classic Load Balancer の名前が既存の Application Load Balancer または ネットワークロードバランサーの名前と一致する場合、ウィザードでは、新しいロードバランサー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スがある場合は、ネットワーク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:  
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 HTTP ヘルスチェックに変更され、デフォルトでは Application Load Balancer の作成時にパスが「/」に設定されます。
- Classic Load Balancer を ネットワークロードバランサー に移行した場合、ヘルスチェック設定は Network Load Balancer の要件を満たすように変更されます。
- 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 TCP リスナーがある場合、ネットワークロードバランサーは TCP リスナーを使用しますが、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. <https://console.aws.amazon.com/ec2/> にある Amazon EC2 コンソールを開きます。
2. ナビゲーションペインの [LOAD BALANCING] で [Load Balancers] を選択します。
3. 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る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: ロードバランサーコピーユーティリティを使用した移行

このユーティリティは GitHub で入手できます。詳細については、「[ロードバランサーコピーユーティリティ](#)」を参照してください。

## オプション 3: 手動の移行

以下に示すのは、Classic Load Balancer に基づいて手動で新しいロードバランサーを作成するための一般的なプロセスです。AWS マネジメントコンソール、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、「[Elastic Load Balancing の開始方法 \(p. 8\)](#)」を参照してください。

- 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。
- 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。
- 以下のいずれかを行います。
  - Classic Load Balancer が Auto Scaling グループにアタッチされている場合は、ターゲットグループをその Auto Scaling グループにアタッチします。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。



- 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。
- 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後、トラフィックを段階的にリダイレクトするときに新しいロードバランサーのテストプロセスを開始できます。

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、ブラウザにサーバー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、Classic Load Balancer 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 Classic Load Balancer 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。これは、ドメイン名を解決した DNS サーバーは、レコード情報を 60 秒間キャッシュに保持することを意味します。したがって、前のステップを完了しても、これらの DNS サーバーでトラフィックの Classic Load Balancer へのルーティングが最大 60 秒間続行される可能性があり、その後、世界中の DNS サーバーに変更の伝達が始まります。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、Classic Load Balancer の DNS レコードを削除できます。

## ステップ 3: Classic Load Balancer への参照を更新する

Classic Load Balancer の移行が完了したら、以下のような &CLB; への参照を必ず更新してください。

- (aws elbv2 コマンドではなく) AWS CLI aws elb コマンドを使用するスクリプト
- バージョンではなく 2015-12-01) Elastic Load Balancing API バージョン 2012-06-01 を使用するコード
- API バージョン 2012-06-01 を使用する IAM ポリシー (バージョン 2015-12-01 ではなく)
- CloudWatch メトリクスを使用するプロセス
- AWS CloudFormation テンプレート

### リソース

- AWS CLI Command Reference の [elbv2](#)
- [Elastic Load Balancing API リファレンスバージョン 2015-12-01](#)



- [ロードバランサーの認証とアクセスコントロール](#) (p. 9)
- Application Load Balancer 用ユーザーガイドの[Application Load Balancer メトリクス](#)
- Network Load Balancer 用ユーザーガイドの[ネットワークロードバランサー メトリクス](#)
- AWS CloudFormation ユーザーガイドの [AWS::ElasticLoadBalancingV2::LoadBalancer](#)

## ステップ 4: Classic Load Balancer を削除する

すべてのトラフィックを新しいロードバランサーにリダイレクトし、Classic Load Balancer 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た後、Classic Load Balancer を削除できます。