



Benutzer-Leitfaden

Elastic Load Balancing



Elastic Load Balancing: Benutzer-Leitfaden

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Marken, die nicht im Besitz von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist ELB?	1
Vorteile des Load Balancers	1
Funktionen von ELB	1
Zugriff auf ELB	2
Zugehörige Services	2
Preisgestaltung	4
So funktioniert ELB	5
Availability Zones und Load-Balancer-Knoten	5
Zonenübergreifendes Load Balancing	6
Zonale Verschiebung	9
Weiterleitung von Anforderungen	9
Weiterleitungsalgorithmus	10
HTTP-Verbindungen	11
HTTP-Header	12
HTTP-Header-Limits	13
Load-Balancer-Schema	13
IP-Adressarten	14
Netzwerk-MTU	16
Erste Schritte	17
Sicherheit	18
Datenschutz	19
Verschlüsselung im Ruhezustand	20
Verschlüsselung während der Übertragung	20
Identity and Access Management	20
Zielgruppe	21
Authentifizierung mit Identitäten	21
Verwalten des Zugriffs mit Richtlinien	23
Wie arbeitet ELB mit IAM	25
API-Berechtigungen für Ressourcen-Tagging	37
Servicegebundene Rolle	39
AWS verwaltete Richtlinien	40
Compliance-Validierung	43
Ausfallsicherheit	43
Sicherheit der Infrastruktur	44

Netzwerkisolierung	45
Steuern des Netzwerkverkehrs	45
AWS PrivateLink	46
Erstellen Sie einen Schnittstellenendpunkt für ELB	46
Erstellen Sie eine VPC-Endpunktrichtlinie für ELB	47
Drosselung von API-Anforderungen	48
Wie wird die Drosselung angewendet	48
Anforderungsratenbegrenzung	49
Größen und Nachfüllraten für Token-Buckets anfragen	49
API-Anfragen überwachen	53
Fakturierungs- und Nutzungsberichte	54
Application Load Balancer	54
Network Load Balancers	55
Gateway Load Balancer	55
Classic Load Balancer	55
Protokollieren von -API-Aufrufen	57
Veranstaltungen des ELB-Managements in CloudTrail	58
Beispiele für ELB-Ereignisse	59
Migrieren Ihres Classic Load Balancers	63
Vorteile der Migration	63
Migrationsassistent	64
Kopieren Sie die Migration des Dienstprogramms	66
Manuelle Migration	66
Verhindern Sie, dass Benutzer Classic Load Balancer erstellen	69

Was ist ELB?

ELB verteilt Ihren eingehenden Datenverkehr automatisch auf mehrere Ziele wie EC2 Instances, Container und IP-Adressen in einer oder mehreren Availability Zones. Es überwacht den Zustand der registrierten Ziele und leitet den Datenverkehr nur an die fehlerfreien Ziele weiter. ELB skaliert Ihre Load Balancer-Kapazität automatisch als Reaktion auf Änderungen des eingehenden Datenverkehrs.

Vorteile des Load Balancers

Ein Load Balancer verteilt Workloads über mehrere Datenverarbeitungs-Ressourcen, wie z. B. virtuelle Server. Durch Verwendung eines Load Balancers erhöhen sich die Verfügbarkeit und die Fehlertoleranz Ihrer Anwendungen.

Sie können Datenverarbeitungs-Ressourcen zu Ihrem Load Balancer hinzufügen oder entfernen, wenn sich Ihre Bedürfnisse ändern, ohne den allgemeinen Fluss von Anfragen an Ihre Anwendung zu unterbrechen.

Sie können Zustandsprüfungen konfigurieren, mit denen der Zustand der Datenverarbeitungsressourcen überwacht wird, sodass der Load Balancer nur an die fehlerfreien Ziele Anfragen sendet. Sie können zudem die Ver- und Entschlüsselung auf Ihren Load Balancer auslagern, sodass sich Ihre Datenverarbeitungsressourcen auf ihre Hauptaufgaben konzentrieren können.

Funktionen von ELB

ELB unterstützt mehrere Load Balancer-Typen. Sie können den Typ des Load Balancers, der Ihren Anforderungen am besten entspricht, auswählen. Weitere Informationen finden Sie unter .

Weitere Informationen zu den Load Balancern der aktuellen Generation finden Sie in der folgenden Dokumentation:

- [Benutzerhandbuch für Application Load Balancer](#)
- [Benutzerhandbuch für Network Load Balancer](#)
- [Benutzerhandbuch für Gateway Load Balancer](#)

Classic Load Balancers sind die vorherige Generation von Load Balancern von ELB. Es wird empfohlen, zu einem Load Balancer der aktuellen Generation zu migrieren. Weitere Informationen finden Sie unter [Migration Ihres Classic Load Balancer](#).

Zugriff auf ELB

Mit den folgenden Schnittstellen können Sie Ihre Load Balancer erstellen, verwalten und darauf zugreifen:

- AWS-Managementkonsole— Stellt eine Weboberfläche bereit, über die Sie auf ELB zugreifen können.
- AWS Befehlszeilenschnittstelle (AWS CLI) — Stellt Befehle für eine Vielzahl von AWS Diensten bereit, einschließlich ELB. Das AWS CLI wird unter Windows, MacOS und Linux unterstützt. Weitere Informationen finden Sie unter [AWS Command Line Interface](#).
- AWS SDKs— Geben Sie sprachspezifische Informationen an APIs und kümmern Sie sich um viele Verbindungsdetails, z. B. um die Berechnung von Signaturen, die Bearbeitung von Wiederholungsversuchen von Anfragen und die Fehlerbehandlung. Weitere Informationen finden Sie unter [AWS SDKs](#).
- Abfrage-API – Bietet API-Aktionen auf niedriger Ebene, die Sie mithilfe von HTTPS-Anforderungen aufrufen. Die Verwendung der Abfrage-API ist der direkteste Weg, um auf ELB zuzugreifen. Allerdings müssen dann viele technische Abläufe, wie beispielsweise das Erzeugen des Hashwerts zum Signieren der Anforderung und die Fehlerbehandlung, in der Anwendung durchgeführt werden. Weitere Informationen finden Sie hier:
 - [Application Load Balancer, Network Load Balancer und Gateway Load Balancer — API-Version 2015-12-01](#)
 - Classic Load Balancer – [API-Version 2012-06-01](#)

Zugehörige Services

ELB arbeitet mit den folgenden Diensten zusammen, um die Verfügbarkeit und Skalierbarkeit Ihrer Anwendungen zu verbessern.

- Amazon EC2 — Virtuelle Server, auf denen Ihre Anwendungen in der Cloud ausgeführt werden. Sie können Ihren Load Balancer so konfigurieren, dass er den Datenverkehr an Ihre EC2 Instances weiterleitet. Weitere Informationen finden Sie im [EC2 Amazon-Benutzerhandbuch](#).

- Amazon EC2 Auto Scaling — Stellt sicher, dass Sie die gewünschte Anzahl von Instances ausführen, auch wenn eine Instance ausfällt. Amazon EC2 Auto Scaling ermöglicht es Ihnen auch, die Anzahl der Instances automatisch zu erhöhen oder zu verringern, wenn sich die Nachfrage nach Ihren Instances ändert. Wenn Sie Auto Scaling mit ELB aktivieren, werden Instances, die von Auto Scaling gestartet werden, automatisch beim Load Balancer registriert. Ebenso werden Instances, die durch Auto Scaling beendet werden, automatisch vom Load Balancer entfernt. Weitere Informationen finden Sie im [Amazon EC2 Auto Scaling Benutzerhandbuch](#).
- AWS Certificate Manager – Wenn Sie einen HTTPS-Listener erstellen, können Sie von ACM bereitgestellte Zertifikate festlegen. Der Load Balancer verwendet Zertifikate, um Verbindungen zu beenden und Anfragen von Clients zu entschlüsseln.
- Amazon CloudWatch — Ermöglicht es Ihnen, Ihren Load Balancer zu überwachen und bei Bedarf Maßnahmen zu ergreifen. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).
- Amazon ECS — Ermöglicht das Ausführen, Stoppen und Verwalten von Docker-Containern in einem Cluster von EC2 Instances. Sie können Ihren Load Balancer so konfigurieren, dass der Datenverkehr an Ihre Container geleitet wird. Weitere Informationen finden Sie im [Amazon Elastic Container Service-Entwicklerhandbuch](#).
- AWS Global Accelerator – Verbessert die Verfügbarkeit und Leistung Ihrer Anwendung. Verwenden Sie einen Beschleuniger, um den Datenverkehr auf mehrere Load Balancer in einer oder mehreren AWS Regionen zu verteilen. Weitere Informationen finden Sie im [AWS Global Accelerator - Entwicklerhandbuch](#).
- Route 53 – Bietet eine zuverlässige und kostengünstige Möglichkeit, um Besucher zu Webseiten zu leiten, indem Domainnamen in numerische IP-Adressen, die Computer zur gegenseitigen Vernetzung verwenden, übersetzt werden. Dies würde beispielsweise `www.example.com` in die numerische IP-Adresse `192.0.2.1` übersetzt werden. AWS weist Ihren Ressourcen URLs zu, z. B. Load Balancern. Sie können jedoch auch eine URL verwenden, die aussagekräftig und leicht zu merken ist. So können Sie zum Beispiel Ihren Domainnamen einem Load Balancer zuordnen. Weitere Informationen finden Sie im [Amazon Route 53-Entwicklerhandbuch](#).
- AWS WAF— Sie können es AWS WAF zusammen mit Ihrem Application Load Balancer verwenden, um Anfragen auf der Grundlage der Regeln in einer Web-Zugriffskontrollliste (Web-ACL) zuzulassen oder zu blockieren. Weitere Informationen finden Sie im [AWS WAF - Entwicklerhandbuch](#).

Preisgestaltung

Mit Ihrem Load Balancer zahlen Sie nur für das, was Sie auch tatsächlich nutzen. Weitere Informationen finden Sie unter [ELB-Preise](#).

So funktioniert ELB

Ein Load Balancer akzeptiert eingehenden Datenverkehr von Clients und leitet Anfragen an seine registrierten Ziele (z. B. EC2 Instances) in einer oder mehreren Availability Zones weiter. Der Load Balancer überwacht auch den Zustand seiner registrierten Ziele und stellt sicher, dass er den Datenverkehr nur an ordnungsgemäß funktionierende Ziele weiterleitet. Wenn der Load Balancer ein fehlerhaftes Ziel erkennt, stoppt er das Weiterleiten von Datenverkehr an dieses Ziel. Anschließend wird die Weiterleitung von Datenverkehr an dieses Ziel fortgesetzt, wenn er erkennt, dass das Ziel wieder fehlerfrei ist.

Sie konfigurieren Ihren Load Balancer für eingehenden Datenverkehr, indem Sie einen oder mehrere Listener angeben. Ein Listener ist ein Prozess, der Verbindungsanfragen überprüft. Es wird mit einem Protokoll und einer Portnummer für Verbindungen von Clients zum Load Balancer konfiguriert. Ebenso ist es mit einem Protokoll und einer Portnummer für Verbindungen vom Load Balancer zu den Zielen konfiguriert.

Inhalt

- [Availability Zones und Load-Balancer-Knoten](#)
- [Weiterleitung von Anforderungen](#)
- [Load-Balancer-Schema](#)
- [IP-Adressarten](#)
- [Netzwerk-MTU für Ihren Load Balancer](#)

Availability Zones und Load-Balancer-Knoten

Wenn Sie eine Availability Zone für Ihren Load Balancer aktivieren, erstellt ELB einen Load Balancer-Knoten in der Availability Zone. Wenn Sie Ziele in einer Availability Zone registrieren, aber die Availability Zone nicht aktivieren, erhalten diese registrierten Ziele keinen Datenverkehr. Ihr Load Balancer ist am effektivsten, wenn Sie dafür sorgen, dass jede aktivierte Availability Zone mindestens ein registriertes Ziel hat.

Wir empfehlen, mehrere Availability Zones für alle Load Balancer zu aktivieren. Bei einem Application Load Balancer ist es jedoch erforderlich, dass Sie mindestens zwei Availability Zones aktivieren. Diese Konfiguration stellt sicher, dass der Load Balancer weiterhin Datenverkehr weiterleiten kann. Der Load Balancer kann den Datenverkehr an fehlerfreie Ziele in einer anderen Availability Zone weiterleiten, falls eine Availability Zone ausfällt oder keine fehlerfreien Ziele mehr hat.

Nachdem Sie eine Availability Zone deaktiviert haben, bleiben die Ziele in dieser Availability Zone für den Load Balancer registriert. Auch wenn sie registriert bleiben, leitet der Load Balancer keinen Datenverkehr an sie weiter.

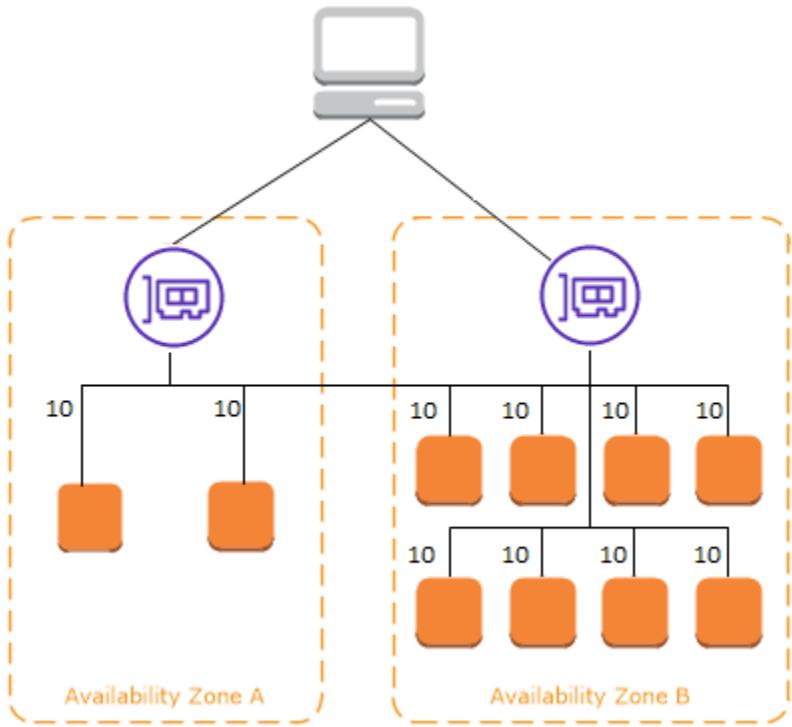
Zonenübergreifendes Load Balancing

Die Knoten für Ihren Load Balancer verteilen Anforderungen von Clients auf registrierte Ziele.

Wenn zonenübergreifendes Load Balancing aktiviert ist, verteilt jeder Load Balancer-Knoten den Datenverkehr gleichmäßig auf die registrierten Ziele in allen aktivierten Availability Zones. Wenn zonenübergreifendes Load Balancing deaktiviert ist, verteilt jeder Load Balancer-Knoten den Datenverkehr gleichmäßig nur auf die registrierten Ziele in seiner Availability Zone.

Die folgenden Diagramme veranschaulichen die Auswirkungen des zonenübergreifenden Load Balancings mit Round Robin als Standard-Routing-Algorithmus. Es gibt zwei aktivierte Availability Zones mit zwei Zielen in Availability Zone A und acht Zielen in Availability Zone B. Clients senden Anfragen und Amazon Route 53 beantwortet jede Anfrage mit der IP-Adresse eines der Load-Balancer-Knoten. Basierend auf dem Round-Robin-Routing-Algorithmus wird der Datenverkehr so verteilt, dass jeder Load-Balancer-Knoten 50 % des Datenverkehrs von den Clients erhält. Jeder Load Balancer-Knoten verteilt seinen Anteil des Datenverkehrs auf die registrierten Ziele in seinem Anwendungsbereich.

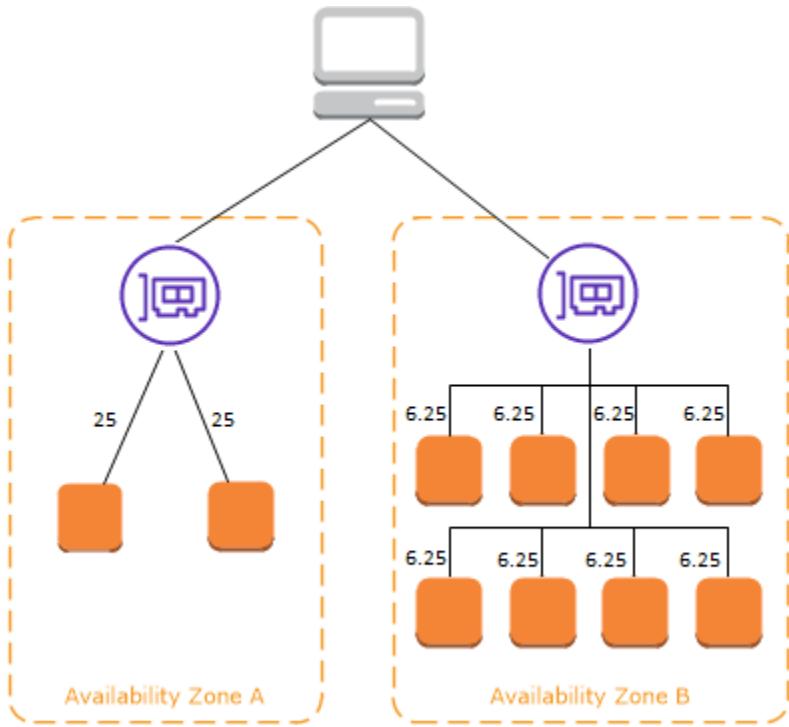
Wenn zonenübergreifendes Load Balancing aktiviert ist, erhält jedes der 10 Ziele 10 % des Datenverkehrs. Der Grund hierfür ist, dass jeder Load Balancer-Knoten seine 50 % des Client-Datenverkehrs an alle 10 Ziele weiterleiten kann.



Wenn zonenübergreifendes Load Balancing deaktiviert ist:

- Jedes der beiden Ziele in Availability Zone A erhält 25 % des Datenverkehrs.
- Jedes der acht Ziele in Availability Zone B erhält 6,25 % des Datenverkehrs.

Der Grund hierfür ist, dass jeder Load Balancer-Knoten seine 50 % des Client-Datenverkehrs nur an Ziele in seiner Availability Zone weiterleiten kann.



Bei Application Load Balancern ist zonenübergreifendes Load Balancing immer auf Load-Balancer-Ebene aktiviert. Auf Zielgruppenebene kann zonenübergreifendes Load Balancing deaktiviert werden. Weitere Informationen finden Sie unter [Deaktivieren von zonenübergreifendem Load Balancing](#) im Benutzerhandbuch für Application Load Balancer.

Bei Network Load Balancern und Gateway Load Balancern ist zonenübergreifendes Load Balancing standardmäßig deaktiviert. Nachdem Sie einen Load Balancer erstellt haben, können Sie zonenübergreifendes Load Balancing jederzeit aktivieren oder deaktivieren. Weitere Informationen finden Sie unter [Zonenübergreifendes Load Balancing](#) im Benutzerhandbuch für Network Load Balancers.

Wenn Sie einen Classic Load Balancer erstellen, hängt die Voreinstellung für zonenübergreifendes Load Balancing davon ab, wie Sie den Load Balancer erstellen. Mit der API oder CLI wird das zonenübergreifende Load Balancing standardmäßig deaktiviert. Bei der ist die AWS-Managementkonsole Option zur Aktivierung des zonenübergreifenden Lastenausgleichs standardmäßig ausgewählt. Nachdem Sie einen Classic Load Balancer erstellt haben, können Sie zonenübergreifendes Load Balancing jederzeit aktivieren oder deaktivieren. Weitere Informationen finden Sie unter [Aktivieren von zonenübergreifendem Load Balancing](#) im Benutzerhandbuch für Classic Load Balancer.

Zonale Verschiebung

Zonal Shift ist eine Funktion in Amazon Application Recovery Controller (ARC) (ARC). Mit der Zonenverschiebung können Sie eine Load-Balancer-Ressource mit einer einzigen Aktion aus einer beeinträchtigten Availability Zone verlagern. Auf diese Weise können Sie den Betrieb von anderen fehlerfreien Availability Zones in einer AWS-Region fortsetzen.

Wenn Sie eine Zonenverschiebung starten, sendet Ihr Load Balancer den Datenverkehr für die Ressource nicht mehr an die betroffene Availability Zone. ARC erzeugt die Zonenverschiebung sofort. Es kann jedoch eine kurze Zeit dauern, in der Regel bis zu einigen Minuten, bis bestehende Verbindungen in der betroffenen Availability Zone hergestellt sind. Weitere Informationen finden Sie unter [So funktioniert eine zonale Schicht: Zustandsprüfungen und zonale IP-Adressen](#) im Amazon Application Recovery Controller (ARC) Developer Guide.

Bevor Sie die Zonenverschiebung verwenden, sollten Sie Folgendes beachten:

- Zonal Shift wird unterstützt, wenn Sie einen Network Load Balancer mit aktiviertem oder deaktiviertem zonenübergreifendem Load Balancing verwenden.
- Sie können eine Zonenverschiebung für einen bestimmten Load Balancer nur für eine Availability Zone starten. Eine Zonenverschiebung lässt sich nicht für mehrere Availability Zones starten.
- AWS entfernt proaktiv IP-Adressen des zonalen Load Balancers aus DNS, wenn sich mehrere Infrastrukturprobleme auf Dienste auswirken. Prüfen Sie immer die aktuelle Kapazität der Availability Zone, bevor Sie mit einer Zonenverschiebung beginnen. Wenn bei Ihren Load Balancern das zonenübergreifende Load Balancing deaktiviert ist und Sie eine Zonenverschiebung verwenden, um eine zonale Load-Balancer-IP-Adresse zu entfernen, verliert die Availability Zone, die von der Zonenverschiebung betroffen ist, auch die Zielkapazität.

Weitere Anleitungen und Informationen finden Sie unter [Bewährte Methoden für Zonenverschiebungen in ARC](#) im Amazon Application Recovery Controller (ARC) Developer Guide.

Weiterleitung von Anforderungen

Bevor ein Client eine Anforderung an Ihren Load Balancer sendet, löst der Load Balancer den Domainnamen mithilfe eines Domain Name System- (DNS)-Server auf. Der DNS-Eintrag wird von Amazon gesteuert, da sich Ihre Load Balancer in der `amazonaws.com`-Domain befinden. Die Amazon DNS-Server geben mindestens eine IP-Adresse an den Client zurück. Dies sind die IP-

Adressen der Load Balancer-Knoten für Ihren Load Balancer. Mit Network Load Balancers erstellt ELB eine Netzwerkschnittstelle für jede Availability Zone, die Sie aktivieren, und verwendet sie, um eine statische IP-Adresse abzurufen. Sie können optional eine Elastic-IP-Adresse mit jeder Netzwerkschnittstelle verknüpfen, wenn Sie den Network Load Balancer erstellen.

Wenn sich der Datenverkehr zu Ihrer Anwendung im Laufe der Zeit ändert, skaliert ELB Ihren Load Balancer und aktualisiert den DNS-Eintrag. Der DNS-Eintrag gibt auch die time-to-live (TTL) von 60 Sekunden an. Dadurch wird sichergestellt, dass die IP-Adressen aufgrund des sich ändernden Datenverkehrs schnell neu zugeordnet werden können.

Der Client bestimmt, welche IP-Adresse zum Senden von Anforderungen an den Load Balancer verwendet werden. Der Load Balancer-Knoten, der die Anforderung empfängt, wählt ein fehlerfreies registriertes Ziel aus und sendet die Anforderung über die private IP-Adresse an das Ziel.

Weitere Informationen finden Sie unter [Weiterleiten von Datenverkehr an einen ELB Load Balancer](#) im Entwicklerhandbuch von Amazon Route 53.

Weiterleitungsalgorithmus

Bei Application Load Balancern verwendet der Load-Balancer-Knoten, der die Anfrage empfängt, den folgenden Prozess:

1. Wertet die Listener-Regeln in der Reihenfolge ihrer Priorität aus, um zu bestimmen, welche Regel angewendet werden soll.
2. Wählt ein Ziel aus der Zielgruppe für die Regelaktion aus, wobei der für die Zielgruppe konfigurierte Routingalgorithmus verwendet wird. Der Standard-Routingalgorithmus ist Round Robin. Die Weiterleitung erfolgt unabhängig für jede Zielgruppe, auch wenn ein Ziel bei mehreren Zielgruppen registriert ist.

Bei Network Load Balancern verwendet der Load-Balancer-Knoten, der die Verbindung empfängt, den folgenden Prozess:

1. Wählt ein Ziel aus der Zielgruppe für die Standardregel mit einem Flow-Hash-Algorithmus aus. Basiert den Algorithmus auf:
 - Protokoll
 - Quell-IP-Adresse und Quellport
 - Ziel-IP-Adresse und Zielport
 - TCP-Sequenznummer

2. Leitet jede einzelne TCP-Verbindung für die Dauer der Verbindung an ein einzelnes Ziel weiter.

Die TCP-Verbindungen von einem Client verfügen über unterschiedliche Quell-Ports und Sequenznummern und können an verschiedene Ziele geleitet werden.

Bei Gateway Load Balancers verwendet der Load Balancer-Knoten, der die Verbindung empfängt, einen 5-Tupel-Flow-Hash-Algorithmus, um eine Ziel-Appliance auszuwählen. Nachdem ein Flow eingerichtet wurde, werden alle Pakete für denselben Flow konsistent an dieselbe Ziel-Appliance weitergeleitet. Der Load Balancer und die Ziel-Appliances tauschen Datenverkehr über das GENEVE-Protokoll auf Port 6081 aus.

Bei Classic Load Balancern wählt der Load-Balancer-Knoten, der die Anfrage empfängt, wie folgt eine registrierte Instance aus:

- Verwendet den Roundrobin-Weiterleitungsalgorithmus für TCP-Listener
- Routingalgorithmus für HTTP- und HTTPS-Listener mit den wenigsten ausstehenden Anforderungen

HTTP-Verbindungen

Classic Load Balancer verwenden vorab geöffnete Verbindungen, Application Load Balancer jedoch nicht. Sowohl Classic Load Balancer als auch Application Load Balancer verwenden Verbindigmultiplexing. Das bedeutet, dass Anfragen von mehreren Clients auf mehreren Front-End-Verbindungen über eine einzige Back-End-Verbindung an ein bestimmtes Ziel weitergeleitet werden können. Das Verbindigmultiplexing verbessert die Latenz und reduziert die Last für Ihre Anwendungen. Um das Verbindigmultiplexing zu verhindern, deaktivieren Sie HTTP-keep-alive-Header, indem Sie den Connection: close-Header in Ihren HTTP-Antworten festlegen.

Application Load Balancer und Classic Load Balancer unterstützen HTTP über Pipelines auf Frontend-Verbindungen. Es werden jedoch keine HTTP-Pipelines für Backend-Verbindungen unterstützt.

Application Load Balancer unterstützen die folgenden HTTP-Anforderungsmethoden: GET, HEAD, POST, PUT, DELETE, OPTIONS und PATCH.

Application Load Balancer unterstützen die folgenden Protokolle in Frontend-Verbindungen: HTTP/0.9, HTTP/1.0, HTTP/1.1 und HTTP/2. Sie können HTTP/2 nur mit HTTPS-Listenern verwenden und bis zu 128 Anfragen parallel mit einer HTTP/2-Verbindung senden. Application Load Balancers unterstützen auch Verbindungs-Upgrades von HTTP auf WebSockets. Wenn es jedoch

ein Verbindungs-Upgrade gibt, gelten die Routing-Regeln und AWS WAF Integrationen für den Application Load Balancer Listener nicht mehr.

Application Load Balancer verwenden standardmäßig HTTP/1.1 für Backend-Verbindungen (Load Balancer zum registrierten Ziel). Sie können jedoch die Protokollversion verwenden, um die Anfrage mit HTTP/2 oder gRPC an die Ziele zu senden. Weitere Informationen finden Sie unter [Protokollversionen](#). Der `keep-alive`-Header wird bei Backend-Verbindungen standardmäßig unterstützt. Für HTTP/1.0 Anforderungen von Clients, die keinen Host-Header haben, generiert der Load Balancer einen Host-Header für die HTTP/1.1-Anforderungen, die über die Backend-Verbindungen gesendet werden. Der Host-Header enthält den DNS-Namen des Load Balancers.

Classic Load Balancer unterstützen die folgenden Protokolle in Frontend-Verbindungen (Client zu Load Balancer): HTTP/0.9, HTTP/1.0 und HTTP/1.1. Sie verwenden HTTP/1.1 für Backend-Verbindungen (Load Balancer zu registriertem Ziel). Der `keep-alive`-Header wird bei Backend-Verbindungen standardmäßig unterstützt. Für HTTP/1.0 Anforderungen von Clients, die keinen Host-Header haben, generiert der Load Balancer einen Host-Header für die HTTP/1.1-Anforderungen, die über die Backend-Verbindungen gesendet werden. Der Host-Header enthält die IP-Adresse des Load-Balancer-Knotens.

HTTP-Header

Application Load Balancer und Classic Load Balancer fügen automatisch die Header X-Forwarded-For, X-Forwarded-Proto und X-Forwarded-Port zur Anfrage hinzu.

Application Load Balancer konvertieren die Hostnamen in HTTP-Host-Header in Kleinbuchstaben, bevor sie an Ziele gesendet werden.

Für Frontend-Verbindungen mit HTTP/2 sind die Header-Namen in Kleinbuchstaben angegeben. Bevor die Anfrage per HTTP/1.1 an das Ziel gesendet wird, werden die folgenden Header-Namen in Groß- und Kleinschreibung konvertiert: X-Forwarded-For, X-Forwarded-Proto, X-Forwarded-Port, Host, X-Amzn-Trace-ID, Upgrade und Connection. Alle anderen Header-Namen sind in Kleinbuchstaben.

Application Load Balancer und Classic Load Balancer berücksichtigen den Verbindungs-Header aus der eingehenden Client-Anfrage, nachdem die Antwort über Proxy an den Client zurückgegeben wurde.

Wenn Application Load Balancer und Classic Load Balancer, die HTTP/1.1 verwenden, einen `Expect: 100-Continue`-Header erhalten, antworten sie sofort mit `HTTP/1.1 100 Continue`, ohne den Content-

Length-Header zu prüfen. Der Anforderungsheader Expect: 100-Continue wird nicht an seine Ziele weitergeleitet.

Bei Verwendung von HTTP/2 unterstützen Application Load Balancer den Expect: 100-Continue-Header von Client-Anfragen nicht. Der Application Load Balancer wird nicht mit HTTP/2 100 Continue antworten oder diesen Header an die Ziele weiterleiten.

HTTP-Header-Limits

Die folgenden Größenbeschränkungen für Application Load Balancer sind harte Grenzwerte, die nicht geändert werden können.

- Anforderungszeile: 16 K
- Einzelner Header: 16 K
- Gesamter Antwort-Header: 32 K
- Gesamter Anfrage-Header: 64 K

Load-Balancer-Schema

Wenn Sie einen Load Balancer erstellen, müssen Sie entscheiden, ob es ein interner Load Balancer oder ein mit dem Internet verbundener Load Balancer werden soll.

Die Knoten eines mit dem Internet verbundenen Load Balancers haben öffentliche IP-Adressen. Der DNS-Name eines mit dem Internet verbundenen Load Balancers ist öffentlich zu den öffentlichen IP-Adressen der Knoten auflösbar. Daher können mit dem Internet verbundene Load Balancer Anfragen von Clients über das Internet weiterleiten.

Die Knoten eines internen Load Balancers haben nur private IP-Adressen. Der DNS-Name eines internen Load Balancers ist öffentlich zu den privaten IP-Adressen der Knoten auflösbar. Daher kann der interne Load Balancer nur Anforderungen von Clients mit Zugriff auf die VPC für den Load Balancer weiterleiten.

Sowohl mit dem Internet verbundene als auch interne Load Balancer leiten Anfragen an Ihre Ziele unter Verwendung privater IP-Adressen weiter. Daher benötigen Ihre Ziele keine öffentlichen IP-Adressen für den Empfang von Anfragen von einem internen oder einem mit dem Internet verbundenen Load Balancer.

Wenn Ihre Anwendung über mehrere Ebenen verfügt, können Sie eine Architektur entwerfen, die sowohl interne als auch internetbasierte Load Balancer verwendet. Dies gilt beispielsweise,

wenn Ihre Anwendung Webserver verwendet, die mit dem Internet verbunden sein müssen, und Anwendungsserver, die nur mit den Webservern verbunden sind. Erstellen Sie einen mit dem Internet verbundenen Load Balancer und registrieren Sie die Webserver bei ihm. Erstellen Sie einen internen Load Balancer und registrieren Sie die Anwendungsserver bei ihm. Die Webserver empfangen Anforderungen von dem mit dem Internet verbundenen Load Balancer und senden die Anforderungen für die Anwendungsserver an den internen Load Balancer. Die Anwendungsserver empfangen Anforderungen vom internen Load Balancer.

IP-Adresstypen

Der IP-Adresstyp, den Sie für Ihren Load Balancer angeben, bestimmt, wie Clients mit dem Load Balancer kommunizieren können.

- IPv4 nur — Clients kommunizieren über öffentliche und private IPv4 Adressen. Die Subnetze, die Sie für Ihren Load Balancer auswählen, müssen IPv4 Adressbereiche haben.
- Dualstack — Clients kommunizieren über öffentliche und private Adressen. IPv4 IPv6 Die Subnetze, die Sie für Ihren Load Balancer auswählen, müssen Adressbereiche haben IPv4 . IPv6
- Dualstack ohne öffentlich IPv4 — Clients kommunizieren über öffentliche und private IPv6 Adressen sowie private Adressen. IPv4 Die Subnetze, die Sie für Ihren Load Balancer auswählen, müssen Adressbereiche haben IPv4 . IPv6 Diese Option wird vom `internal` Load Balancer-Schema nicht unterstützt.

In der folgenden Tabelle werden die IP-Adresstypen beschrieben, die für jeden Load Balancer-Typ unterstützt werden.

Load balancer type (Load-Balancer-Typ)	IPv4 nur	Dualstack	Dualstack ohne Öffentlichkeit IPv4
Application Load Balancer	Ja	Ja	Ja
Network Load Balancer	Ja	Ja	Nein
Gateway Load Balancer	Ja	Ja	Nein

Load balancer type (Load-Balancer-Typ)	IPv4 nur	Dualstack	Dualstack ohne Öffentlichkeit IPv4
Classic Load Balancer	Ja	Nein	Nein

Der IP-Adresstyp, den Sie für Ihre Zielgruppe angeben, bestimmt, wie der Load Balancer mit Zielen kommunizieren kann.

- IPv4 nur — Der Load Balancer kommuniziert über private Adressen. IPv4 Sie müssen Ziele mit IPv4 Adressen registrieren, die zu einer IPv4 Zielgruppe gehören.
- IPv6 nur — Der Load Balancer kommuniziert über IPv6 Adressen. Sie müssen Ziele mit IPv6 Adressen registrieren, die einer IPv6 Zielgruppe angehören. Die Zielgruppe muss mit einem Dual-Stack-Loadbalancer verwendet werden.

In der folgenden Tabelle werden die IP-Adresstypen beschrieben, die für jedes Zielgruppenprotokoll unterstützt werden.

Protokoll der Zielgruppe	IPv4 nur	IPv6 nur
HTTP und HTTPS	Ja	Ja
TCP	Ja	Ja
TLS	Ja	Ja
UDP und TCP_UDP	Ja	Ja
GENF	-	-

Netzwerk-MTU für Ihren Load Balancer

Die maximale Übertragungseinheit (MTU) bestimmt die Größe des größten Pakets, das über das Netzwerk gesendet werden kann, in Bytes. Je größer die MTU einer Verbindung, desto mehr Daten können in einem einzelnen Paket übergeben werden. Ethernet-Rahmen bestehen aus dem Paket, also den eigentlichen Daten, die Sie senden, sowie aus den dazugehörigen Netzwerk-Overhead-Informationen. Über ein Internet-Gateway gesendeter Datenverkehr hat eine MTU von 1500. Das bedeutet, dass ein Paket mit mehr als 1500 Bytes fragmentiert und in mehreren Frames versendet wird. Wenn im IP-Header `Don't Fragment` festgelegt ist, wird das Paket gelöscht.

Die MTU-Größe auf Load-Balancer-Knoten ist nicht konfigurierbar. Jumbo-Frames (9001 MTU) sind Standard bei Load-Balancer-Knoten für Application Load Balancer, Network Load Balancer und Classic Load Balancer. Gateway Load Balancer unterstützen 8500 MTU. Weitere Informationen finden Sie unter [Maximum Transmission Unit \(MTU\)](#) im Benutzerhandbuch für Gateway Load Balancer.

Die Pfad-MTU ist die maximale Paketgröße, die auf dem Pfad zwischen dem sendenden Host und dem empfangenden Host unterstützt wird. Path MTU Discovery (PMTUD) wird verwendet, um den Pfad-MTU-Wert zwischen zwei Geräten zu ermitteln. Path MTU Discovery ist besonders wichtig, wenn der Client oder das Ziel keine Jumbo-Frames unterstützt.

Wenn ein Host ein Paket sendet, das größer als die MTU des empfangenden Hosts ist bzw. das größer als die MTU eines Geräts auf dem Pfad ist, löscht der empfangende Host bzw. das Gerät das Paket und gibt dann die folgende ICMP-Meldung zurück: `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Type 3, Code 4)`. Dies weist den übertragenden Host an, die Nutzlast in mehrere kleinere Pakete aufzuteilen und diese erneut zu übertragen.

Wenn Pakete, die größer als die MTU-Größe der Client- oder Zielschnittstelle sind, weiterhin gelöscht werden, funktioniert die Path MTU Discovery (PMTUD) wahrscheinlich nicht. Um dies zu vermeiden, stellen Sie sicher, dass die Path MTU Discovery durchgängig funktioniert und dass Sie Jumbo-Frames für Ihre Clients und Ziele aktiviert haben. Weitere Informationen zu Path MTU Discovery und zur Aktivierung von Jumbo Frames finden Sie unter [Path MTU Discovery](#) im EC2 Amazon-Benutzerhandbuch.

Erste Schritte mit ELB

ELB unterstützt mehrere Load Balancer-Typen. Sie können den Typ des Load Balancers, der Ihren Anforderungen am besten entspricht, auswählen. Weitere Informationen finden Sie unter [.](#)

Load Balancers

- [Erstellen Sie einen Application Load Balancer](#)
- [Einen Network Load Balancer erstellen](#)
- [Erstellen eines Gateway Load Balancers](#)

[Demos gängiger Load Balancer-Konfigurationen finden Sie unter ELB-Demos.](#)

Wenn bereits ein Classic Load Balancer vorhanden ist, können Sie zu einem Application Load Balancer oder zu einem Network Load Balancer migrieren. Weitere Informationen finden Sie unter [Migrieren Ihres Classic Load Balancers.](#)

Sicherheit in Elastic Load Balancing

Cloud-Sicherheit AWS hat höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die auf die Anforderungen der sicherheitssensibelsten Unternehmen zugeschnitten sind.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das [Modell der übergreifenden Verantwortlichkeit](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der [AWS](#) und [. Weitere Informationen zu den Compliance-Programmen, die für ELB gelten, finden Sie unter \[AWS Dienstleistungen im Umfang nach Compliance-Programmen AWS\]\(#\).](#)
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Dienst, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, einschließlich der Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Verwendung von ELB anwenden können. Es zeigt Ihnen, wie Sie ELB konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Dienste nutzen können, die Ihnen bei der Überwachung und Sicherung Ihrer ELB-Ressourcen helfen.

Bei einem [Gateway Load Balancer](#) sind Sie für die Auswahl und Qualifizierung der Software von Appliance-Anbietern verantwortlich. Sie müssen der Appliance-Software vertrauen, um den Datenverkehr vom Load Balancer zu untersuchen oder zu ändern, der auf Ebene 3 des Open Systems Interconnection (OSI)-Modells arbeitet, der Netzwerkebene. Die als [ELB-Partner](#) aufgeführten Appliance-Anbieter haben ihre Appliance-Software in integriert und qualifiziert AWS. Sie können der Appliance-Software von Anbietern aus dieser Liste ein höheres Maß an Vertrauen entgegenbringen. AWS Garantiert jedoch nicht die Sicherheit oder Zuverlässigkeit der Software dieser Anbieter.

Inhalt

- [Datenschutz in Elastic Load Balancing](#)
- [Identitäts- und Zugriffsmanagement für ELB](#)

- [Compliance-Validierung für Elastic Load Balancing](#)
- [Ausfallsicherheit beim Elastic Load Balancing](#)
- [Infrastruktursicherheit in Elastic Load Balancing](#)
- [Greifen Sie über einen Schnittstellenendpunkt \(AWS PrivateLink\) auf ELB zu](#)

Datenschutz in Elastic Load Balancing

Das AWS [Modell](#) der gilt für den Datenschutz in ELB. Wie in diesem Modell beschrieben, AWS ist verantwortlich für den Schutz der globalen Infrastruktur, auf der AWS Cloud alle Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im Blog-Beitrag [AWS -Modell der geteilten Verantwortung und in der DSGVO](#) im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter [Arbeiten mit CloudTrail Pfaden](#) im AWS CloudTrail Benutzerhandbuch.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-3-validated kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-3](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit ELB oder anderen Geräten arbeiten und dabei die Konsole, die API oder AWS-Services verwenden. AWS CLI AWS SDKs Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

Verschlüsselung im Ruhezustand

Wenn Sie die serverseitige Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) für Ihren S3-Bucket für ELB-Zugriffsprotokolle aktivieren, verschlüsselt ELB automatisch jede Zugriffsprotokolldatei, bevor sie in Ihrem S3-Bucket gespeichert wird. ELB entschlüsselt auch die Zugriffs-Protokolldateien, wenn Sie darauf zugreifen. Jede Protokolldatei ist mit einem eindeutigen Schlüssel verschlüsselt, der wiederum mit einem KMS-Schlüssel verschlüsselt wird, der regelmäßig gewechselt wird.

Verschlüsselung während der Übertragung

ELB vereinfacht den Prozess der Erstellung sicherer Webanwendungen, indem der HTTPS- und TLS-Verkehr von Clients am Load Balancer beendet wird. Der Load Balancer übernimmt die Ver- und Entschlüsselung des Datenverkehrs, anstatt zu verlangen, dass jede EC2 Instanz die Arbeit für die TLS-Terminierung übernimmt. Wenn Sie einen sicheren Listener konfigurieren, geben Sie die Verschlüsselungssammlungen und Protokollversionen an, die von Ihrer Anwendung unterstützt werden, sowie ein Serverzertifikat, das auf dem Load Balancer installiert werden soll. Sie können AWS Certificate Manager (ACM) oder AWS Identity and Access Management (IAM) verwenden, um Ihre Serverzertifikate zu verwalten. Application Load Balancer unterstützen HTTPS-Listener. Network Load Balancer unterstützen TLS-Listener. Classic Load Balancer unterstützen sowohl HTTPS- als auch TLS-Listener.

Identitäts- und Zugriffsmanagement für ELB

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service, den Zugriff auf Ressourcen sicher zu AWS kontrollieren. IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um ELB-Ressourcen zu verwenden. IAM ist ein Programm AWS-Service, das Sie ohne zusätzliche Kosten nutzen können.

Inhalt

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [Wie arbeitet ELB mit IAM](#)
- [ELB-API-Berechtigungen zum Markieren von Ressourcen während der Erstellung](#)
- [Serviceverknüpfte ELB-Rolle](#)
- [AWS verwaltete Richtlinien für ELB](#)

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in ELB ausführen.

Dienstbenutzer — Wenn Sie den ELB-Dienst für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr ELB-Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen.

Dienstadministrator — Wenn Sie in Ihrem Unternehmen für die ELB-Ressourcen verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf ELB. Es ist Ihre Aufgabe, zu bestimmen, auf welche ELB-Funktionen und -Ressourcen Ihre Servicebenutzer zugreifen sollen. Anschließend müssen Sie Anforderungen an Ihren IAM-Administrator senden, um die Berechtigungen der Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen.

IAM-Administrator — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien zur Verwaltung des Zugriffs auf ELB schreiben können.

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM-Benutzer authentifizieren oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich als föderierte Identität anmelden, indem Sie Anmeldeinformationen aus einer Identitätsquelle wie AWS IAM Identity Center (IAM Identity Center), Single Sign-On-Authentifizierung

oder Anmeldeinformationen verwenden. Google/Facebook Weitere Informationen zum Anmelden finden Sie unter [So melden Sie sich bei Ihrem AWS-Konto an](#) im Benutzerhandbuch für AWS-Anmeldung .

AWS bietet für den programmatischen Zugriff ein SDK und eine CLI zum kryptografischen Signieren von Anfragen. Weitere Informationen finden Sie unter [AWS Signature Version 4 for API requests](#) im IAM-Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, dem sogenannten AWS-Konto Root-Benutzer, der vollständigen Zugriff auf alle AWS-Services Ressourcen hat. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Eine Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Tasks that require root user credentials](#) im IAM-Benutzerhandbuch.

Verbundidentität

Es hat sich bewährt, dass menschliche Benutzer für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen einen Verbund mit einem Identitätsanbieter verwenden müssen.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensverzeichnis, Ihrem Directory Service Web-Identitätsanbieter oder der AWS-Services mithilfe von Anmeldeinformationen aus einer Identitätsquelle zugreift. Verbundene Identitäten übernehmen Rollen, die temporäre Anmeldeinformationen bereitstellen.

Für die zentrale Zugriffsverwaltung empfehlen wir AWS IAM Identity Center. Weitere Informationen finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center -Benutzerhandbuch.

IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität mit bestimmten Berechtigungen für eine einzelne Person oder Anwendung. Verwenden Sie möglichst temporäre Anmeldeinformationen anstelle von IAM-Benutzern mit langfristigen Anmeldeinformationen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter [Erfordern, dass menschliche Benutzer den Verbund mit einem Identitätsanbieter verwenden müssen, um AWS mithilfe temporärer Anmeldeinformationen darauf zugreifen zu können](#).

Eine [IAM-Gruppe](#) spezifiziert eine Sammlung von IAM-Benutzern und erleichtert die Verwaltung von Berechtigungen für große Gruppen von Benutzern. Weitere Informationen finden Sie unter [Anwendungsfälle für IAM-Benutzer](#) im IAM-Benutzerhandbuch.

IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität mit spezifischen Berechtigungen, die temporäre Anmeldeinformationen bereitstellt. Sie können eine Rolle übernehmen, indem Sie [von einer Benutzer zu einer IAM-Rolle \(Konsole\) wechseln](#) oder indem Sie eine AWS Oder-API-Operation AWS CLI aufrufen. Weitere Informationen finden Sie unter [Methoden, um eine Rolle zu übernehmen](#) im IAM-Benutzerhandbuch.

IAM-Rollen sind nützlich für Verbundbenutzerzugriff, temporäre IAM-Benutzerberechtigungen, kontoübergreifenden Zugriff, dienstübergreifenden Zugriff und Anwendungen, die auf Amazon ausgeführt werden. EC2 Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an Identitäten oder Ressourcen anhängen. AWS Eine Richtlinie definiert Berechtigungen, wenn sie mit einer Identität oder Ressource verknüpft sind. AWS bewertet diese Richtlinien, wenn ein Principal eine Anfrage stellt. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit Hilfe von Richtlinien legen Administratoren fest, wer Zugriff auf was hat, indem sie definieren, welches Prinzipal welche Aktionen auf welchen Ressourcen und unter welchen Bedingungen durchführen darf.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator erstellt IAM-Richtlinien und fügt sie zu Rollen hinzu, die die Benutzer dann übernehmen können. IAM-Richtlinien definieren Berechtigungen unabhängig von der Methode, die zur Ausführung der Operation verwendet wird.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität (Benutzer, Gruppe oder Rolle) anfügen können. Diese Richtlinien steuern, welche Aktionen Identitäten für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können Inline-Richtlinien (direkt in eine einzelne Identität eingebettet) oder verwaltete Richtlinien (eigenständige Richtlinien, die mit mehreren Identitäten verbunden sind) sein. Informationen dazu, wie Sie zwischen verwalteten und Inline-Richtlinien wählen, finden Sie unter [Choose between managed policies and inline policies](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele hierfür sind Vertrauensrichtlinien für IAM-Rollen und Amazon S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#).

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

Weitere Richtlinientypen

AWS unterstützt zusätzliche Richtlinientypen, mit denen die maximalen Berechtigungen festgelegt werden können, die durch gängigere Richtlinientypen gewährt werden:

- Berechtigungsgrenzen – Eine Berechtigungsgrenze legt die maximalen Berechtigungen fest, die eine identitätsbasierte Richtlinie einer IAM-Entität erteilen kann. Weitere Informationen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- Richtlinien zur Dienstkontrolle (SCPs) — Geben Sie die maximalen Berechtigungen für eine Organisation oder Organisationseinheit in an AWS Organizations. Weitere Informationen finden Sie unter [Service-Kontrollrichtlinien](#) im AWS Organizations -Benutzerhandbuch.
- Richtlinien zur Ressourcenkontrolle (RCPs) — Legen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten fest. Weitere Informationen finden Sie im AWS Organizations Benutzerhandbuch unter [Richtlinien zur Ressourcenkontrolle \(RCPs\)](#).
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die als Parameter übergeben werden, wenn Sie eine temporäre Sitzung für eine Rolle oder einen Verbundbenutzer erstellen. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

Mehrere Richtlinientypen

Wenn für eine Anfrage mehrere Arten von Richtlinien gelten, sind die daraus resultierenden Berechtigungen schwieriger zu verstehen. Informationen darüber, wie AWS bestimmt wird, ob eine

Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie unter [Bewertungslogik für Richtlinien](#) im IAM-Benutzerhandbuch.

Wie arbeitet ELB mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf ELB verwenden, sollten Sie sich darüber informieren, welche IAM-Funktionen für ELB verfügbar sind.

IAM-Funktionen, die Sie mit ELB verwenden können

IAM-Feature	ELB-Unterstützung
Identitätsbasierte Richtlinien	Ja
Ressourcenbasierte Richtlinien	Nein
Richtlinienaktionen	Ja
Richtlinienressourcen	Ja
Richtlinienbedingungsschlüssel (services spezifisch)	Ja
ACLs	Nein
ABAC (Tags in Richtlinien)	Ja
Temporäre Anmeldeinformationen	Ja
Prinzipalberechtigungen	Ja
Servicerollen	Nein
Serviceverknüpfte Rollen	Ja

Identitätsbasierte Richtlinien für ELB

Unterstützt Richtlinien auf Identitätsbasis: Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern,

welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

Ressourcenbasierte Richtlinien innerhalb von ELB

Unterstützt ressourcenbasierte Richtlinien: Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Weitere Informationen finden Sie unter [Kontoübergreifender Ressourcenzugriff in IAM](#) im IAM-Benutzerhandbuch.

Politische Maßnahmen für ELB

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Nehmen Sie Aktionen in eine Richtlinie auf, um Berechtigungen zur Ausführung des zugehörigen Vorgangs zu erteilen.

Eine Liste der [ELB-Aktionen finden Sie in der Serviceauthorisierungsreferenz unter Von ELB V2 definierte Aktionen und Von ELB V1](#) definierte Aktionen.

Richtlinienaktionen in ELB verwenden vor der Aktion das folgende Präfix:

```
elasticloadbalancing
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

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

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort Describe beginnen, einschließlich der folgenden Aktion:

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

Die vollständige Liste der API-Aktionen für ELB finden Sie in der folgenden Dokumentation:

- Application Load Balancer, Network Load Balancer und Gateway Load Balancer – [API-Referenz Version 2015-12-01](#)
- Classic Load Balancer – [API-Referenz Version 2012-06-01](#)

Politische Ressourcen für ELB

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Als Best Practice geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Einige ELB-API-Aktionen unterstützen mehrere Ressourcen. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie sie ARNs durch Kommas.

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

Eine Liste der ELB-Ressourcentypen und ihrer ARNs Eigenschaften finden Sie unter [Ressourcen definiert durch ELB V2](#) und [Ressourcen definiert durch ELB V1](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter [Von ELB V2 definierte Aktionen](#) und [Von ELB V1 definierte Aktionen](#).

Schlüssel für Richtlinienbedingungen für ELB

Unterstützt servicespezifische Richtlinienbedingungsschlüssel: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

Das Element Condition gibt an, wann Anweisungen auf der Grundlage definierter Kriterien ausgeführt werden. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Eine Liste der ELB-Bedingungsschlüssel finden Sie unter [Bedingungsschlüssel für ELB V2](#) und [Bedingungsschlüssel für ELB V1](#) in der Service Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Von ELB V2 definierte Aktionen](#) und [Von ELB V1 definierte Aktionen](#).

Bedingungsschlüssel

- [elasticloadbalancing:ListenerProtocol-Bedingungsschlüssel](#)
- [elasticloadbalancing:SecurityPolicy-Bedingungsschlüssel](#)
- [elasticloadbalancing:Scheme-Bedingungsschlüssel](#)
- [elasticloadbalancing:SecurityGroup-Bedingungsschlüssel](#)

- [elasticloadbalancing:Subnet-Bedingungsschlüssel](#)
- [elasticloadbalancing:ResourceTag-Bedingungsschlüssel](#)

elasticloadbalancing:ListenerProtocol-Bedingungsschlüssel

Der `elasticloadbalancing:ListenerProtocol` Bedingungsschlüssel kann für Bedingungen verwendet werden, die die Typen von Listenern definieren, die erstellt und verwendet werden können. Die Richtlinie ist für Application Load Balancers, Network Load Balancers und Classic Load Balancers verfügbar. Die folgenden Aktionen unterstützen diesen Bedingungsschlüssel:

API Version 2015-12-01

- `CreateListener`
- `ModifyListener`

API-Version 2012-06-01

- `CreateLoadBalancer`
- `CreateLoadBalancerListeners`

In der folgenden Beispielrichtlinie müssen Benutzer das HTTPS-Protokoll für die Listener für ihre Application Load Balancer und das TLS-Protokoll für die Listener für ihre Network Load Balancer auswählen.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": {  
    "Effect": "Allow",  
    "Action": [  
      "elasticloadbalancing:CreateListener",  
      "elasticloadbalancing:ModifyListener"  
    ],  
    "Resource": "*",  
    "Condition": {  
      "ForAnyValue:StringEquals": {  
        "elasticloadbalancing:ListenerProtocol": [  
          "https",  
          "TLS"  
        ]  
      }  
    }  
  }  
}
```

```
        "HTTPS",
        "TLS"
    ]
}
}
```

Mit einem Classic Load Balancer können Sie mehrere Listener in einem einzigen Aufruf angeben. Daher muss Ihre Richtlinie einen [Kontextschlüssel mit mehreren Werten](#) verwenden, wie im folgenden Beispiel gezeigt.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:CreateLoadBalancerListeners"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "elasticloadbalancing:ListenerProtocol": [
                        "TCP",
                        "HTTP",
                        "HTTPS"
                    ]
                }
            }
        ]
    }
}
```

elasticloadbalancing:SecurityPolicy-Bedingungsschlüssel

Der `elasticloadbalancing:SecurityPolicy` Bedingungsschlüssel kann für Bedingungen verwendet werden, die bestimmte Sicherheitsrichtlinien für die Load Balancer definieren und durchsetzen. Die Richtlinie ist für Application Load Balancers, Network Load Balancers und Classic Load Balancers verfügbar. Die folgenden Aktionen unterstützen diesen Bedingungsschlüssel:

API Version 2015-12-01

- `CreateListener`
- `ModifyListener`

API-Version 2012-06-01

- `CreateLoadBalancerPolicy`
- `SetLoadBalancerPoliciesOfListener`

In der folgenden Beispielrichtlinie müssen Benutzer eine der angegebenen Sicherheitsrichtlinien für ihre Application Load Balancer und Network Load Balancer auswählen.

JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": {  
    "Effect": "Allow",  
    "Action": [  
      "elasticloadbalancing:CreateListener",  
      "elasticloadbalancing:ModifyListener"  
    ],  
    "Resource": "*",  
    "Condition": {  
      "ForAnyValue:StringEquals": {  
        "elasticloadbalancing:SecurityPolicy": [  
          "ELBSecurityPolicy-TLS13-1-2-2021-06",  
          "ELBSecurityPolicy-TLS13-1-2-Res-2021-06",  
          "ELBSecurityPolicy-TLS13-1-1-2021-06"  
        ]  
      }  
    }  
  }  
}
```

```
    }  
}
```

elasticloadbalancing:Scheme-Bedingungsschlüssel

Der `elasticloadbalancing:Scheme` Bedingungsschlüssel kann für Bedingungen verwendet werden, die definieren, welches Schema bei der Erstellung des Load Balancers ausgewählt werden kann. Die Richtlinie ist für Application Load Balancers, Network Load Balancers und Classic Load Balancers verfügbar. Die folgenden Aktionen unterstützen diesen Bedingungsschlüssel:

API Version 2015-12-01

- `CreateLoadBalancer`

API-Version 2012-06-01

- `CreateLoadBalancer`

In der folgenden Beispielrichtlinie müssen Benutzer das angegebene Schema für ihre Load Balancer auswählen.

JSON

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": "elasticloadbalancing:CreateLoadBalancer",  
        "Resource": "*",  
        "Condition": {  
            "StringEquals": {  
                "elasticloadbalancing:Scheme": "internal"  
            }  
        }  
    }  
}
```

elasticloadbalancing:SecurityGroup-Bedingungsschlüssel

Important

ELB akzeptiert alle Großbuchstaben der Sicherheitsgruppe. IDs Achten Sie jedoch beispielsweise darauf, die entsprechenden Bedingungsoperatoren ohne Berücksichtigung der Groß- und Kleinschreibung zu verwenden. `StringEqualsIgnoreCase`

Der `elasticloadbalancing:SecurityGroup` Bedingungsschlüssel kann für Bedingungen verwendet werden, die definieren, welche Sicherheitsgruppen auf die Load Balancer angewendet werden können. Die Richtlinie ist für Application Load Balancers, Network Load Balancers und Classic Load Balancers verfügbar. Die folgenden Aktionen unterstützen diesen Bedingungsschlüssel:

API Version 2015-12-01

- `CreateLoadBalancer`
- `SetSecurityGroups`

API-Version 2012-06-01

- `CreateLoadBalancer`
- `ApplySecurityGroupsToLoadBalancer`

In der folgenden Beispielrichtlinie müssen Benutzer eine der angegebenen Sicherheitsgruppen für ihre Load Balancer auswählen.

```
"Version": "2012-10-17",
"Statement": [
    "Effect": "Allow",
    "Action": [
        "elasticloadbalancing:CreateLoadBalancer",
        "elasticloadbalancing:SetSecurityGroup"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEqualsIgnoreCase":{
            "elasticloadbalancing:SecurityGroup": [
```

```
        "sg-51530134",
        "sg-51530144",
        "sg-51530139"
    ],
},
}
}
```

elasticloadbalancing:Subnet-Bedingungsschlüssel

Important

ELB akzeptiert alle Großbuchstaben des Subnetzes. IDs Achten Sie jedoch beispielsweise darauf, die entsprechenden Bedingungsoperatoren ohne Berücksichtigung der Groß- und Kleinschreibung zu verwenden. `StringEqualsIgnoreCase`

Der `elasticloadbalancing:Subnet` Bedingungsschlüssel kann für Bedingungen verwendet werden, die definieren, welche Subnetze erstellt und an Load Balancer angehängt werden können. Die Richtlinie ist für Application Load Balancer, Network Load Balancer, Gateway Load Balancer und Classic Load Balancer verfügbar. Die folgenden Aktionen unterstützen diesen Bedingungsschlüssel:

API Version 2015-12-01

- `CreateLoadBalancer`
- `SetSubnets`

API-Version 2012-06-01

- `CreateLoadBalancer`
- `AttachLoadBalancerToSubnets`

In der folgenden Beispielrichtlinie müssen Benutzer eines der angegebenen Subnetze für ihre Load Balancer auswählen.

JSON

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "elasticloadbalancing:CreateLoadBalancer",  
            "elasticloadbalancing:SetSubnets"  
        ],  
        "Resource": "*",  
        "Condition": {  
            "ForAnyValue:StringEqualsIgnoreCase": {  
                "elasticloadbalancing:Subnet": [  
                    "subnet-01234567890abcdef",  
                    "subnet-01234567890abcdg"  
                ]  
            }  
        }  
    }  
}
```

elasticloadbalancing:ResourceTag-Bedingungsschlüssel

Der **key** Bedingungsschlüsselelasticloadbalancing:ResourceTag/ist ELB-spezifisch. Alle mutierenden Aktionen unterstützen diesen Bedingungsschlüssel.

ACLs in ELB

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

ABAC mit ELB

Unterstützt ABAC (Tags in Richtlinien): Ja

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen, auch als Tags bezeichnet, definiert werden. Sie können Tags an IAM-

Entitäten und AWS -Ressourcen anhängen und dann ABAC-Richtlinien entwerfen, um Operationen zu ermöglichen, wenn das Tag des Prinzipals mit dem Tag auf der Ressource übereinstimmt.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Definieren von Berechtigungen mit ABAC-Autorisierung](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

Verwenden temporärer Anmeldeinformationen mit ELB

Unterstützt temporäre Anmeldeinformationen: Ja

Temporäre Anmeldeinformationen ermöglichen kurzfristigen Zugriff auf AWS Ressourcen und werden automatisch erstellt, wenn Sie einen Verbund verwenden oder die Rollen wechseln. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Anmeldeinformationen in IAM und AWS-Services , die mit IAM funktionieren](#) im IAM-Benutzerhandbuch.

Serviceübergreifende Prinzipalberechtigungen für ELB

Unterstützt Forward Access Sessions (FAS): Ja

Forward Access Sessions (FAS) verwenden die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. Einzelheiten zu den Richtlinien für FAS-Anforderungen finden Sie unter [Zugriffssitzungen weiterleiten](#).

Servicerollen für ELB

Unterstützt Servicerollen: Nein

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

Servicebezogene Rollen für ELB

Unterstützt serviceverknüpfte Rollen: Ja

Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zum Erstellen oder Verwalten von dienstbezogenen ELB-Rollen finden Sie unter.
[Serviceverknüpfte ELB-Rolle](#)

ELB-API-Berechtigungen zum Markieren von Ressourcen während der Erstellung

Damit Benutzer Ressourcen während der Erstellung taggen können, benötigen sie die Berechtigung zum Verwenden der Aktion, mit der die Ressource erstellt wird, z. B. `elasticloadbalancing:CreateLoadBalancer` oder `elasticloadbalancing:CreateTargetGroup`. Wenn Tags in der Aktion angegeben werden, mit der die Ressource erstellt wird, ist eine zusätzliche Autorisierung für die `elasticloadbalancing:AddTags`-Aktion erforderlich, um die Berechtigungen der Benutzer zum Taggen der erstellten Ressourcen zu überprüfen. Daher benötigen die Benutzer außerdem die expliziten Berechtigungen zum Verwenden der `elasticloadbalancing:AddTags`-Aktion.

Verwenden Sie in der IAM-Richtliniendefinition für die `elasticloadbalancing:AddTags`-Aktion das `Condition`-Element mit dem `elasticloadbalancing:CreateAction`-Bedingungsschlüssel, um der Aktion, die die Ressource erstellt, Berechtigungen fürs Tagging zu erteilen.

Das folgende Beispiel veranschaulicht eine Richtlinie, die es Benutzern ermöglicht, Zielgruppen zu erstellen und ihnen bei der Erstellung beliebige Tags zuzuweisen. Das Tagging von bestehenden Ressourcen durch die Benutzer ist nicht zulässig. (Sie können die `elasticloadbalancing:AddTags`-Aktion nicht direkt aufrufen.)

JSON

```
{  
  "Version": "2012-10-17",
```

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

In ähnlicher Weise erlaubt die folgende Richtlinie den Benutzern, einen Load Balancer zu erstellen und während der Erstellung Tags anzuwenden. Das Tagging von bestehenden Ressourcen durch die Benutzer ist nicht zulässig. (Sie können die elasticloadbalancing:AddTags-Aktion nicht direkt aufrufen.)

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:CreateLoadBalancer"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:AddTags"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "elasticloadbalancing:CreateAction" : "CreateLoadBalancer"
        }
      }
    }
]
```

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

Die `elasticloadbalancing:AddTags`-Aktion wird nur ausgewertet, wenn die Tags während der Aktion zur Ressourcenerstellung angewendet werden. Folglich benötigt ein Benutzer, der über die Berechtigungen zum Erstellen einer Ressource verfügt (vorausgesetzt, es bestehen keine Markierungsbedingungen), keine Berechtigungen zur Verwendung der `elasticloadbalancing:AddTags`-Aktion, wenn keine Tags in der Anforderung angegeben werden. Wenn der Benutzer allerdings versucht, eine Ressource mit Tags zu erstellen, schlägt die Anforderung fehl, wenn der Benutzer nicht über die Berechtigungen für die `elasticloadbalancing:AddTags`-Aktion verfügt.

Serviceverknüpfte ELB-Rolle

ELB verwendet eine dienstbezogene Rolle für die Berechtigungen, die erforderlich sind, um andere AWS Dienste in Ihrem Namen aufzurufen. Weitere Informationen finden Sie unter [Serviceverknüpfte Rollen](#) im IAM-Benutzerhandbuch.

Von der serviceverknüpften Rolle erteilte Berechtigungen

ELB verwendet die angegebene dienstbezogene Rolle `AWSServiceRoleForElasticLoadBalancing`, um in Ihrem Namen andere AWS Dienste aufzurufen.

`AWSServiceRoleForElasticLoadBalancing` vertraut darauf, dass der `elasticloadbalancing.amazonaws.com` Dienst die Rolle übernimmt.

Die Richtlinie für Rollenberechtigungen lautet `AWSElasticLoadBalancingServiceRolePolicy`. Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [AWSElasticLoadBalancingServiceRolePolicy](#) in der Referenz zu von AWS verwalteten Richtlinien.

Erstellen der serviceverknüpften Rolle

Sie müssen die Rolle `AWSServiceRoleForElasticLoadBalancing` nicht manuell erstellen. ELB erstellt diese Rolle für Sie, wenn Sie einen Load Balancer oder eine Zielgruppe erstellen.

Damit ELB in Ihrem Namen eine dienstbezogene Rolle erstellen kann, müssen Sie über die erforderlichen Berechtigungen verfügen. Weitere Informationen finden Sie unter [Serviceverknüpfte Rollenberechtigungen](#) im IAM-Benutzerhandbuch.

Bearbeiten der serviceverknüpften Rolle

Sie können die Beschreibung der `AWSServiceRoleForElasticLoadBalancing` Verwendung von IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rollenbeschreibung](#) im IAM-Benutzerhandbuch.

Löschen der serviceverknüpften Rolle

Wenn Sie ELB nicht mehr verwenden müssen, empfehlen wir Ihnen, es zu löschen `AWSServiceRoleForElasticLoadBalancing`.

Sie können diese dienstbezogene Rolle erst löschen, nachdem Sie alle Load Balancer in Ihrem AWS Konto gelöscht haben. Auf diese Weise wird sichergestellt, dass Sie nicht versehentlich die Berechtigung für den Zugriff auf Ihre Load Balancers entfernen. Weitere Informationen finden Sie unter [Löschen eines Application Load Balancers](#), [Löschen eines Network Load Balancers](#) und [Löschen eines Classic Load Balancers](#).

Sie können die IAM-Konsole, die IAM-CLI oder die IAM-API verwenden, um serviceverknüpfte Rollen zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Nach dem Löschen erstellt ELB die Rolle erneut `AWSServiceRoleForElasticLoadBalancing`, wenn Sie einen Load Balancer erstellen.

AWS verwaltete Richtlinien für ELB

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet wird AWS. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Denken Sie daran, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie für alle AWS Kunden verfügbar sind. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie [vom Kunden verwaltete Richtlinien](#) definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzipalitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS-Service wird oder neue API-Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie unter [Von AWS verwaltete Richtlinien](#) im IAM-Benutzerhandbuch.

AWS verwaltete Richtlinie: AWSElasticLoadBalancingClassicServiceRolePolicy

Diese Richtlinie umfasst alle Berechtigungen, die ELB (Classic Load Balancer) benötigt, um in Ihrem Namen andere AWS Dienste aufzurufen. Serviceverknüpfte Rollen sind vordefiniert. Bei vordefinierten Rollen müssen Sie die erforderlichen Berechtigungen nicht manuell hinzufügen, damit ELB Aktionen in Ihrem Namen ausführen kann. Sie können diese Richtlinie nicht anhängen, trennen, ändern oder löschen.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [AWSElasticLoadBalancingClassicServiceRolePolicy](#) in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: AWSElasticLoadBalancingServiceRolePolicy

Diese Richtlinie umfasst alle Berechtigungen, die ELB benötigt, um in Ihrem Namen andere AWS Dienste anzurufen. Serviceverknüpfte Rollen sind vordefiniert. Bei vordefinierten Rollen müssen Sie die erforderlichen Berechtigungen nicht manuell hinzufügen, damit ELB Aktionen in Ihrem Namen ausführen kann. Sie können diese Richtlinie nicht anhängen, trennen, ändern oder löschen.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [AWSElasticLoadBalancingServiceRolePolicy](#) in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: ElasticLoadBalancingFullAccess

Diese Richtlinie gewährt vollen Zugriff auf den ELB-Dienst und eingeschränkten Zugriff auf andere Dienste über die AWS Managementkonsole.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [ElasticLoadBalancingFullAccess](#) in der Referenz zu von AWS verwalteten Richtlinien.

AWS verwaltete Richtlinie: ElasticLoadBalancingReadOnly

Diese Richtlinie bietet schreibgeschützten Zugriff auf ELB und abhängige Dienste.

Informationen zu den Berechtigungen für diese Richtlinie finden Sie unter [ElasticLoadBalancingReadOnly](#) in der Referenz zu von AWS verwalteten Richtlinien.

ELB-Aktualisierungen verwalteter Richtlinien AWS

Hier finden Sie Informationen zu Aktualisierungen der AWS verwalteten Richtlinien für ELB, seit dieser Dienst begonnen hat, diese Änderungen zu verfolgen.

Änderungen	Beschreibung	Datum
AWSElasticLoadBalancingServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Die <code>ec2:AllocateIpamPoolCidr</code> Aktion zur Erteilung von Berechtigungen zur Zuweisung von CIDR-Blöcken aus IPAM-Pools wurde hinzugefügt.	17. Februar 2025
ElasticLoadBalancingFullAccess – Aktualisierung auf eine bestehende Richtlinie	Die <code>arc-zonal-shift:*</code> Aktionen zum Erteilen von Berechtigungen, die für Zonal Shift erforderlich sind, wurden hinzugefügt.	28. November 2023
ElasticLoadBalancingReadOnly – Aktualisierung auf eine bestehende Richtlinie	Die folgenden Aktionen wurden hinzugefügt, um die für die Zonenschicht erforderlichen Berechtigungen zu gewähren: <code>arc-zonal-shift:GetManagedResource</code> , und <code>arc-zonal-shift>ListManagedResources</code> . <code>arc-zonal-shift>ListZonalShifts</code>	28. November 2023
AWSElasticLoadBalancingServiceRolePolicy – Aktualisierung auf eine bestehende Richtlinie	Die <code>ec2:DescribeVpcPeeringConnections</code> Aktion zum Erteilen von Berechtigungen, die für Peering-Verbindungen erforderlich sind, wurde hinzugefügt.	11. Oktober 2021

Änderungen	Beschreibung	Datum
<u>ElasticLoadBalancingFullAccess</u> – Aktualisierung auf eine bestehende Richtlinie	Die ec2:DescribeVpcPeeringConnections Aktion zum Erteilen von Berechtigungen, die für Peering-Verbindungen erforderlich sind, wurde hinzugefügt.	11. Oktober 2021
<u>ElasticLoadBalancingFullAccess</u> – Neue Richtlinie	Bietet vollen Zugriff auf ELB und abhängige Dienste.	20. September 2018
<u>ElasticLoadBalancingReadOnly</u> – Neue Richtlinie	Bietet schreibgeschützten Zugriff auf ELB und abhängige Dienste.	20. September 2018
ELB hat begonnen, Änderungen zu verfolgen	ELB begann, Änderungen an seinen AWS verwalteten Richtlinien nachzuverfolgen.	20. September 2018

Compliance-Validierung für Elastic Load Balancing

Informationen darüber, ob AWS-Service ein [AWS-Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS-Services unter](#) . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter [AWS Compliance-Programme AWS](#) .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte herunterladen unter](#) .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. Weitere Informationen zu Ihrer Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS-Services finden Sie in der [AWS Sicherheitsdokumentation](#).

Ausfallsicherheit beim Elastic Load Balancing

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch

Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

Zusätzlich zur AWS globalen Infrastruktur bietet ELB die folgenden Funktionen zur Unterstützung Ihrer Datenstabilität:

- Verteilung des eingehenden Datenverkehrs auf mehrere Instances in einer einzigen Availability Zone oder mehreren Availability Zones.
- Sie können es AWS Global Accelerator zusammen mit Ihren Application Load Balancers verwenden, um eingehenden Datenverkehr auf mehrere Load Balancer in einer oder mehreren Regionen zu verteilen. AWS Weitere Informationen finden Sie im [AWS Global Accelerator - Entwicklerhandbuch](#).
- Amazon ECS ermöglicht es Ihnen, Docker-Container in einem Cluster von EC2 Instances auszuführen, zu stoppen und zu verwalten. Sie können Ihren Amazon ECS-Service so konfigurieren, dass er einen Load Balancer verwendet, um eingehenden Datenverkehr über die Services in einem Cluster zu verteilen. Weitere Informationen finden Sie im [Amazon Elastic Container Service-Entwicklerhandbuch](#).

Infrastruktursicherheit in Elastic Load Balancing

Als verwalteter Dienst ist ELB durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf ELB zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Netzwerkisolierung

Eine Virtual Private Cloud (VPC) ist ein virtuelles Netzwerk in Ihrem eigenen logisch isolierten Bereich in der AWS Cloud. Ein Subnetz ist ein Bereich von IP-Adressen in einer VPC. Wenn Sie einen Load Balancer erstellen, können Sie mindestens ein Subnetz für die Load Balancer-Knoten angeben. Sie können EC2 Instances in den Subnetzen Ihrer VPC bereitstellen und sie bei Ihrem Load Balancer registrieren. Weitere Informationen über VPC und Subnetze finden Sie im [Amazon VPC-Benutzerhandbuch](#).

Wenn Sie einen Load Balancer in einer VPC erstellen, kann er entweder mit dem Internet verbunden oder intern sein. Ein interner Load Balancer kann nur Anforderungen von Clients mit Zugriff auf die VPC für den Load Balancer weiterleiten.

Ihr Load Balancer sendet Anfragen über private IP-Adressen an seine registrierten Ziele. Daher benötigen Ihre Ziele keine öffentlichen IP-Adressen, um Anfragen von einem Load Balancer zu empfangen.

Um die ELB-API von Ihrer VPC aus mit privaten IP-Adressen aufzurufen, verwenden Sie AWS PrivateLink. Weitere Informationen finden Sie unter [Greifen Sie über einen Schnittstellenendpunkt \(AWS PrivateLink\) auf ELB zu](#).

Steuern des Netzwerkverkehrs

Berücksichtigen Sie die folgenden Optionen zum Sichern des Netzwerkverkehrs, wenn Sie einen Load Balancer verwenden:

- Verwenden Sie sichere Listener, um die verschlüsselte Kommunikation zwischen Clients und Ihren Load Balancern zu unterstützen. Application Load Balancer unterstützen HTTPS-Listener. Network Load Balancer unterstützen TLS-Listener. Classic Load Balancer unterstützen sowohl HTTPS- als auch TLS-Listener. Sie können aus vordefinierten Sicherheitsrichtlinien für Ihren Load Balancer wählen, um die Verschlüsselungssammlungen und Protokollversionen anzugeben, die von Ihrer Anwendung unterstützt werden. Sie können AWS Certificate Manager (ACM) oder AWS Identity and Access Management (IAM) verwenden, um die auf Ihrem Load Balancer installierten Serverzertifikate zu verwalten. Sie können das SNI-Protokoll (Server Name Induation) verwenden, um mehrere sichere Websites mit einem einzigen sicheren Listener bereitzustellen. SNI wird automatisch für Ihren Load Balancer aktiviert, wenn Sie mehr als ein Serverzertifikat mit einem sicheren Listener verknüpfen.
- Konfigurieren Sie die Sicherheitsgruppen für Ihre Application Load Balancer und Classic Load Balancer, um nur Datenverkehr von bestimmten Clients anzunehmen. Diese Sicherheitsgruppen

müssen eingehenden Datenverkehr von Clients an die Listener-Ports und ausgehenden Datenverkehr zu den Clients zulassen.

- Konfigurieren Sie die Sicherheitsgruppen für Ihre EC2 Amazon-Instances so, dass sie nur Traffic vom Load Balancer akzeptieren. Diese Sicherheitsgruppen müssen eingehenden Datenverkehr vom Load Balancer an die Listener-Ports und die Zustandsprüfung Ports zulassen.
- Konfigurieren Sie Ihren Application Load Balancer, um Benutzer über einen Identitätsanbieter oder über Unternehmensidentitäten sicher zu authentifizieren. Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mithilfe eines Application Load Balancer](#).
- Sie können [AWS WAF](#) mit Ihren Application Load Balancern verwenden, um Anforderungen basierend auf den Regeln in einer Web-ACL (Web-Zugriffskontrollliste) zu erlauben oder zu blockieren.

Greifen Sie über einen Schnittstellenendpunkt (AWS PrivateLink) auf ELB zu

Sie können eine private Verbindung zwischen Ihrer Virtual Private Cloud (VPC) und der ELB-API herstellen, indem Sie einen VPC-Schnittstellen-Endpunkt erstellen. Sie können diese Verbindung verwenden, um die ELB-API von Ihrer VPC aus aufzurufen, ohne dass Sie ein Internet-Gateway, eine NAT-Instance oder eine VPN-Verbindung mit Ihrer VPC verbinden müssen. Der Endpunkt bietet zuverlässige, skalierbare Konnektivität zur ELB-API, Versionen 2015-12-01 und 2012-06-01, mit der Sie Ihre Load Balancer erstellen und verwalten.

Schnittstelle, mit der VPC-Endpunkte betrieben werden AWS PrivateLink, eine Funktion, die die Kommunikation zwischen Ihren Anwendungen und die AWS-Services Verwendung privater IP-Adressen ermöglicht. Weitere Informationen finden Sie unter [AWS PrivateLink](#).

Limit

AWS PrivateLink unterstützt keine Network Load Balancer mit mehr als 50 Listenern.

Erstellen Sie einen Schnittstellenendpunkt für ELB

Erstellen Sie einen Endpunkt für ELB mit dem folgenden Dienstnamen:

com.amazonaws.*region*.elasticloadbalancing

Weitere Informationen finden Sie unter [Erstellen eines Schnittstellenendpunkts](#) im AWS PrivateLink - Leitfaden.

Erstellen Sie eine VPC-Endpunktrichtlinie für ELB

Sie können Ihrem VPC-Endpunkt eine Richtlinie hinzufügen, um den Zugriff auf die ELB-API zu kontrollieren. Die Richtlinie legt Folgendes fest:

- Prinzipal, der die Aktionen ausführen kann.
- Die Aktionen, die ausgeführt werden können.
- Die Ressource, auf der die Aktionen ausgeführt werden können.

Das folgende Beispiel zeigt eine VPC-Endpunktrichtlinie, die jedem die Berechtigung zum Erstellen eines Load Balancer über den Endpunkt verweigert. Die Beispielrichtlinie gewährt auch jedem die Berechtigung, alle anderen Aktionen auszuführen.

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

Weitere Informationen finden Sie unter [Steuern des Zugriffs auf Services mit Endpunktrichtlinien](#) im AWS PrivateLink -Leitfaden.

Drosselung von Anfragen für die ELB-API

ELB drosselt seine API-Anfragen für jedes AWS Konto pro Region. Wir tun dies, um die Leistung und Verfügbarkeit des Dienstes zu verbessern. Durch die Drosselung wird sichergestellt, dass Anfragen an die ELB-API die maximal zulässigen API-Anforderungsgrenzen nicht überschreiten. API-Anfragen unterliegen den Anforderungsbegrenkungen, unabhängig davon, ob Sie sie aufrufen oder ob sie in Ihrem Namen aufgerufen werden (z. B. von der Anwendung AWS-Managementkonsole oder einer Drittanbieteranwendung).

Wenn Sie ein ELB-API-Drosselungslimit überschreiten, erhalten Sie den `ThrottlingException` Fehlercode und eine `Rate exceeded` Fehlermeldung.

Wir empfehlen Ihnen, sich darauf vorzubereiten, vorsichtig mit der Drosselung umzugehen. Weitere Informationen finden Sie unter [Timeouts, Wiederholungen und Backoff mit Jitter](#). Wenn Sie ein hohes Maß an Drosselung feststellen, können Sie sich an uns wenden, um Ihnen bei der Bewertung Ihrer API-Nutzung und möglicher Lösungen AWS Support zu helfen. Jeder Fall wird einzeln bewertet. Support kann Ihre Grenzwerte innerhalb der Sicherheitsgrenzen des Systems erhöhen, um eine hohe Verfügbarkeit und vorhersehbare Leistung aufrechtzuerhalten.

Wie wird die Drosselung angewendet

ELB verwendet den [Token-Bucket-Algorithmus, um die API-Drosselung](#) zu implementieren. Mit diesem Algorithmus verfügt Ihr Konto über einen Bucket, der eine bestimmte Anzahl von Token enthält. Die Anzahl der Token im Bucket entspricht Ihrem Drosselungslimit für eine bestimmte Sekunde.

ELB bietet zwei Gruppen von API-Aktionen. ELB API Version 2 unterstützt die folgenden Typen von Load Balancers: Application Load Balancer, Network Load Balancer und Gateway Load Balancer. ELB API-Version 1 unterstützt Classic Load Balancers. Jede ELB-API-Version hat ihre eigenen Buckets und Tokens.

Dienste, die die ELB-API in Ihrem Namen aufrufen, wie Amazon, Amazon ECS EC2, Amazon EC2 Auto Scaling, und AWS CloudFormation haben ihre eigenen Buckets auf Kontoebene. Diese Dienste verbrauchen keine Token aus Ihren Buckets.

Anforderungsratenbegrenzung

Durch die Begrenzung der Anforderungsraten wird die Anzahl der API-Anfragen, die Sie stellen, eingeschränkt. Jede Anforderung, die Sie stellen, entfernt ein Token aus dem Bucket. Beispielsweise beträgt die Token-Bucket-Größe für API-Aktionen, die nicht mutieren, 40 Token. Sie können in einer Sekunde bis zu 40 `Describe*` Anfragen stellen. Wenn Sie in einer Sekunde mehr als 40 `Describe*` Anfragen haben, werden Sie gedrosselt und die verbleibenden Anfragen innerhalb dieser Sekunde schlagen fehl.

Eimer werden automatisch mit einer festgelegten Geschwindigkeit wieder aufgefüllt. Wenn ein Bucket seine maximale Kapazität unterschreitet, wird jede Sekunde eine festgelegte Anzahl von Tokens hinzugefügt, bis der Bucket seine maximale Kapazität erreicht hat. Wenn ein Eimer voll ist, wenn Nachfüll-Token eintreffen, werden sie verworfen. Ein Bucket kann nicht mehr als die maximale Anzahl an Tokens aufnehmen. Beispielsweise beträgt die Bucket-Größe für API-Aktionen, die nicht mutieren, 40 Token und die Nachfüllrate beträgt 10 Token pro Sekunde. Wenn Sie 40 `DescribeLoadBalancers` Anfragen in einer Sekunde stellen, wird der Bucket auf null (0) Token reduziert. Wir fügen dem Bucket jede Sekunde 10 Refill-Token hinzu, bis er seine maximale Kapazität von 40 Tokens erreicht hat. Das bedeutet, dass es 4 Sekunden dauert, bis ein leerer Bucket seine maximale Kapazität erreicht hat, wenn während dieser Zeit keine Anfragen gestellt werden.

Sie müssen nicht warten, bis ein Bucket vollständig gefüllt ist, bevor Sie API-Anfragen stellen können. Sie können Token verwenden, wenn sie einem Bucket hinzugefügt werden. Wenn Sie die Nachfüll-Token sofort verwenden, erreicht der Eimer nicht seine maximale Kapazität.

Es gibt ein Drosselungslimit auf Kontoebene, das für alle ELB-API-Aktionen gilt. Die Kapazität des Buckets auf Kontoebene beträgt 40 Token und die Nachfüllrate beträgt 10 Anforderungstoken pro Sekunde.

Größen und Nachfüllraten für Token-Buckets anfragen

Zur Begrenzung der Anforderungsraten sind API-Aktionen in Kategorien unterteilt. Jede Kategorie hat ihre eigenen Grenzen.

Kategorien

- Mutierende Aktionen — API-Aktionen, die Ressourcen erstellen, ändern oder löschen. Diese Kategorie umfasst im Allgemeinen alle API-Aktionen, die nicht als nicht mutierende Aktionen eingestuft werden. Für diese Aktionen gilt eine niedrigere Drosselungsgrenze als für nicht mutierende API-Aktionen.

- Nicht mutierende Aktionen — API-Aktionen, die Daten über Ressourcen abrufen. Diese API-Aktionen haben in der Regel die höchsten API-Drosselungsgrenzen.
- Ressourcenintensive Aktionen — API-Aktionen, deren Ausführung am meisten Zeit in Anspruch nimmt und die meisten Ressourcen beansprucht. Für diese Aktionen gilt eine noch niedrigere Drosselungsgrenze als für mutierende Aktionen. Diese Aktionen werden getrennt von anderen mutierenden Aktionen gedrosselt.
- Registrierungsaktionen — API-Aktionen, die Ziele registrieren oder abmelden. Diese API-Aktionen werden getrennt von anderen mutierenden Aktionen gedrosselt.
- Nicht kategorisierte Aktionen — Diese API-Aktionen erhalten ihre eigenen Token-Bucket-Größen und Wiederauffüllraten, obwohl sie unter eine der anderen Kategorien fallen.

Die folgende Tabelle zeigt die Standardkapazität und die Nachfüllraten für die kategorisierten Request-Token-Buckets.

Kategorie	ELBv2 Aktionen	ELBv1 Aktionen	Kapazität des Eimers	Nachfüllrate (pro Sekunde)
Ressourcenintensiv	CreateLoadBalancer , SetSubnets	CreateLoadBalancer , AttachLoadBalancerToSubnets , DetachLoadBalancerFromSubnets , EnableAvailabilityZonesForLoadBalancer , DisableAvailabilityZonesForLoadBalancer	10	0,2 †
Registrierung	RegisterTargets , DeregisterTargets	RegisterInstancesWithLoadBalancer , DeregisterInstancesFromLoadBalancer	20	4
Nicht mutierend	DescribeAccountLimits , DescribeC	Describe*	40	10

Kategorie	ELBv2 Aktionen	ELBv1 Aktionen	Kapazität des Eimers	Nachfüllrate (pro Sekunde)
	<code>CapacityReservation</code> <code>, DescribeListenerAttributes</code> <code>, DescribeListenerCertificate</code> <code>, DescribeListeners</code> <code>, DescribeLoadBalancerAttributes</code> <code>, DescribeLoadBalancers</code> <code>, DescribeRules</code> <code>, DescribeSSLPolicies</code> <code>, DescribeTags</code> <code>, DescribeTargetGroupAttributes</code> <code>, DescribeTargetGroups</code> <code>, DescribeTargetHealth</code>			

Kategorie	ELBv2 Aktionen	ELBv1 Aktionen	Kapazität des Eimers	Nachfüllrate (pro Sekunde)
Mutierend	AddListenerCertificates, AddTags, CreateListener, CreateRule, CreateTargetGroup, DeleteListener, DeleteLoadBalancer, DeleteRule, DeleteTargetGroup, ModifyCapacityReservation, ModifyIpPools, ModifyListener, ModifyListenerAttributes, ModifyLoadBalancerAttribute, ModifyRule, ModifyTargetGroup, ModifyTargetGroupAttributes, RemoveListenerCertificates, RemoveTags, SetIpAddressType, SetRulePriorities, SetSecurityGroups	AddTags, ApplySecurityGroupsToLoadBalancer, ConfigureHealthCheck, CreateAppCookieStickinessPolicy, CreateLbCookieStickinessPolicy, CreateLoadBalancerListener, CreateLoadBalancerPolicy, Delete*, ModifyLoadBalancerAttribute, RemoveTags, SetLoadBalancer*	20	3

Die folgende Tabelle zeigt die Standardkapazität und die Nachfüllraten für die Token-Buckets mit unkategorisierten Anforderungen für ELBv2

ELBv2 Aktionen	Kapazität des Eimers	Nachfüllrate (pro Sekunde)
CreateTrustStore	10	0,2 †

ELBv2 Aktionen	Kapazität des Eimers	Nachfüllrate (pro Sekunde)
AddTrustStoreRevocations , DeleteSharedTrustStoreAssociation , DeleteTrustStore , ModifyTrustStore , RemoveTrustStoreRevocations	10	0,2 †
GetResourcePolicy , GetTrustStoreCaCertificatesBundle , GetTrustStoreRevocationContent	20	4
DescribeTrustStoreAssociations , DescribeTrustStoreRevocations , DescribeTrustStores	40	10

† Bei teilweisen Nachfüllraten dauert es mehrere Sekunden, bis ein vollständiger Token generiert ist.

API-Anfragen überwachen

Sie können AWS CloudTrail damit Ihre ELB-API-Anfragen überwachen. Weitere Informationen finden Sie unter [API-Aufrufe für ELB protokollieren mit AWS CloudTrail](#).

Verstehen Sie die Codes für ELB in Abrechnungs- und Nutzungsberichten

Wenn Sie ELB verwenden, nehmen wir die entsprechenden Codes in Ihre AWS Abrechnungs- und Nutzungsberichte auf. Die Überprüfung dieser Codes hilft Ihnen dabei, Ihre Load Balancer-Kosten und Nutzungsmuster zu verstehen. Die Nachverfolgung und Verwaltung Ihrer Ausgaben ist für die Optimierung Ihrer Kosten unerlässlich.

Weitere Informationen finden Sie unter [ELB-Preise](#).

In den folgenden Tabellen werden die Codes für ELB beschrieben, die in Ihren Abrechnungs- und Nutzungsberichten erscheinen. Bei den Einheiten handelt es sich um Stunden oder Load Balancer-Kapazitätseinheiten (LCU). Jeder Load Balancer-Typ hat eine spezifische Definition von LCU.

[Informationen zu den LCUs einzelnen Load Balancer-Typen finden Sie unter ELB-Preise](#). Eine Liste der Regionscodes in den Fakturierungs- und Nutzungsberichten finden Sie unter [AWS -Regions-Fakturierungscodes](#).

Application Load Balancer

Code	Description	Einheiten
<code>region</code> -LoadBalancerUsage	Die Laufzeit.	Stunden
<code>region</code> -LCUUsage	Die LCUs gebrauchten.	LCU
<code>region</code> -IdleProvisionedLBCapacity	Das LCUs reservierte, aber nicht genutzte.	LCU
<code>region</code> -TS-LoadBalancerUsage	Der Zeitpunkt, zu dem ein Trust Store von Mutual TLS verwendet wird.	Stunden
<code>region</code> -Outposts-LoadBalancerUsage	Die Laufzeit auf Outposts.	Stunden

Code	Description	Einheiten
<i>region</i> -Outposts-LCUUsage	Die auf Outposts LCUs verwendeten.	LCU
<i>region</i> -ReservedLCUUsage	Das LCUs Reservierte.	LCU

Network Load Balancers

Code	Description	Einheiten
<i>region</i> -LoadBalancerUsage	Die Laufzeit.	Stunden
<i>region</i> -LCUUsage	Die LCUs gebrauchten.	LCU

Gateway Load Balancer

Code	Description	Einheiten
<i>region</i> -LoadBalancerUsage	Die Laufzeit.	Stunden
<i>region</i> -LCUUsage	Die LCUs gebrauchten.	LCU

Classic Load Balancer

Code	Description	Einheiten
<i>region</i> -LoadBalancerUsage	Die Laufzeit.	Stunden

Code	Description	Einheiten
<i>region</i> -DataProcessing-Bytes	Die verarbeiteten Daten.	GB
<i>region</i> -IdleProvisionedLB Capacity	Die LCUs reservierten, aber nicht verwendeten.	LCU

API-Aufrufe für ELB protokollieren mit AWS CloudTrail

ELB ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der von einem Benutzer, einer Rolle oder einem AWS Dienst ausgeführten Aktionen bereitstellt. CloudTrail erfasst API-Aufrufe für ELB als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von AWS-Managementkonsole und Codeaufrufen der ELB-API-Operationen. Anhand der von CloudTrail gesammelten Informationen können Sie die Anfrage an ELB, die IP-Adresse, von der aus die Anfrage gestellt wurde, den Zeitpunkt der Anfrage und weitere Details ermitteln.

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anforderung mit Anmeldeinformationen des Root-Benutzers oder des Benutzers gestellt wurde.
- Die Anforderung wurde im Namen eines IAM-Identity-Center-Benutzers erstellt.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anforderung aus einem anderen AWS-Service gesendet wurde.

CloudTrail ist in Ihrem aktiv AWS-Konto, wenn Sie das Konto erstellen, und Sie haben automatisch Zugriff auf den CloudTrail Eventverlauf. Der CloudTrail Ereignisverlauf bietet eine einsehbare, durchsuchbare, herunterladbare und unveränderliche Aufzeichnung der aufgezeichneten Verwaltungsereignisse der letzten 90 Tage in einem AWS-Region. Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter [Arbeiten mit dem CloudTrail Ereignisverlauf](#). Für die Anzeige des Eventverlaufs CloudTrail fallen keine Gebühren an.

Für eine fortlaufende Aufzeichnung der Ereignisse in AWS-Konto den letzten 90 Tagen erstellen Sie einen Trail- oder [CloudTrailLake-Event-Datenspeicher](#).

CloudTrail Pfade

Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Alle mit dem erstellten Pfade AWS-Managementkonsole sind regionsübergreifend. Sie können mithilfe von AWS CLI einen Einzel-Region- oder einen Multi-Region-Trail erstellen. Es wird empfohlen, einen Trail mit mehreren Regionen zu erstellen, da Sie alle Aktivitäten AWS-Regionen in Ihrem Konto erfassen. Wenn Sie einen Einzel-Region-Trail erstellen, können Sie nur die Ereignisse anzeigen, die im AWS-Region des Trails protokolliert wurden. Weitere

Informationen zu Trails finden Sie unter [Erstellen eines Trails für Ihr AWS-Konto](#) und [Erstellen eines Trails für eine Organisation](#) im AWS CloudTrail -Benutzerhandbuch.

Sie können eine Kopie Ihrer laufenden Verwaltungsereignisse kostenlos an Ihren Amazon S3 S3-Bucket senden, CloudTrail indem Sie einen Trail erstellen. Es fallen jedoch Amazon S3 S3-Speichergebühren an. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter [AWS CloudTrail Preise](#). Informationen zu Amazon-S3-Preisen finden Sie unter [Amazon S3 – Preise](#).

CloudTrail Datenspeicher für Ereignisse in Lake

CloudTrail Mit Lake können Sie SQL-basierte Abfragen für Ihre Ereignisse ausführen. CloudTrail [Lake konvertiert bestehende Ereignisse im zeilenbasierten JSON-Format in das Apache ORC-Format](#). ORC ist ein spaltenförmiges Speicherformat, das für den schnellen Abruf von Daten optimiert ist. Die Ereignisse werden in Ereignisdatenspeichern zusammengefasst, bei denen es sich um unveränderliche Sammlungen von Ereignissen handelt, die auf Kriterien basieren, die Sie mit Hilfe von [erweiterten Ereignisselektoren](#) auswählen. Die Selektoren, die Sie auf einen Ereignisdatenspeicher anwenden, steuern, welche Ereignisse bestehen bleiben und für Sie zur Abfrage verfügbar sind. Weitere Informationen zu CloudTrail Lake finden Sie unter [Arbeiten mit AWS CloudTrail Lake](#) im AWS CloudTrail Benutzerhandbuch.

CloudTrail Für das Speichern und Abfragen von Ereignisdaten in Lake fallen Kosten an. Beim Erstellen eines Ereignisdatenspeichers wählen Sie die [Preisoption](#) aus, die für den Ereignisdatenspeicher genutzt werden soll. Die Preisoption bestimmt die Kosten für die Erfassung und Speicherung von Ereignissen sowie die standardmäßige und maximale Aufbewahrungszeit für den Ereignisdatenspeicher. Weitere Informationen zur CloudTrail Preisgestaltung finden Sie unter [AWS CloudTrail Preise](#).

Veranstaltungen des ELB-Managements in CloudTrail

[Managementereignisse](#) bieten Informationen zu Verwaltungsvorgängen, die an Ressourcen in Ihrem ausgeführt werden AWS-Konto. Sie werden auch als Vorgänge auf Steuerebene bezeichnet. CloudTrail Protokolliert standardmäßig Verwaltungsergebnisse.

ELB protokolliert Operationen auf der Kontrollebene als Verwaltungsergebnisse. Eine Liste der Operationen auf der Kontrollebene finden Sie im Folgenden:

- Application Load Balancers — [Elastic Load Balancing API-Referenzversion 2015-12-01](#)
- Network Load Balancers — [Elastic Load Balancing API-Referenzversion 2015-12-01](#)
- Gateway Load Balancers — [Elastic Load Balancing API-Referenzversion 2015-12-01](#)

- Classic Load Balancers — [Elastic Load Balancing API-Referenzversion 2012-06-01](#)

Beispiele für ELB-Ereignisse

Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und enthält Informationen über den angeforderten API-Vorgang, Datum und Uhrzeit des Vorgangs, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass Ereignisse nicht in einer bestimmten Reihenfolge angezeigt werden.

Die folgenden Beispiele zeigen CloudTrail Ereignisse für einen Benutzer, der einen Load Balancer erstellt und ihn dann mit dem AWS CLI gelöscht hat. Sie können die CLI mithilfe der `userAgent`-Elemente identifizieren. Sie können die angeforderten API-Aufrufe mithilfe der `eventName`-Elemente identifizieren. Informationen zum Benutzer (Alice) finden Sie im `userIdentity`-Element.

Example Beispiel 1: CreateLoadBalancer aus der API ELBv2

```

        "loadBalancerName": "my-load-balancer",
        "vpcId": "vpc-3ac0fb5f",
        "securityGroups": ["sg-5943793c"],
        "state": {"code": "provisioning"},
        "availabilityZones": [
            {"subnetId": "subnet-8360a9e7", "zoneName": "us-west-2a"},
            {"subnetId": "subnet-b7d581c0", "zoneName": "us-west-2b"}
        ],
        "dNSName": "my-load-balancer-1836718677.us-west-2.elb.amazonaws.com",
        "canonicalHostedZoneId": "Z2P70J7HTTPPLU",
        "createdTime": "Apr 11, 2016 5:23:50 PM",
        "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/ffcddace1759e1d0",
        "scheme": "internet-facing"
    ],
},
"requestID": "b9960276-b9b2-11e3-8a13-f1ef1EXAMPLE",
"eventID": "6f4ab5bd-2daa-4d00-be14-d92efEXAMPLE",
"eventType": "AwsApiCall",
"apiVersion": "2015-12-01",
"recipientAccountId": "123456789012"
}

```

Example Beispiel 2: DeleteLoadBalancer aus der ELBv2 API

```

{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "DeleteLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {

```

```
        "loadBalancerArn": "arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/ffcddace1759e1d0"
    },
    "responseElements": null,
    "requestID": "349598b3-000e-11e6-a82b-298133eEXAMPLE",
    "eventID": "75e81c95-4012-421f-a0cf-babdaEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-12-01",
    "recipientAccountId": "123456789012"
}
```

Example Beispiel 3: CreateLoadBalancer aus der ELB-API

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAJDPLRKLG7UEXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "CreateLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "subnets": ["subnet-12345678", "subnet-76543210"],
        "loadBalancerName": "my-load-balancer",
        "listeners": [
            {
                "protocol": "HTTP",
                "loadBalancerPort": 80,
                "instanceProtocol": "HTTP",
                "instancePort": 80
            }
        ],
        "responseElements": {
            "dNSName": "my-loadbalancer-1234567890.elb.amazonaws.com"
        },
        "requestID": "b9960276-b9b2-11e3-8a13-f1ef1EXAMPLE"
    }
}
```

```
"eventID": "6f4ab5bd-2daa-4d00-be14-d92efEXAMPLE",
"eventType": "AwsApiCall",
"apiVersion": "2012-06-01",
"recipientAccountId": "123456789012"
}
```

Example Beispiel 4: DeleteLoadBalancer aus der ELB-API

```
{
  "eventVersion": "1.03",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAJDPLRKLG7UEXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/Alice",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2016-04-08T12:39:25Z",
  "eventSource": "elasticloadbalancing.amazonaws.com",
  "eventName": "DeleteLoadBalancer",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "198.51.100.1",
  "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
  "requestParameters": {
    "loadBalancerName": "my-load-balancer"
  },
  "responseElements": null,
  "requestID": "f0f17bb6-b9ba-11e3-9b20-999fdEXAMPLE",
  "eventID": "4f99f0e8-5cf8-4c30-b6da-3b69fEXAMPLE"
  "eventType": "AwsApiCall",
  "apiVersion": "2012-06-01",
  "recipientAccountId": "123456789012"
}
```

Informationen zu CloudTrail Datensatzinhalten finden Sie im AWS CloudTrail Benutzerhandbuch unter [CloudTrail-Datensatzinhalte](#).

Migrieren Ihres Classic Load Balancers

Elastic Load Balancing unterstützt die folgenden Arten von Load Balancern: Application Load Balancer, Network Load Balancer, Gateway Load Balancer und Classic Load Balancer. Informationen zu den verschiedenen Funktionen der einzelnen Load Balancer-Typen finden Sie unter [ELB-Funktionen](#).

Sie können auch wählen, ob Sie einen vorhandenen Classic Load Balancer in einer VPC, auf einen Application Load Balancer oder einen Network Load Balancer migrieren möchten.

Vorteile der Migration von einem Classic Load Balancer

Jeder Load Balancer-Typ hat seine eigenen einzigartigen Merkmale, Funktionen und Konfigurationen. Informieren Sie sich über die Vorteile der einzelnen Load Balancer, um zu entscheiden, welcher für Sie am besten geeignet ist.

Application Load Balancer

Die Verwendung eines Application Load Balancer anstelle eines Classic Load Balancer hat die folgenden Vorteile:

Support für:

- [Pfadbedingungen](#), [Hostbedingungen](#) und [HTTP-Header-Bedingungen](#).
- Umleiten von Anfragen von einer URL zu einer anderen und Weiterleiten von Anfragen an mehrere Anwendungen auf einer einzigen EC2 Instanz.
- Rückgabe benutzerdefinierter HTTP-Antworten.
- Registrierung von Zielen nach IP-Adresse und Registrierung von Lambda-Funktionen als Ziele. Einschließlich Ziele außerhalb der VPC für den Load Balancer.
- Authentifizierung von Benutzern über Unternehmens- oder soziale Identitäten.
- Container-Anwendungen von Amazon Elastic Container Service (Amazon ECS).
- Unabhängige Überwachung des Zustands der einzelnen Dienste.

Zugriffsprotokolle enthalten zusätzliche Informationen und werden in einem komprimierten Format gespeichert.

Die Leistung des Load Balancers wurde insgesamt verbessert.

Network Load Balancer

Die Verwendung eines Network Load Balancer anstelle eines Classic Load Balancer hat die folgenden Vorteile:

Support für:

- Statische IP-Adressen, die die Zuweisung einer Elastic IP-Adresse pro für den Load Balancer aktiviertem Subnetz ermöglichen.
- Registrierung von Zielen nach IP-Adresse, einschließlich Zielen außerhalb der VPC für den Load Balancer.
- Weiterleiten von Anfragen an mehrere Anwendungen auf einer einzigen EC2 Instanz.
- Container-Anwendungen von Amazon Elastic Container Service (Amazon ECS).
- Unabhängige Überwachung des Zustands der einzelnen Dienste.

Möglichkeit, temporäre Verarbeitungslasten zu verarbeiten und eine Skalierung auf Millionen Anfragen pro Sekunde durchzuführen.

Migrieren Sie mit dem Migrationsassistenten

Der Migrationsassistent verwendet die Konfiguration Ihres Classic Load Balancer, um einen gleichwertigen Application Load Balancer oder Network Load Balancer zu erstellen. Es reduziert den Zeit- und Arbeitsaufwand für die Migration eines Classic Load Balancer im Vergleich zu anderen Methoden.

Note

Der Assistent erstellt einen neuen Load Balancer. Der Assistent konvertiert den vorhandenen Classic Load Balancer nicht in einen Application Load Balancer oder Network Load Balancer. Sie müssen den Datenverkehr manuell an den neu erstellten Load Balancer umleiten.

Einschränkungen

- Der Name des neuen Load Balancers darf nicht mit einem vorhandenen Load Balancer desselben Typs in derselben Region identisch sein.

- Wenn der Classic Load Balancer Tags hat, die das aws : Präfix in ihrem Schlüssel enthalten, werden diese Tags nicht migriert.

Bei der Migration zu einem Application Load Balancer

- Wenn der Classic Load Balancer nur ein Subnetz hat, müssen Sie ein zweites Subnetz angeben.
- Wenn der Classic Load Balancer über HTTP/HTTPS Listener verfügt, die TCP-Zustandsprüfungen verwenden, wird das Integritätsprüfungsprotokoll auf HTTP aktualisiert und der Pfad auf „/“ gesetzt.
- Wenn der Classic Load Balancer über HTTPS-Listener verfügt, die eine benutzerdefinierte oder nicht unterstützte Sicherheitsrichtlinie verwenden, verwendet der Migrationsassistent die Standardsicherheitsrichtlinie für den neuen Load Balancer-Typ.

Bei der Migration zu einem Network Load Balancer

- Die folgenden Instance-Typen werden nicht bei der neuen Zielgruppe registriert: C1,,, CC1, CC2, CG1 CG2, G1 CR1 CS1, G2,,, M1 HI1 HS1, M2, M3, T1
- Bestimmte Einstellungen für den Gesundheitscheck Ihres Classic Load Balancer sind möglicherweise nicht auf die neue Zielgruppe übertragbar. Diese Fälle werden im Übersichtsbereich des Migrationsassistenten als Änderung angezeigt.
- Wenn der Classic Load Balancer über SSL-Listener verfügt, erstellt der Migrationsassistent einen TLS-Listener unter Verwendung des Zertifikats und der Sicherheitsrichtlinie des SSL-Listeners.

Prozess des Migrationsassistenten

So migrieren Sie einen Classic Load Balancer mithilfe des Migrationsassistenten

1. Öffnen Sie die EC2 Amazon-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter LOAD BALANCING die Option Load Balancers aus.
3. Wählen Sie den Classic Load Balancer aus, den Sie migrieren möchten.
4. Wählen Sie im Abschnitt Details zum Load Balancer die Option Migrationsassistent starten aus.
5. Wählen Sie Zu Application Load Balancer migrieren oder Zu Network Load Balancer migrieren, um den Migrationsassistenten zu öffnen.
6. Geben Sie unter Name new load balancer für Load Balancer name einen Namen für Ihren neuen Load Balancer ein.

7. Geben Sie unter Neue Zielgruppe benennen und Ziele überprüfen für Zielgruppenname einen Namen für Ihre neue Zielgruppe ein.
8. (Optional) Unter Ziele können Sie die Ziel-Instances überprüfen, die für die neue Zielgruppe registriert werden.
9. (Optional) Unter Tags überprüfen können Sie die Tags überprüfen, die auf Ihren neuen Load Balancer angewendet werden
10. Überprüfen und verifizieren Sie unter Zusammenfassung für Application Load Balancer oder Zusammenfassung für Network Load Balancer die vom Migrationsassistenten zugewiesenen Konfigurationsoptionen.
11. Wenn Sie mit der Zusammenfassung der Konfiguration zufrieden sind, wählen Sie Create Application Load Balancer oder Create Network Load Balancer, um die Migration zu starten.

Migrieren Sie mit dem Load Balancer-Kopierprogramm

Die Load Balancer-Kopierdienstprogramme sind im ELB Tools-Repository auf der AWS GitHub Seite verfügbar.

Ressourcen

- [ELB-Tools](#)
- [Programm zum Kopieren von Classic Load Balancer zu Application Load Balancer](#)
- [Programm zum Kopieren von Classic Load Balancer zu Network Load Balancer](#)

Migrieren Sie Ihren Load Balancer manuell

Die folgenden Informationen enthalten allgemeine Anweisungen zur manuellen Erstellung eines neuen Application Load Balancers oder Network Load Balancers auf der Grundlage eines vorhandenen Classic Load Balancers in einer VPC. Sie können mit dem AWS-Managementkonsole AWS CLI, dem oder einem AWS SDK migrieren. Weitere Informationen finden Sie unter [Erste Schritte mit ELB](#).

Nach Abschluss der Migration können Sie die Vorteile der Features Ihres neuen Load Balancers nutzen.

Manueller Migrationsprozess

Schritt 1: Erstellen eines neuen Load Balancers

Erstellen Sie einen Load Balancer mit einer Konfiguration, die dem zu migrierenden Classic Load Balancer entspricht.

1. Erstellen Sie einen neuen Load Balancer mit dem gleichen Schema (mit dem Internet verbunden oder intern), den gleichen Subnetzen und Sicherheitsgruppen wie der Classic Load Balancer.
2. Erstellen Sie eine Zielgruppe für Ihren Load Balancer mit den gleichen Einstellungen für die Zustandsprüfung, die Sie für Ihren Classic Load Balancer haben.
3. Führen Sie eine der folgenden Aktionen aus:
 - Wenn Ihr Classic Load Balancer mit einer Auto-Scaling-Gruppe verbunden ist, fügen Sie Ihre Zielgruppe der Auto-Scaling-Gruppe hinzu. Dadurch wird die Auto-Scaling-Instance außerdem bei der Zielgruppe registriert.
 - Registrieren Sie Ihre EC2 Instanzen bei Ihrer Zielgruppe.
4. Erstellen Sie eine oder mehrere Listener, jeweils mit einer Standardregel, die Anfragen an die Zielgruppe weiterleitet. Wenn Sie einen HTTPS-Listener erstellen, können Sie dasselbe Zertifikat angeben, das Sie für Ihren Classic Load Balancer angegeben haben. Wir empfehlen, dass Sie die Standardsicherheitsrichtlinie verwenden.
5. Wenn Ihr Classic Load Balancer Tags hat, überprüfen Sie diese und fügen Sie die relevanten Tags Ihrem neuen Load Balancer hinzu.

Schritt 2: Allmähliche Umleitung von Datenverkehr auf Ihren neuen Load Balancer

Nachdem Ihre Instances bei Ihrem neuen Load Balancer registriert wurden, können Sie mit der Umleitung des Datenverkehrs vom alten Load Balancer auf den neuen beginnen. So können Sie Ihren neuen Load Balancer testen und gleichzeitig das Risiko für die Verfügbarkeit Ihrer Anwendung minimieren.

Allmähliches Umleiten des Datenverkehrs auf den neuen Load Balancer

1. Fügen Sie den DNS-Namen Ihres neuen Load Balancers in das Adressfeld eines mit dem Internet verbundenen Webbrowsers ein. Wenn alles funktioniert, zeigt der Browser die Standardseite Ihrer Anwendung an.
2. Erstellen Sie einen neuen DNS-Datensatz, der Ihren Domainnamen mit Ihrem neuen Load Balancer verknüpft. Wenn Ihr DNS-Service Gewichtung unterstützt, geben Sie die Gewichtung 1 in den neuen DNS-Datensatz und eine Gewichtung von 9 in den vorhandenen DNS-Datensatz für Ihren alten Load Balancer ein. Dies leitet 10 % des Datenverkehrs an den neuen Load Balancer und 90 % des Datenverkehrs an den alten.

3. Überwachen Sie Ihren neuen Load Balancer, um sicherzustellen, dass er Datenverkehr empfängt und Anforderungen an die Instances weiterleitet.

 **Important**

Die time-to-live (TTL) im DNS-Eintrag beträgt 60 Sekunden. Dies bedeutet, dass jeder DNS-Server, der Ihren Domainnamen auflöst, die Datensatzinformationen 60 Sekunden lang im Cache aufbewahrt, während die Änderungen weitergeleitet werden. Daher können diese DNS-Server noch bis zu 60 Sekunden nach Abschluss des vorherigen Schritts Datenverkehr an Ihren alten Load Balancer weiterleiten. Während dieser Übertragung kann der Datenverkehr an einen der beiden Load Balancer weitergeleitet werden.

4. Aktualisieren Sie weiterhin die Gewichtung Ihrer DNS-Datensätze, bis der gesamte Datenverkehr an Ihren neuen Load Balancer geleitet wird. Wenn dies abgeschlossen ist, können Sie den DNS-Datensatz für Ihren alten Load Balancer löschen.

Schritt 3: Aktualisieren von Richtlinien, Skripts und Code

Wenn Sie Ihren Classic Load Balancer auf einen Application Load Balancer oder Network Load Balancer migriert haben, müssen Sie Folgendes erledigen:

- Aktualisieren Sie IAM-Richtlinien, die die API-Version 2012-06-01 verwenden, auf die Version 2015-12-01.
- Aktualisieren Sie Prozesse, die CloudWatch Metriken im AWS/ELB Namespace verwenden, um Metriken aus dem AWS/ApplicationELB Or-Namespace zu verwenden. AWS/NetworkELB
- Aktualisieren Sie Skripts, die aws elb AWS CLI Befehle verwenden, um Befehle zu verwenden aws elbv2 AWS CLI .
- Aktualisieren Sie CloudFormation Vorlagen, die die AWS::ElasticLoadBalancing::LoadBalancer Ressource verwenden, um die AWS::ElasticLoadBalancingV2 Ressourcen zu verwenden.
- Aktualisieren Sie den Code, der die ELB-API-Version 2012-06-01 verwendet, um Version 2015-12-01 zu verwenden.

Ressourcen

- [elbv2](#) in der AWS CLI -Befehlsreferenz

- [Elastic-Load-Balancing-API-Referenz Version 2015-12-01](#)
- [Identitäts- und Zugriffsmanagement für ELB](#)
- [Application-Load-Balancer-Metriken](#) im Benutzerhandbuch für Application Load Balancer
- [Network-Load-Balancer-Metriken](#) im Benutzerhandbuch für Network Load Balancer
- [AWS::ElasticLoadBalancingV2::LoadBalancer](#) im AWS CloudFormation -Benutzerhandbuch

Schritt 4: Löschen des alten Load Balancers

Sie können den alten Classic Load Balancer löschen, nachdem:

- Sie den gesamten Datenverkehr an den neuen Load Balancer umgeleitet haben und
- alle vorhandenen Anfragen, die an den alten Load Balancer weitergeleitet wurden, abgeschlossen wurden.

Verhindern Sie, dass Benutzer Classic Load Balancer erstellen

Sie können eine IAM-Richtlinie erstellen, die verhindert, dass Benutzer Classic Load Balancer in Ihrem Konto erstellen.

Sowohl [ELB V2](#) als auch [ELB V1](#) APIs bieten eine `CreateLoadBalancer` API-Aktion. Wenn Sie einen Classic Load Balancer erstellen, verwenden Sie die V1-API-Aktion, die sowohl den Load Balancer als auch die Listener erstellt. Wenn Sie einen Application Load Balancer, Network Load Balancer oder Gateway Load Balancer erstellen, verwenden Sie die V2-API-Aktion, die nur den Load Balancer erstellt. Die V2-API bietet eine `CreateListener` Aktion, mit der Sie Listener für einen Load Balancer erstellen, nachdem Sie ihn erstellt haben.

Die folgende Richtlinie verweigert Benutzern die Erlaubnis, einen Load Balancer zu erstellen, wenn das Listener-Protokoll angegeben ist. Da Sie bei der Erstellung eines Classic Load Balancers mindestens einen Listener konfigurieren müssen, verhindert diese Richtlinie, dass Benutzer Classic Load Balancer erstellen. Sie hindert Benutzer nicht daran, andere Arten von Load Balancers zu erstellen, da es separate API-Aktionen für die Erstellung dieser Load Balancer und ihrer Listener gibt.

```
{  
  "Version": "2012-10-17",  
  "Effect": "Deny",  
  "Action": "elasticloadbalancing:CreateLoadBalancer",  
  "Resource": [
```

```
    "arn:aws:elasticloadbalancing:*:*:loadbalancer/*"
],
"Condition": {
    "Null": {
        "elasticloadbalancing:ListenerProtocol": false
    }
}
}
```

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.