



Application Load Balancer

Elastic Load Balancing



Elastic Load Balancing: Application Load Balancer

Copyright © 2024 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 ein Application Load Balancer?	1
Application-Load-Balancer-Komponenten	1
Application Load Balancer – Übersicht	2
Vorteile der Migration von einem Classic Load Balancer	3
Zugehörige Services	4
Preisgestaltung	5
Erste Schritte	6
Bevor Sie beginnen	6
Schritt 1: Konfigurieren Ihrer Zielgruppe	6
Schritt 2: Wählen eines Load Balancer-Typs	7
Schritt 3: Konfigurieren Ihres Load Balancers und Listeners	8
Schritt 4: Testen Ihres Load Balancers	9
Schritt 5: Löschen Ihres Load Balancers (optional)	9
Tutorial: Erstellen eines Application Load Balancers mithilfe der AWS CLI	11
Bevor Sie beginnen	11
Erstellen Ihres Load Balancers	11
Hinzufügen eines HTTPS-Listeners	13
Hinzufügen von pfadbasierter Weiterleitung	14
Löschen des Load Balancers	15
Load Balancers	16
Subnetze für Ihren Load Balancer	17
Availability-Zone-Subnetze	17
Local-Zone-Subnetze	18
Outpost-Subnetze	18
Load-Balancer-Sicherheitsgruppen	20
Load Balancer-Status	20
Load Balancer-Attribute	20
IP-Adresstyp	23
Ressourcenübersicht für den Load Balancer	24
Komponenten der Ressourcenübersicht	25
Load Balancer-Verbindungen	26
Zeitlimit für Verbindungsleerlauf	26
Keepalive-Dauer des HTTP-Clients	27
Zonenübergreifendes Load Balancing	28

Löschschutz	29
Desynchroner Mitigationsmodus	30
Beibehalten der Host-Header	32
AWS WAF	34
Erstellen eines Load Balancers	36
Schritt 1: Konfigurieren einer Zielgruppe	6
Schritt 2: Registrieren von Zielen	38
Schritt 3: Konfigurieren von Load Balancer und Listener	38
Schritt 4: Testen des Load Balancers	9
Aktualisieren von Availability Zones	43
Aktualisieren von Sicherheitsgruppen	44
Empfohlene Regeln	44
Aktualisieren der zugeordneten Sicherheitsgruppen	47
Aktualisieren des Adresstyps	48
Aktualisieren der Tags	49
Löschen eines -Load Balancers	50
Zonenverschiebung	51
Starten einer Zonenverschiebung	52
Aktualisieren einer Zonenverschiebung	53
Abbrechen einer Zonenverschiebung	54
Listener und Regeln	55
Listener-Konfiguration	55
Listener-Regeln	56
Standardregeln	57
Priorität der Regel	57
Regelaktionen	57
Regelbedingungen	57
Regelaktionstypen	57
Aktionen mit feststehender Antwort	58
Weiterleitungsaktionen	59
Weiterleitungsaktionen	62
Regelbedingungstypen	66
HTTP-Header-Bedingungen	67
Bedingungen für HTTP-Anforderungsmethoden	68
Hostbedingungen	69
Pfadbedingungen	70

Abfragezeichenfolgebedingungen	71
Bedingungen für die Quell-IP-Adresse	72
Erstellen eines HTTP-Listeners	73
Voraussetzungen	73
Hinzufügen eines HTTP-Listeners	73
Erstellen eines HTTPS-Listeners	74
SSL-Zertifikate	75
Sicherheitsrichtlinien	78
Hinzufügen eines HTTPS-Listeners	102
Aktualisieren von Listener-Regeln	104
Voraussetzungen	104
Hinzufügen einer Regel	105
Bearbeiten einer Regel	107
Ändern der Reihenfolge von Regeln	108
Löschen einer Regel	109
Aktualisieren eines HTTPS-Listeners	110
Ersetzen des Standardzertifikats	110
Hinzufügen von Zertifikaten zu einer Zertifikatliste	111
Entfernen eines Zertifikats aus der Zertifikatliste	112
Aktualisieren der Sicherheitsrichtlinie	112
Verwenden Sie die gegenseitige TLS-Authentifizierung	113
Bevor Sie beginnen	114
HTTP-Header	117
Konfiguration von gegenseitigem TLS	119
Verbindungsprotokolle	125
Authentifizieren von Benutzern	125
Vorbereiten der Nutzung eines OIDC-konformen Identitätsanbieters	126
Vorbereitung für die Verwendung von Amazon Cognito	126
Bereiten Sie sich auf die Nutzung von Amazon vor CloudFront	128
Konfigurieren der Benutzerauthentifizierung	129
Authentifizierungsfluss	132
Codierung von Benutzeransprüchen und Signaturverifizierung	134
Zeitüberschreitung	138
Authentifizierung und Abmeldung	139
X-Forwarded-Header	140
X-Forwarded-For	140

X-Forwarded-Proto	144
X-Forwarded-Port	144
Aktualisieren der Tags	145
Aktualisieren der Listener-Tags	145
Aktualisieren von Regel-Tags	146
Löschen eines Listeners	147
Zielgruppen	148
Weiterleitungskonfiguration	149
Zieltyp	150
IP-Adresstyp	151
Protokollversion	152
Registrierte Ziele	153
Zielgruppenattribute	154
Routing-Algorithmen	156
Ändern Sie den Routing-Algorithmus einer Zielgruppe	158
Automatische Zielgewichte (ATW)	158
Anomalie-Erkennung	159
Minimierung von Anomalien	160
Verzögerung der Registrierungsaufhebung	162
Modus des langsamen Hochfahrens	163
Erstellen einer Zielgruppe	164
Konfigurieren von Zustandsprüfungen	167
Zustandsprüfungseinstellungen	167
Zustandsstatus des Ziels	170
Ursachencodes für Zustandsprüfungen	172
Zustand der Ziele prüfen	173
Ändern der Zustandsprüfungseinstellungen für eine Zielgruppe	174
Zonenübergreifendes Load Balancing	174
Wenn zonenübergreifendes Load Balancing deaktivieren	176
Zonenübergreifendes Load Balancing aktivieren	177
Zustand der Zielgruppe	178
Maßnahmen bei fehlerhaftem Zustand	178
Anforderungen und Überlegungen	178
Überwachen	179
Beispiel	179
Ändern der Zustandseinstellungen der Zielgruppe	180

Verwenden des Route-53-DNS-Failovers für Ihren Load Balancer	182
Ziele registrieren	183
Zielsicherheitsgruppen	183
Gemeinsam genutzte Subnetze	184
Registrieren oder Aufheben der Registrierung von Zielen	184
Sticky Sessions	187
Sticky Sessions auf Basis der Dauer	189
Anwendungsbasierte Sticky Sessions	191
Lambda-Funktionen als Ziele	194
Vorbereiten der Lambda-Funktion	195
Erstellen Sie einer Zielgruppe für die Lambda-Funktion	187
Empfangen von Ereignissen vom Load Balancer	197
Antwort an den Load Balancer	198
Header mit mehreren Werten	199
Aktivieren von Zustandsprüfungen	202
Aufheben der Registrierung der Lambda-Funktion	203
Aktualisieren der Tags	204
Löschen einer Zielgruppe	205
Überwachen Ihrer Load Balancers	206
CloudWatch Metriken	207
Application-Load-Balancer-Metriken	207
Metrik-Dimensionen für Application Load Balancer	227
Statistiken für Application-Load-Balancer-Metriken	228
CloudWatch Metriken für Ihren Load Balancer anzeigen	229
Zugriffsprotokolle	232
Zugriffsprotokolldateien	232
Zugriffsprotokolleinträge	234
Beispiel-Protokolleinträge	249
Verarbeiten von Zugriffsprotokolldateien	252
Aktivieren der Zugriffsprotokolle	252
Deaktivieren der Zugriffsprotokolle	260
Verbindungsprotokolle	261
Verbindungsprotokolldateien	262
Verbindungsprotokolleinträge	263
Beispiel-Protokolleinträge	267
Verbindungsprotokolldateien werden verarbeitet	268

Aktivieren Sie Verbindungsprotokolle	268
Deaktivieren Sie die Verbindungsprotokolle	275
Anfragenachverfolgung	275
Syntax	276
Einschränkungen	277
CloudTrail Logs	277
Informationen zu Elastic Load Balancing in CloudTrail	278
Verstehen der Elastic-Load-Balancing-Protokolldateieinträge	279
Fehlerbehebung bei Ihren Load Balancern	282
Ein registriertes Ziel ist nicht in Betrieb	282
Clients können keine Verbindung zu einem mit dem Internet verbundenen Load Balancer herstellen	284
Anfragen, die an eine benutzerdefinierte Domain gesendet werden, werden vom Load Balancer nicht empfangen	284
An den Load Balancer gesendete HTTPS-Anfragen geben „NET::ERR_CERT_COMMON_NAME_INVALID“ zurück	285
Der Load Balancer zeigt erhöhte Verarbeitungszeiten an	285
Der Load Balancer sendet als Antwortcode „000“	286
Der Load Balancer generiert einen HTTP-Fehler	286
HTTP 400: Schlechte Anfrage	287
HTTP 401: Unauthorized (Nicht autorisiert)	287
HTTP 403: Forbidden (Verboten)	287
HTTP 405: Methode nicht erlaubt	287
HTTP 408: Anfrage-Timeout	288
HTTP 413: Nutzlast zu hoch	288
HTTP 414: URI zu lang	288
HTTP 460	288
HTTP 463	288
HTTP 464	288
HTTP 500: Interner Serverfehler	289
HTTP 501: Nicht implementiert	289
HTTP 502: Schlechtes Gateway	289
HTTP 503: Dienst nicht verfügbar	290
HTTP 504: Gateway-Timeout	290
HTTP 505: Version wird nicht unterstützt	291
HTTP 507: Nicht genügend Speicherplatz	291

HTTP 501: Unauthorized (Nicht autorisiert)	291
Ein Ziel generiert einen HTTP-Fehler	291
Ein AWS Certificate Manager Zertifikat kann nicht verwendet werden	291
Header mit mehreren Zeilen werden nicht unterstützt	292
Beheben Sie fehlerhafte Ziele mithilfe der Ressourcenübersicht	292
Kontingente	295
Dokumentverlauf	298
.....	ccvii

Was ist ein Application Load Balancer?

Elastic Load Balancing verteilt Ihren eingehenden Datenverkehr automatisch auf mehrere Ziele, z. B. EC2-Instances, Container und IP-Adressen oder eine oder mehrere Availability Zones. Es überwacht den Zustand der registrierten Ziele und leitet den Datenverkehr nur an die fehlerfreien Ziele weiter. Elastic Load Balancing skaliert Ihren Load Balancer, wenn sich der eingehende Datenverkehr im Laufe der Zeit ändert. Es kann automatisch auf die meisten Workloads skaliert werden.

Elastic Load Balancing unterstützt die folgenden Load Balancers: Application Load Balancers, Network Load Balancers, Gateway Load Balancers und Classic Load Balancers. Sie können den Typ des Load Balancers, der Ihren Anforderungen am besten entspricht, auswählen. In diesem Handbuch werden Application Load Balancer beschrieben. Weitere Informationen zu den anderen Load Balancers finden Sie im [Benutzerhandbuch für Network Load Balancer](#), im [Benutzerhandbuch für Gateway Load Balancer](#) und im [Benutzerhandbuch für Classic Load Balancer](#).

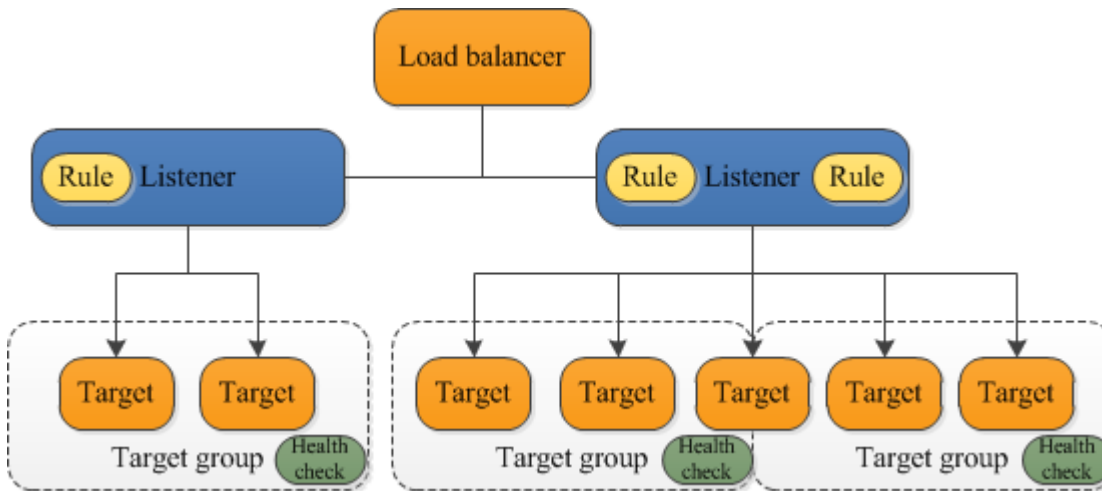
Application-Load-Balancer-Komponenten

Ein Load Balancer dient als zentraler Kontaktpunkt für Clients. Der Load Balancer verteilt eingehenden Anwendungsdatenverkehr auf mehrere Ziele, wie EC2-Instances, in mehreren Availability Zones. Dies erhöht die Verfügbarkeit Ihrer Anwendung. Sie fügen Ihrem Load Balancer einen oder mehrere Listener hinzu.

Ein Listener prüft Verbindungsanforderungen von Clients mit dem Protokoll und dem Port, das bzw. den Sie konfigurieren. Die für einen Listener definierten Regeln bestimmen, wie der Load Balancer Anforderungen an seine registrierten Ziele weiterleitet. Jede Rolle besteht aus einer Priorität, mindestens einer Aktion und mindestens einer Bedingung. Wenn die Bedingungen für eine Regel erfüllt sind, wird die dazugehörige Aktion durchgeführt. Sie müssen für jeden Listener eine Standardregel definieren und können optional zusätzliche Regeln definieren.

Jede Zielgruppe leitet Anforderungen an ein oder mehrere registrierte Ziele, z. B. EC2-Instances, über das Protokoll und die Port-Nummer, die Sie angeben, weiter. Sie können ein Ziel bei mehreren Zielgruppen registrieren. Sie können Zustandsprüfungen pro Zielgruppe konfigurieren. Zustandsprüfungen werden auf allen Zielen ausgeführt, die bei einer Zielgruppe registriert sind, welche in einer Listener-Regel für Ihren Load Balancer abgegeben ist.

Das folgende Diagramm veranschaulicht die grundlegenden Komponenten. Beachten Sie, dass jeder Listener eine Standardregel enthält und ein Listener eine zweite Regel enthält, die Anforderungen an eine andere Zielgruppe weiterleitet. Ein Ziel ist bei zwei Zielgruppen registriert.



Weitere Informationen finden Sie in der folgenden -Dokumentation:

- [Load Balancers](#)
- [Listener](#)
- [Zielgruppen](#)

Application Load Balancer – Übersicht

Ein Application Load Balancer funktioniert auf Anwendungsebene, der siebten Ebene der Open Systems Interconnection (OSI)-Modells. Nachdem der Load Balancer eine Anforderung empfangen hat, bewertet er die Listener-Regeln in der Reihenfolge ihrer Priorität, um zu ermitteln, welche Regel angewendet werden soll. Anschließend wählt er ein Ziel aus der Zielgruppe für die Regelaktion aus. Sie können Listener-Regeln zum Weiterleiten von Anforderungen an verschiedene Zielgruppen basierend auf dem Inhalt des Anwendungsdatenverkehrs konfigurieren. Die Weiterleitung erfolgt unabhängig für jede Zielgruppe, auch wenn ein Ziel bei mehreren Zielgruppen registriert ist. Sie können den Weiterleitungsalgorithmus konfigurieren, der auf der Ebene der Zielgruppe verwendet wird. Als standardmäßiger Routing-Algorithmus wird „Round Robin“ verwendet. Alternativ können Sie auch den Weiterleitungsalgorithmus „Am wenigsten ausstehende Anfragen“ angeben.

Sie können Ziele zu Ihrem Load Balancer hinzufügen und wieder entfernen, wenn sich Ihr Bedarf ändert, ohne den allgemeinen Fluss von Anfragen an Ihre Anwendung zu unterbrechen. Elastic Load Balancing skaliert Ihren Load Balancer, wenn sich der Datenverkehr zu Ihrer Anwendung im Laufe der Zeit ändert. Elastic Load Balancing kann für die meisten Workloads automatisch skaliert werden.

Sie können Zustandsprüfungen konfigurieren, mit denen der Zustand der registrierten Ziele überwacht wird, sodass der Load Balancer nur an die fehlerfreien Ziele Anfragen senden kann.

Weitere Informationen finden Sie unter [Funktionsweise von Elastic Load Balancing](#) im Benutzerhandbuch für Elastic Load Balancing.

Vorteile der Migration von einem Classic Load Balancer

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

- Unterstützung für [Pfadbedingungen](#). Sie können Regeln für Ihre Listener konfigurieren, die Anforderungen basierend auf der URL in der Anforderung weiterleiten. Auf diese Weise können Sie Ihre Anwendung als kleinere Services strukturieren und Anforderungen basierend auf dem Inhalt der URL an den richtigen Service weiterleiten.
- Unterstützung für [Hostbedingungen](#). Sie können Regeln für Ihre Listener konfigurieren, die Anfragen basierend auf dem Hostfeld im HTTP-Header weiterleiten. Auf diese Weise können Sie mit einem einzigen Load Balancer Anfragen zu mehreren Domains weiterleiten.
- Unterstützung für Weiterleitung auf Grundlage von Feldern in der Anforderung, z. B. [HTTP-Header-Bedingungen](#) und Methoden, Abfrageparameter und IP-Adressen.
- Unterstützung von Weiterleitungsanfragen an mehrere Anwendungen auf einer einzelnen EC2-Instance. Sie können jede Instance oder IP-Adresse mit mehreren Zielgruppen registrieren, jede auf einem unterschiedlichen Port.
- Unterstützung für das Weiterleiten von Anforderungen von einer URL an eine andere.
- Unterstützung für das Zurückgeben einer benutzerdefinierten HTTP-Antwort.
- Unterstützung einer Registrierung von Zielen unter Verwendung von IP-Adressen, auch für Ziele, die außerhalb der VPC für den Load Balancer liegen.
- Unterstützung für die Registrierung von Lambda-Funktionen als Ziele.
- Unterstützung für das Authentifizieren von Benutzern Ihrer Anwendungen über deren Unternehmensidentitäten oder Social Identities vor dem Weiterleiten von Anfragen durch den Load Balancer.
- Unterstützung für Anwendungen in Containern. Amazon Elastic Container Service (Amazon ECS) kann beim Planen einer Aufgabe und Registrieren der Aufgabe bei einer Zielgruppe einen unbenutzten Port verwenden. Auf diese Weise können Sie Ihre Cluster effizient einsetzen.
- Unterstützung für die unabhängige Überwachung des Zustands jedes Services, da Zustandsprüfungen auf Zielgruppenebene definiert sind und viele CloudWatch Metriken auf Zielgruppenebene gemeldet werden. Wenn Sie eine Zielgruppe einer Auto-Scaling-Gruppe zuweisen, können Sie jeden Service je nach Bedarf dynamisch skalieren.

- Zugriffsprotokolle enthalten weitere Informationen und werden in komprimiertem Format gespeichert.
- Verbesserte Load Balancer-Performance.

Weitere Informationen zu den von den einzelnen Load-Balancer-Typen unterstützten Features finden Sie unter [Produktvergleich](#) für Elastic Load Balancing.

Zugehörige Services

Elastic Load Balancing arbeitet mit den folgenden Services, um die Verfügbarkeit und Skalierbarkeit Ihrer Anwendungen zu verbessern.

- Amazon EC2 – Virtuelle Server, die Ihre Anwendungen in der Cloud ausführen. Sie können Ihren Load Balancer so konfigurieren, dass der Datenverkehr zu Ihren EC2-Instances geleitet wird.
- Amazon EC2 Auto Scaling – Stellt sicher, dass die gewünschte Anzahl von Instances ausgeführt wird, auch wenn eine Instance ausfällt, und ermöglicht die automatische Erhöhung oder Verringerung der Anzahl der Instances, falls sich die Anforderungen an Ihre Instances ändern. Wenn Sie Auto Scaling mit Elastic Load Balancing aktivieren, werden Instances, die von Auto Scaling gestartet werden, automatisch bei der Zielgruppe registriert, und Instances, die durch Auto Scaling beendet wurden, werden automatisch von der Zielgruppe abgemeldet.
- 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. Weitere Informationen finden Sie unter [SSL-Zertifikate](#).
- Amazon CloudWatch – Ermöglicht es Ihnen, Ihren Load Balancer zu überwachen und bei Bedarf Maßnahmen zu ergreifen. Weitere Informationen finden Sie unter [CloudWatch Metriken für Ihren Application Load Balancer](#).
- Amazon ECS – Sie können Docker-Container in einem Cluster von EC2-Instances ausführen, anhalten und verwalten. Sie können Ihren Load Balancer so konfigurieren, dass der Datenverkehr an Ihre Container geleitet wird. Weitere Informationen finden Sie unter [Service – Load Balancing](#) 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 (wie zum Beispiel `www.example.com`) in numerische IP-Adressen (wie zum Beispiel `192.0.2.1`), die Computer zur gegenseitigen Vernetzung verwenden, übersetzt werden. AWS weist Ressourcen URLs zu, wie etwa Load Balancern. Sie können jedoch auch eine URL verwenden, die aussagekräftig und leicht zu merken ist. So können zum Beispiel Ihren Domainnamen einem Load Balancer zuordnen. Weitere Informationen finden Sie unter [Weiterleiten von Datenverkehr an einen ELB Load Balancer](#) im Entwicklerhandbuch von Amazon Route 53.
- AWS WAF – Sie können AWS WAF mit Ihrem Application Load Balancer verwenden, um Anforderungen basierend auf den Regeln in einer Web-ACL (Web-Zugriffskontrollliste) zu erlauben oder zu blockieren. Weitere Informationen finden Sie unter [Application Load Balancers und AWS WAF](#).

Um Informationen über Services anzuzeigen, die in Ihren Load Balancer integriert sind, wählen Sie Ihren Load Balancer in der AWS Management Console aus und klicken Sie dann auf die Registerkarte Integriertes Services.

Preisgestaltung

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

Erste Schritte mit Application Load Balancern

Dieses Tutorial bietet eine praktische Einführung in Application Load Balancern über eine webbasierte Oberfläche. AWS Management Console Führen Sie die folgenden Schritte aus, um Ihren ersten Application Load Balancer zu erstellen.

Aufgaben

- [Bevor Sie beginnen](#)
- [Schritt 1: Konfigurieren Ihrer Zielgruppe](#)
- [Schritt 2: Wählen eines Load Balancer-Typs](#)
- [Schritt 3: Konfigurieren Ihres Load Balancers und Listeners](#)
- [Schritt 4: Testen Ihres Load Balancers](#)
- [Schritt 5: Löschen Ihres Load Balancers \(optional\)](#)

Demos häufiger Load-Balancer-Konfigurationen finden Sie unter [Elastic-Load-Balancing-Demos](#).

Bevor Sie beginnen

- Entscheiden Sie, welche beiden Availability Zones Sie für Ihre EC2-Instances verwenden. Konfigurieren Sie Ihre VPC (Virtual Private Cloud) in jeder dieser Availability Zones mit mindestens einem öffentlichen Subnetz. Diese öffentlichen Subnetze werden verwendet, um den Load Balancer zu konfigurieren. Sie können Ihre EC2-Instances in anderen Subnetzen dieser Availability Zones starten.
- Starten Sie in jeder Availability Zone mindestens eine EC2-Instance. Achten Sie darauf, auf jeder EC2-Instance einen Webserver wie z. B. Apache oder Internet Information Services (IIS) zu installieren. Stellen Sie sicher, dass die Sicherheitsgruppen für diese Instances den HTTP-Zugriff auf Port 80 erlauben.

Schritt 1: Konfigurieren Ihrer Zielgruppe

Erstellen Sie eine Zielgruppe, die bei der Weiterleitung von Anforderungen verwendet wird. Die Standardregel für Ihren Listener leitet Anforderungen an die registrierten Ziele in dieser Zielgruppe weiter. Der Load Balancer prüft anhand der Zustandsprüfungseinstellungen, die für die Zielgruppe definiert sind, den Zustand der Ziele in dieser Zielgruppe.

So konfigurieren Sie Ihre Zielgruppe mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter LOAD BALANCING die Option Load Balancers aus.
3. Wählen Sie Zielgruppe erstellen aus.
4. Behalten Sie unter Basiskonfiguration den Zieltyp als Instance bei.
5. Geben Sie unter Zielgruppenname einen Namen für die neue Zielgruppe ein.
6. Behalten Sie das Standardprotokoll (HTTP) und den Port (80) bei.
7. Wählen Sie die VPC aus, die Ihre Instances enthält. Behalten Sie die Protokollversion als HTTP1 bei.
8. Behalten Sie für Zustandsprüfungen die Standardeinstellungen bei.
9. Wählen Sie Weiter.
10. Führen Sie auf der Seite Ziele registrieren die folgenden Schritte aus. Dies ist ein optionaler Schritt zum Erstellen des Load Balancers. Sie müssen dieses Ziel jedoch registrieren, wenn Sie Ihren Load Balancer testen und sicherstellen möchten, dass er den Datenverkehr zu diesem Ziel weiterleitet.
 - a. Wählen Sie im Feld Verfügbare Instances eine oder mehrere Instances aus.
 - b. Behalten Sie den Standardport 80 bei und wählen Sie Schließen Sie die unten angeführten als ausstehend ein aus.
11. Wählen Sie Zielgruppe erstellen aus.

Schritt 2: Wählen eines Load Balancer-Typs

Elastic Load Balancing unterstützt verschiedene Load-Balancer-Typen. In diesem Lernprogramm erstellen Sie einen Application Load Balancer.

So erstellen Sie einen Application Load Balancer mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie in der Navigationsleiste eine Region für Ihren Load Balancer aus. Stellen Sie sicher, dass Sie die gleiche Region auswählen, die Sie für Ihre EC2-Instances verwendet haben.
3. Wählen Sie im Navigationsbereich unter Load Balancing die Option Load Balancers aus.
4. Klicken Sie auf Load Balancer erstellen.

5. Wählen Sie für Application Load Balancer Erstellen aus.

Schritt 3: Konfigurieren Ihres Load Balancers und Listeners

Um einen Application Load Balancer zu erstellen, müssen Sie zunächst grundlegende Konfigurationsinformationen für Ihren Load Balancer angeben, z. B. einen Namen, ein Schema und einen IP-Adresstyp. Geben Sie anschließend Informationen über Ihr Netzwerk und einen oder mehrere Listener an. Ein Listener ist ein Prozess, der Verbindungsanfragen überprüft. Er wird mit einem Protokoll und einem Port für Verbindungen von Clients zum Load Balancer konfiguriert. Weitere Informationen zu unterstützten Protokollen und Ports finden Sie unter [Listener-Konfiguration](#).

So konfigurieren Sie Load Balancer und Listener

1. Geben Sie im Feld Name des Load Balancers einen Namen für Ihren Load Balancer ein. z. B. my-a1b.
2. Behalten Sie für Schema und IP-Adresstyp die Standardwerte.
3. Wählen Sie im Feld Netzwerkzuordnung die VPC aus, die Sie für Ihre EC2-Instances verwendet haben. Wählen Sie mindestens zwei Availability Zones und ein Subnetz pro Zone aus. Wählen Sie für jede Availability Zone, die Sie verwendet haben, um Ihre EC2-Instances zu starten, die Availability Zone und dann ein öffentliches Subnetz für diese Availability Zone aus.
4. Wählen Sie für Sicherheitsgruppen die Sicherheitsgruppe für die VPC aus, die Sie im vorherigen Schritt ausgewählt haben. Sie können stattdessen eine andere Sicherheitsgruppe wählen. Die Sicherheitsgruppe muss Regeln enthalten, die dem Load Balancer die Kommunikation mit registrierten Zielen sowohl auf dem Listener-Port als auch auf dem Zustandsprüfungs-Port ermöglichen. Weitere Informationen finden Sie unter [Regeln zu Sicherheitsgruppen](#).
5. Behalten Sie für Listener und Routing das Standardprotokoll und den Standardport bei und wählen Sie die Zielgruppe aus der Liste aus. Dadurch wird ein Listener konfiguriert, der HTTP-Verkehr auf Port 80 akzeptiert und den Datenverkehr standardmäßig an die ausgewählte Zielgruppe weiterleitet. In diesem Tutorial erstellen Sie keinen HTTPS-Listener.
6. Wählen Sie für Standardaktion die Zielgruppe aus, die Sie in „Schritt 1: Konfigurieren Ihrer Zielgruppe“ erstellt und registriert haben.
7. (Optional) Sie können zwecks Kategorisierung Ihrem Load Balancer ein Tag hinzufügen. Tag-Schlüssel müssen für jeden Load Balancer eindeutig sein. Erlaubte Zeichen sind Buchstaben, Leerzeichen und Zahlen (in UTF-8) sowie die folgenden Sonderzeichen: + - = _ : / @. Verwenden Sie keine führenden oder nachgestellten Leerzeichen. Bei Tag-Werten muss die Groß- und Kleinschreibung beachtet werden.

- Überprüfen Sie Ihre Konfiguration und wählen Sie Load Balancer erstellen aus. Bei der Erstellung werden einige Standardattribute auf Ihren Load Balancer angewendet. Sie können sie nach der Erstellung des Load Balancers anzeigen und bearbeiten. Weitere Informationen finden Sie unter [Load Balancer-Attribute](#).

Schritt 4: Testen Ihres Load Balancers

Überprüfen Sie nach dem Erstellen des Load Balancer, ob er Datenverkehr an Ihre EC2-Instances sendet.

Testen des Load Balancers

- Nachdem Sie darüber informiert wurden, dass Ihr Load Balancer erfolgreich erstellt wurde, klicken Sie auf Schließen.
- Wählen Sie im Navigationsbereich unter LOAD BALANCING die Option Load Balancers aus.
- Wählen Sie die neu erstellte Zielgruppe aus.
- Wählen Sie Targets und vergewissern Sie sich, dass die Instances bereit sind. Wenn der Status einer Instance `initial` lautet, liegt das wahrscheinlich daran, dass die Instance gerade registriert wird oder dass sie die Mindestanzahl von Zustandsprüfungen nicht bestanden hat, um als fehlerfrei zu gelten. Wenn der Status von mindestens einer Instance `healthy` lautet, können Sie Ihren Load Balancer testen.
- Wählen Sie im Navigationsbereich unter Load Balancing die Option Load Balancers aus.
- Wählen Sie den neu erstellten Load Balancer aus.
- Wählen Sie Beschreibung und kopieren Sie den DNS-Namen des Load Balancers (z. B. `my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com`). Fügen Sie den DNS-Namen in das Adressfeld eines mit dem Internet verbundenen Webbrowsers ein. Wenn alles funktioniert, zeigt der Browser die Standardseite Ihres Servers an.
- (Optional) Informationen zum Definieren von zusätzlichen Listener-Regeln finden Sie unter [Hinzufügen einer Regel](#).

Schritt 5: Löschen Ihres Load Balancers (optional)

Sobald der Load Balancer verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, in der er ausgeführt wird. Wenn Sie einen Load Balancer nicht mehr benötigen, können Sie ihn löschen. Sobald der Load Balancer gelöscht wurde, fallen keine weiteren Kosten

dafür mehr an. Beachten Sie, dass das Löschen eines Load Balancers keine Auswirkungen auf die beim Load Balancer registrierten Ziele hat. Ihre EC2-Instances laufen beispielsweise weiter, nachdem Sie den in diesem Handbuch erstellten Load Balancer gelöscht haben.

Um Ihren Load Balancer mithilfe der Konsole zu löschen

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Load Balancers aus.
3. Aktivieren Sie das Kontrollkästchen für den Load Balancer und wählen Sie dann Aktionen und Löschen aus.
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Ja, löschen.

Tutorial: Erstellen eines Application Load Balancers mithilfe der AWS CLI

Dieses Tutorial bietet eine praktische Einführung in Application Load Balancers über die AWS CLI

Bevor Sie beginnen

- Verwenden Sie den folgenden Befehl, um zu überprüfen, ob Sie eine Version der AWS CLI verwenden, die Application Load Balancer unterstützt.

```
aws elbv2 help
```

Wenn Sie die Fehlermeldung erhalten, dass elbv2 ungültig ist, aktualisieren Sie die AWS CLI. Weitere Informationen finden Sie unter [Installieren der AWS Command Line Interface](#) im AWS Command Line Interface -Benutzerhandbuch.

- Starten Sie Ihre EC2-Instances in einer Virtual Private Cloud (VPC). Stellen Sie sicher, dass die Sicherheitsgruppen für diese Instances den Zugriff auf den Listener-Port und die Port-Zustandsprüfung erlauben. Weitere Informationen finden Sie unter [Zielsicherheitsgruppen](#).
- Entscheiden Sie, ob Sie einen IPv4- oder Dualstack-Load-Balancer erstellen möchten. Verwenden Sie IPv4, wenn Sie möchten, dass die Clients nur über IPv4-Adressen mit dem Load Balancer kommunizieren. Verwenden Sie Dualstack, wenn Sie möchten, dass die Clients über IPv4- und IPv6-Adressen mit dem Load Balancer kommunizieren. Sie können Dualstack auch verwenden, um mithilfe von IPv6 mit Backend-Zielen wie IPv6-Anwendungen oder Dual-Stack-Subnetzen zu kommunizieren.
- Achten Sie darauf, auf jeder EC2-Instance einen Webserver wie z. B. Apache oder Internet Information Services (IIS) zu installieren. Stellen Sie sicher, dass die Sicherheitsgruppen für diese Instances den HTTP-Zugriff auf Port 80 erlauben.

Erstellen Ihres Load Balancers

Führen Sie die folgenden Schritte aus, um Ihren ersten Load Balancer zu erstellen.

Erstellen Sie einen Load Balancer wie folgt:

1. Verwenden Sie den [create-load-balancer](#) Befehl, um einen Load Balancer zu erstellen. Sie müssen zwei Subnetze angeben, die sich nicht in derselben Availability Zone befinden.

```
aws elbv2 create-load-balancer --name my-load-balancer \  
--subnets subnet-0e3f5cac72EXAMPLE subnet-081ec835f3EXAMPLE --security-groups  
sg-07e8ffd50fEXAMPLE
```

Verwenden Sie den [create-load-balancer](#) Befehl, um einen **dualstack** Load Balancer zu erstellen.

```
aws elbv2 create-load-balancer --name my-load-balancer \  
--subnets subnet-0e3f5cac72EXAMPLE subnet-081ec835f3EXAMPLE --security-groups  
sg-07e8ffd50fEXAMPLE --ip-address-type dualstack
```

Die Ausgabe enthält den Amazon Resource Name (ARN) des Load Balancers im folgenden Format:

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

2. Verwenden Sie den [create-target-group](#) Befehl, um eine Zielgruppe zu erstellen, indem Sie dieselbe VPC angeben, die Sie für Ihre EC2-Instances verwendet haben.

Sie können IPOv4- und IPv6-Zielgruppen erstellen, um sie einem Dual-Stack-Load-Balancer zuzuordnen. Der IP-Adresstyp der Zielgruppe bestimmt die IP-Version, die der Load Balancer verwendet, um sowohl mit Ihren Backend-Zielen zu kommunizieren als auch deren Zustand zu überprüfen.

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

Die Ausgabe umfasst den ARN der Zielgruppe in diesem Format:

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

3. Verwenden Sie den Befehl [register-targets](#), um Ihre Instances bei Ihrer Zielgruppe zu registrieren:

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

4. Verwenden Sie den Befehl [create-listener](#), um einen Listener für Ihren Load Balancer mit einer Standardregel zu erstellen, die Anforderungen an Ihre Zielgruppe weiterleitet:

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

Die Ausgabe enthält den ARN des Listeners im folgenden Format:

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

5. (Optional) Mit diesem [describe-target-health](#) Befehl können Sie den Zustand der registrierten Ziele für Ihre Zielgruppe überprüfen:

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

Hinzufügen eines HTTPS-Listeners

Wenn Sie einen Load Balancer mit einem HTTP-Listener haben, können Sie wie folgt einen HTTP-Listener hinzufügen.

So fügen Sie einen HTTPS-Listener zu Ihrem Load Balancer hinzu

1. Erstellen Sie anhand einer der folgenden Methoden ein SSL-Zertifikat für die Verwendung mit Ihrem Load Balancer:
 - Erstellen oder importieren Sie das Zertifikat mit AWS Certificate Manager (ACM). Weitere Informationen finden Sie unter [Anfordern eines Zertifikats](#) oder [Importieren von Zertifikaten](#) im AWS Certificate Manager -Benutzerhandbuch.
 - Laden Sie das Zertifikat mit AWS Identity and Access Management (IAM) hoch. Weitere Informationen finden Sie unter [Arbeiten mit Serverzertifikaten](#) im IAM-Benutzerhandbuch.

2. Verwenden Sie den Befehl [create-listener](#), um den Listener mit einer Standardregel zu erstellen, die Anforderungen an Ihre Zielgruppe weiterleitet. Sie müssen ein SSL-Zertifikat angeben, wenn Sie einen HTTPS-Listener erstellen. Beachten Sie, dass Sie mithilfe der Option `--ssl-policy` eine andere SSL-Richtlinie als die Standardrichtlinie angeben können.

```
aws elbv2 create-listener --load-balancer-arn loadbalancer-arn \  
--protocol HTTPS --port 443 \  
--certificates CertificateArn=certificate-arn \  
--default-actions Type=forward,TargetGroupArn=targetgroup-arn
```

Hinzufügen von pfadbasierter Weiterleitung

Wenn Sie einen Listener mit einer Standardregel haben, der Anforderungen an eine Zielgruppe weiterleitet, können Sie eine Regel hinzufügen, die Anforderungen an eine andere Zielgruppe auf der Grundlage von URL weiterleitet. Sie können beispielsweise allgemeine Anforderungen an die eine Zielgruppe und Anforderungen zur Anzeige von Bildern an eine andere Zielgruppe weiterleiten.

Hinzufügen einer Regel zu einem Listener mit einem Pfadmuster

1. Verwenden Sie den [create-target-group](#) Befehl, um eine Zielgruppe zu erstellen:

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

2. Verwenden Sie den Befehl [register-targets](#), um Ihre Instances bei Ihrer Zielgruppe zu registrieren:

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

3. Verwenden Sie den Befehl [create-rule](#), um zu Ihrem Listener eine Regel hinzuzufügen, die Anforderungen an die Zielgruppe weiterleitet, wenn die URL das angegebene Muster enthält:

```
aws elbv2 create-rule --listener-arn listener-arn --priority 10 \  
--conditions Field=path-pattern,Values='/img/*' \  
--actions Type=forward,TargetGroupArn=targetgroup-arn
```

Löschen des Load Balancers

Wenn Sie Ihren Load Balancer und Ihre Zielgruppe nicht mehr benötigen, können Sie sie wie folgt löschen:

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


Application Load Balancer

Ein Load Balancer dient als zentraler Kontaktpunkt für Clients. Clients senden Anforderungen an den Load Balancer, die dieser wiederum an Ziele wie z. B. EC2-Instances sendet. Zur Konfiguration Ihres Load Balancers erstellen Sie [Zielgruppen](#) und registrieren anschließend Ziele bei den Zielgruppen. Außerdem erzeugen Sie [Listener](#) für Verbindungsanforderungen von Clients und Listener-Regeln zum Weiterleiten von Anforderungen von Clients an Ziele in einer oder mehreren Zielgruppen.

Weitere Informationen finden Sie unter [Funktionsweise von Elastic Load Balancing](#) im Benutzerhandbuch für Elastic Load Balancing.

Inhalt

- [Subnetze für Ihren Load Balancer](#)
- [Load-Balancer-Sicherheitsgruppen](#)
- [Load Balancer-Status](#)
- [Load Balancer-Attribute](#)
- [IP-Adresstyp](#)
- [Ressourcenübersicht für Application Load Balancer](#)
- [Load Balancer-Verbindungen](#)
- [Zonenübergreifendes Load Balancing](#)
- [Löschschutz](#)
- [Desynchroner Mitigationsmodus](#)
- [Beibehalten der Host-Header](#)
- [Application Load Balancers und AWS WAF](#)
- [Erstellen eines Application Load Balancers](#)
- [Availability Zones für Ihren Application Load Balancer](#)
- [Sicherheitsgruppen für Ihren Application Load Balancer](#)
- [IP-Adresstypen für Ihren Application Load Balancer](#)
- [Tags für Ihren Application Load Balancer](#)
- [Löschen eines Application Load Balancers](#)
- [Zonenverschiebung](#)

Subnetze für Ihren Load Balancer

Wenn Sie einen Application Load Balancer erstellen, müssen Sie die Zonen, die Ihre Ziele enthalten, aktivieren. Um eine Zone zu aktivieren, geben Sie ein Subnetz in der Zone an. Elastic Load Balancing erstellt in jeder Zone, die Sie angeben, einen Load-Balancer-Knoten.

Überlegungen

- Ihr Load Balancer ist am effektivsten, wenn Sie dafür sorgen, dass jede aktivierte Zone mindestens ein registriertes Ziel hat.
- Wenn Sie Ziele in einer Zone registrieren, aber die Zone nicht aktivieren, erhalten diese registrierten Ziele keinen Datenverkehr vom Load Balancer.
- Wenn Sie mehrere Zonen für Ihren Load Balancer aktivieren, müssen die Zonen vom gleichen Typ sein. Sie können beispielsweise nicht sowohl eine Availability Zone als auch eine Local Zone aktivieren.
- Sie können ein Subnetz angeben, das für Sie freigegeben wurde.

Application Load Balancer unterstützen die folgenden Arten von Subnetzen.

Subnetz-Typen

- [Availability-Zone-Subnetze](#)
- [Local-Zone-Subnetze](#)
- [Outpost-Subnetze](#)

Availability-Zone-Subnetze

Sie müssen mindestens zwei Availability-Zone-Subnetze auswählen. Beachten Sie die folgenden Einschränkungen:

- Jedes Subnetz muss einer anderen Availability Zone angehören.
- Damit Ihr Load Balancer ordnungsgemäß skaliert werden kann, stellen Sie sicher, dass jedes Availability-Zone-Subnetz für Ihren Load Balancer über einen CIDR-Block mit mindestens einer /27-Bitmaske (z. B. 10.0.0.0/27) und über mindestens acht freie IP-Adressen pro Subnetz verfügt. Diese acht IP-Adressen sind erforderlich, damit der Load Balancer bei Bedarf skalieren kann. Ihr Load Balancer verwendet diese IP-Adressen zum Einrichten von Verbindungen mit den

Zielen. Ohne sie könnte Ihr Application Load Balancer Schwierigkeiten beim Versuch haben, Knoten zu ersetzen. Das könnte dazu führen, dass er in den Fehlerstatus übergeht.

Hinweis: Wenn einem Application-Load-Balancer-Subnetz beim Versuch der Skalierung die verwendbaren IP-Adressen ausgehen, wird der Application Load Balancer mit unzureichender Kapazität ausgeführt. Während dieser Zeit werden alte Knoten weiterhin Datenverkehr bereitstellen, aber der gescheiterte Skalierungsversuch kann zu 5xx-Fehlern oder Timeouts führen, wenn versucht wird, eine Verbindung herzustellen.

Local-Zone-Subnetze

Sie können ein oder mehrere Local-Zone-Subnetze angeben. Beachten Sie die folgenden Einschränkungen:

- Sie können es nicht AWS WAF mit dem Load Balancer verwenden.
- Sie können keine Lambda-Funktion als Ziel verwenden.
- Sticky Sessions oder Application-Stickiness können nicht verwendet werden.

Outpost-Subnetze

Sie können ein einzelnes Outpost-Subnetz angeben. Beachten Sie die folgenden Einschränkungen:

- Sie müssen ein Outpost in Ihrem On-Premises-Rechenzentrum installiert und konfiguriert haben. Sie müssen über eine zuverlässige Netzwerkverbindung zwischen Ihrem Outpost und der entsprechenden AWS -Region verfügen. Weitere Informationen finden Sie im [AWS Outposts - Benutzerhandbuch](#).
- Der Load Balancer benötigt zwei `large`-Instances auf dem Outpost für die Load-Balancer-Knoten. In der folgenden Tabelle sind die unterstützten Instance-Typen aufgeführt. Der Load Balancer skaliert nach Bedarf und ändert die Größe der Knoten jeweils um eine Größe (von `large` in `xlarge`, dann `xlarge` in `2xlarge` und dann `2xlarge` in `4xlarge`). Wenn Sie nach der Skalierung der Knoten in die größte Instance-Größe zusätzliche Kapazität benötigen, fügt der Load Balancer `4xlarge`-Instances als Load-Balancer-Knoten hinzu. Wenn Sie nicht genügend Instance-Kapazität oder verfügbare IP-Adressen haben, um den Load Balancer zu skalieren, meldet der Load Balancer ein Ereignis an [AWS Health Dashboard](#) und der Load-Balancer-Status ist `active_impaired`.

- Sie können Ziele nach Instance-ID oder IP-Adresse registrieren. Wenn Sie Ziele in der AWS Region für den Außenposten registrieren, werden sie nicht verwendet.
- Die folgenden Funktionen sind nicht verfügbar: Lambda-Funktionen als Ziele, AWS WAF - Integration, Sticky Sessions, Authentifizierungsunterstützung und Integration in AWS Global Accelerator.

Ein Application Load Balancer kann in c5/c5d-, m5/m5d- oder r5/r5d-Instances auf einem Outpost bereitgestellt werden. Die folgende Tabelle enthält die Größe und das EBS-Volumen pro Instance-Typ, die der Load Balancer auf einem Outpost verwenden kann:

Instance-Typ und Größe	EBS-Volumen (GB)
c5/c5d	
large	50
xlarge	50
2xlarge	50
4xlarge	100
m5/m5d	
large	50
xlarge	50
2xlarge	100
4xlarge	100
r5/r5d	
large	50
xlarge	100
2xlarge	100

Instance-Typ und Größe	EBS-Volumen (GB)	
4xlarge	100	

Load-Balancer-Sicherheitsgruppen

Eine Sicherheitsgruppe agiert als Firewall, die den Datenverkehr steuert, der in und aus Ihrem Load Balancer zulässig ist. Sie können Ports und Protokolle festlegen, um den ein- und ausgehenden Datenverkehr zu ermöglichen.

Die Regeln für die Sicherheitsgruppen, die Ihrem Load Balancer zugeordnet sind, müssen den Datenverkehr in beiden Richtungen auf den Listener- und Zustandsprüfungs-Ports zulassen. Wenn Sie einen Listener zu einem Load Balancer hinzufügen oder den Zustandsprüfungs-Port für eine Zielgruppe aktualisieren, müssen Sie Ihre Sicherheitsgruppenregeln überprüfen, um sicherzustellen, dass sie den Datenverkehr auf dem neuen Port in beiden Richtungen zulassen. Weitere Informationen finden Sie unter [Empfohlene Regeln](#).

Load Balancer-Status

Ein Load Balancer kann einen der folgenden Status aufweisen:

provisioning

Der Load Balancer ist eingerichtet.

active

Der Load Balancer ist vollständig eingerichtet und kann den Datenverkehr weiterleiten.

active_impaired

Der Load Balancer leitet den Datenverkehr weiter, verfügt aber nicht über die notwendigen Ressourcen für die Skalierung.

failed

Der Load Balancer konnte nicht eingerichtet werden.

Load Balancer-Attribute

Dies sind die Load Balancer-Attribute:

`access_logs.s3.enabled`

Gibt an, ob in Amazon S3 gespeicherte Zugriffsprotokolle aktiviert sind. Der Standardwert ist `false`.

`access_logs.s3.bucket`

Der Name des Amazon-S3-Buckets für die Zugriffsprotokolle. Dieses Attribut ist erforderlich, wenn Zugriffsprotokolle aktiviert sind. Weitere Informationen finden Sie unter [Aktivieren der Zugriffsprotokolle](#).

`access_logs.s3.prefix`

Das Präfix für den Speicherort im Amazon-S3-Bucket.

`client_keep_alive.seconds`

Der Keepalive-Wert des Clients in Sekunden. Die Standardeinstellung ist 3600 Sekunden.

`deletion_protection.enabled`

Gibt an, ob der Löschschutz aktiviert ist. Der Standardwert ist `false`.

`idle_timeout.timeout_seconds`

Der Timeoutwert für die Leerlaufzeit in Sekunden. Standardmäßig ist ein Zeitraum von 60 Sekunden festgelegt.

`ipv6.deny_all_igw_traffic`

Sperrt den Zugriff des Internet-Gateways (IGW) auf den Load Balancer und verhindert so unbeabsichtigten Zugriff auf Ihren internen Load Balancer über ein Internet-Gateway. Es ist auf für mit dem Internet verbundenen Load Balancern auf `false` und für interne Load Balancern auf `true` eingestellt. Dieses Attribut verhindert nicht den Internetzugriff außerhalb von IGW (z. B. über Peering, Transit Gateway AWS Direct Connect, oder). AWS VPN

`routing.http.desync_mitigation_mode`

Legt fest, wie der Load Balancer Anforderungen verarbeitet, die ein Sicherheitsrisiko für Ihre Anwendung darstellen könnten. Die möglichen Werte sind `monitor`, `defensive` und `strictest`. Der Standardwert ist `defensive`.

`routing.http.drop_invalid_header_fields.enabled`

Gibt an, ob HTTP-Header mit ungültigen Header-Feldern vom Load Balancer entfernt (`true`) oder an Ziele weitergeleitet werden (`false`). Der Standardwert ist `false`. Elastic Load Balancing

erfordert, dass gültige HTTP-Header-Namen dem regulären Ausdruck `[-A-Za-z0-9]+` entsprechen, wie in der HTTP Field Name Registry beschrieben. Jeder Name besteht aus alphanumerischen Zeichen oder Bindestrichen. Wählen Sie `true` aus, wenn Sie möchten, dass HTTP-Header, die diesem Muster nicht entsprechen, aus Anforderungen entfernt werden sollen.

`routing.http.preserve_host_header.enabled`

Gibt an, ob der Application Load Balancer den Host-Header in der HTTP-Anforderung beibehalten und unverändert an das Ziel senden soll. Die möglichen Werte sind `true` und `false`. Der Standardwert ist `false`.

`routing.http.x_amzn_tls_version_and_cipher_suite.enabled`

Gibt an, ob die beiden Header (`x-amzn-tls-version` und `x-amzn-tls-cipher-suite`), die Informationen über die ausgehandelte TLS-Version und die Verschlüsselungssammlung enthalten, der Client-Anforderung hinzugefügt werden, bevor sie an das Ziel gesendet wird. Der `x-amzn-tls-version`-Header enthält Informationen über die mit dem Client ausgehandelte TLS-Protokollversion und der `x-amzn-tls-cipher-suite`-Header enthält Informationen über die mit dem Client ausgehandelte Verschlüsselungssammlung. Beide Header sind im OpenSSL-Format. Die möglichen Werte für dieses Attribut sind `true` und `false`. Der Standardwert ist `false`.

`routing.http.xff_client_port.enabled`

Zeigt an, ob der X-Forwarded-For-Header den Quellport beibehalten sollte, den der Client für die Verbindung mit dem Load Balancer verwendet hat. Die möglichen Werte sind `true` und `false`. Der Standardwert ist `false`.

`routing.http.xff_header_processing.mode`

Ermöglicht das Ändern, Beibehalten oder Entfernen der X-Forward-For-Header in der HTTP-Anforderung, bevor der Application Load Balancer die Anforderung an das Ziel sendet. Die möglichen Werte sind `append`, `preserve` und `remove`. Der Standardwert ist `append`.

- Wenn der Wert `append` ist, fügt der Application Load Balancer die Client-IP-Adresse (des letzten Hop) zum X-Forward-For-Header in der HTTP-Anforderung hinzu, bevor sie an Ziele gesendet wird.
- Wenn der Wert `preserve` ist, behält Application Load Balancer den X-Forward-For-Header in der HTTP-Anforderung und sendet sie ohne Änderung an Ziele.
- Wenn der Wert `remove` ist, entfernt Application Load Balancer den X-Forward-For-Header in der HTTP-Anforderung, bevor sie an Ziele gesendet wird.

`routing.http2.enabled`

Gibt an, ob HTTP/2 aktiviert ist. Der Standardwert ist `true`.

`waf.fail_open.enabled`

Gibt an, ob ein Load Balancer mit AWS WAF aktiviertem Load Balancer Anfragen an Ziele weiterleiten darf, wenn er die Anfrage nicht weiterleiten kann. AWS WAF Die möglichen Werte sind `true` und `false`. Der Standardwert ist `false`.

Note

Das `routing.http.drop_invalid_header_fields.enabled`-Attribut wurde eingeführt, um einen Schutz vor HTTP-Desync-Angriffen zu bieten. Das `routing.http.desync_mitigation_mode`-Attribut wurde hinzugefügt, um Ihren Anwendungen einen umfassenderen Schutz vor HTTP-Desync-Angriffen zu bieten. Sie müssen nicht beide Attribute verwenden und können je nach den Anforderungen Ihrer Anwendung auch nur eines der beiden auswählen.

IP-Adresstyp

Sie können die IP-Adresstypen festlegen, die Clients verwenden können, um auf Ihre Load Balancer, die mit dem Internet verbunden sind, und Ihre internen Load Balancer zuzugreifen.

Application Load Balancers unterstützen die folgenden IP-Adresstypen:

ipv4

Clients müssen über IPv4-Adressen (z. B. 192.0.2.1) mit dem Load Balancer verbunden werden.

dualstack

Clients können sich mit dem Load Balancer sowohl über IPv4-Adressen (z. B. 192.0.2.1) als auch über IPv6-Adressen (z. B. 2001:0db8:85a3:0:0:8a2e:0370:7334) verbinden.

Überlegungen

- Der Load Balancer kommuniziert mit Zielen auf der Grundlage des IP-Adresstyps der Zielgruppe.

- Wenn Sie den Dualstack-Modus für den Load Balancer aktivieren, stellt Elastic Load Balancing einen AAAA-DNS-Eintrag für den Load Balancer bereit. Clients, die mit dem Load Balancer über IPv4-Adressen kommunizieren, lösen den A-DNS-Eintrag auf. Clients, die mit dem Load Balancer über IPv6-Adressen kommunizieren, lösen den AAAA-DNS-Eintrag auf.
- Der Zugriff auf Ihre internen Dualstack-Load-Balancer über das Internet-Gateway ist blockiert, um einen unbeabsichtigten Internetzugriff zu verhindern. Dies verhindert jedoch nicht den Internetzugang, der nicht von IGW stammt (z. B. über Peering, Transit Gateway AWS Direct Connect, oder). AWS VPN

dualstack-without-public-ipv4

Clients müssen über IPv6-Adressen (z. B. 2001:0 db 8:85 a 3:0:0:0:8 a2e: 0370:7334) eine Verbindung zum Load Balancer herstellen.

Überlegungen

- Die Application Load Balancer Balancer-Authentifizierung unterstützt IPv4 nur, wenn eine Verbindung zu einem Identity Provider (IdP) oder Amazon Cognito Cognito-Endpunkt hergestellt wird. Ohne eine öffentliche IPv4-Adresse kann der Load Balancer den Authentifizierungsprozess nicht abschließen, was zu HTTP 500-Fehlern führt.

Weitere Hinweise zu IP-Adresstypen finden Sie unter [IP-Adresstypen für Ihren Application Load Balancer](#)

Ressourcenübersicht für Application Load Balancer

Die Application Load Balancer Balancer-Ressourcenübersicht bietet eine interaktive Darstellung der Architektur Ihres Load Balancers, einschließlich der zugehörigen Listener, Regeln, Zielgruppen und Ziele. In der Ressourcenübersicht werden auch die Beziehungen und Routingpfade zwischen allen Ressourcen hervorgehoben, sodass die Konfiguration Ihres Load Balancers visuell dargestellt wird.

So zeigen Sie die Ressourcenübersicht Ihres Application Load Balancers mithilfe der Konsole an

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Wählen Sie die Registerkarte Ressourcenübersicht, um die Ressourcenübersicht des Load Balancers anzuzeigen.

Komponenten der Ressourcenübersicht

Kartenansichten

In der Application Load Balancer Balancer-Ressourcenübersicht sind zwei Ansichten verfügbar: Overview und Unhealthy Target Map. Die Option „Übersicht“ ist standardmäßig ausgewählt und zeigt alle Ressourcen Ihres Load Balancers an. Wenn Sie die Kartenansicht für fehlerhafte Ziele auswählen, werden nur die fehlerhaften Ziele und die ihnen zugewiesenen Ressourcen angezeigt.

Die Ansicht Unhealthy Target Map kann verwendet werden, um Fehler bei Zielen zu beheben, bei denen die Zustandsprüfungen nicht bestanden wurden. Weitere Informationen finden Sie unter [Beheben Sie fehlerhafte Ziele mithilfe der Ressourcenübersicht](#).

Ressourcengruppen

Die Application Load Balancer Balancer-Ressourcenübersicht enthält vier Ressourcengruppen, eine für jeden Ressourcentyp. Die Ressourcengruppen sind Listener, Regeln, Zielgruppen und Ziele.

Ressourcenkacheln

Jede Ressource innerhalb einer Gruppe hat ihre eigene Kachel, auf der Details zu dieser bestimmten Ressource angezeigt werden.

- Wenn Sie den Mauszeiger über eine Ressourcenkachel bewegen, werden die Beziehungen zwischen dieser und anderen Ressourcen hervorgehoben.
- Wenn Sie eine Ressourcenkachel auswählen, werden die Beziehungen zwischen ihr und anderen Ressourcen hervorgehoben und zusätzliche Details zu dieser Ressource angezeigt.
 - Regelbedingungen: Die Bedingungen für jede Regel.
 - Zusammenfassung des Gesundheitszustands der Zielgruppe: Die Anzahl der registrierten Ziele für jeden Gesundheitsstatus.
 - Zielgesundheitsstatus Der aktuelle Gesundheitszustand und die Beschreibung des Ziels.

Note

Sie können die Option „Ressourcendetails anzeigen“ deaktivieren, um zusätzliche Details in der Ressourcenübersicht auszublenden.

- Jede Ressourcenkachel enthält einen Link, der, wenn er ausgewählt ist, zur Detailseite der Ressource navigiert.

- Listener - Wählen Sie den Listener protocol:port aus. Beispiel: HTTP:80
- Regeln - Wählen Sie die Regelaktion aus. Beispiel: Forward to target group
- Zielgruppen - Wählen Sie den Namen der Zielgruppe aus. Beispiel: my-target-group
- Ziele - Wählen Sie die Ziel-ID aus. Beispiel: i-1234567890abcdef0

Exportieren Sie die Ressourcenübersicht

Wenn Sie Exportieren auswählen, haben Sie die Möglichkeit, die aktuelle Ansicht der Ressourcenübersicht Ihres Application Load Balancers als PDF zu exportieren.

Load Balancer-Verbindungen

Bei der Verarbeitung einer Anfrage unterhält der Load Balancer zwei Verbindungen: eine Verbindung mit dem Client und eine Verbindung mit einem Ziel. Die Verbindung zwischen dem Load Balancer und dem Client wird auch als Front-End-Verbindung bezeichnet. Die Verbindung zwischen dem Load Balancer und dem Ziel wird auch als Back-End-Verbindung bezeichnet.

Zeitlimit für Verbindungsleerlauf

Das Timeout bei der Verbindungsinaktivität ist der Zeitraum, in dem eine bestehende Client- oder Zielverbindung inaktiv bleiben kann, ohne dass Daten gesendet oder empfangen werden, bevor der Load Balancer die Verbindung schließt.

Um sicherzustellen, dass langwierige Vorgänge wie Datei-Uploads rechtzeitig abgeschlossen werden können, senden Sie vor Ablauf jedes Leerlauf-Timeouts mindestens 1 Byte an Daten und erhöhen Sie die Dauer des Leerlauf-Timeouts nach Bedarf. Wir empfehlen außerdem, das Leerlaufzeitlimit für Ihre Anwendung auf einen höheren Wert als das Leerlaufzeitlimit für den Load Balancer festzulegen. Andernfalls könnte der Load Balancer, wenn die Anwendung die TCP-Verbindung zum Load Balancer nicht ordnungsgemäß schließt, eine Anforderung an die Anwendung senden, bevor er das Paket empfängt, um anzugeben, dass die Verbindung geschlossen ist. Wenn dies der Fall ist, sendet der Load Balancer einen HTTP-502-Bad-Gateway-Fehler an den Client.

Standardmäßig legt Elastic Load Balancing den Leerlauf-Timeout-Wert für Ihren Load Balancer auf 60 Sekunden oder 1 Minute fest. Gehen Sie folgendermaßen vor, um einen anderen Leerlauf-Timeoutwert festzulegen.

Um den Wert für das Leerlauf-Timeout der Verbindung mithilfe der Konsole zu aktualisieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Geben Sie unter Verkehrskonfiguration einen Wert für das Timeout bei Verbindungsinaktivität ein. Der gültige Bereich liegt zwischen 1 und 4000 Sekunden.
6. Wählen Sie Änderungen speichern aus.

Um den Wert für das Leerlauf-Timeout mit dem AWS CLI

Verwenden Sie den Befehl [modify-load-balancer-attributes](#) mit dem Attribut `idle_timeout.timeout_seconds`.

Keepalive-Dauer des HTTP-Clients

Die Keepalive-Dauer des HTTP-Clients ist die maximale Zeitdauer, für die ein Application Load Balancer eine persistente HTTP-Verbindung zu einem Client aufrechterhält. Nach Ablauf der konfigurierten Keepalive-Dauer des HTTP-Clients akzeptiert der Application Load Balancer eine Anfrage und gibt eine Antwort zurück, mit der die Verbindung ordnungsgemäß geschlossen wird.

Die Art der vom Load Balancer gesendeten Antwort hängt von der HTTP-Version ab, die von der Client-Verbindung verwendet wird. Für Clients, die über HTTP 1.x verbunden sind, sendet der Load Balancer einen HTTP-Header, der das Feld enthält. `Connection: close` Für Clients, die über HTTP/2 verbunden sind, sendet der Load Balancer einen Frame. `GOAWAY`

Standardmäßig setzen Application Load Balancers den Wert für die Keepalive-Dauer des HTTP-Clients auf 3600 Sekunden oder 1 Stunde. Die Keepalive-Dauer des HTTP-Clients kann nicht ausgeschaltet oder unter den Mindestwert von 60 Sekunden gesetzt werden. Sie können die Keepalive-Dauer des HTTP-Clients jedoch auf maximal 604.800 Sekunden oder 7 Tage erhöhen. Der Application Load Balancer beginnt mit der Dauer der HTTP-Client-Keepalive-Dauer, wenn eine HTTP-Verbindung zu einem Client zum ersten Mal hergestellt wird. Die Dauer läuft weiter, wenn kein Datenverkehr vorhanden ist, und wird erst zurückgesetzt, wenn eine neue Verbindung hergestellt ist.

Note

Wenn Sie den IP-Adresstyp Ihres Application Load Balancers auf den Load Balancer umstellen, wartet `dualstack-without-public-ipv4` der Load Balancer, bis alle aktiven Verbindungen abgeschlossen sind. Um den Zeitaufwand für die Umstellung des IP-Adresstyps Ihres Application Load Balancers zu verringern, sollten Sie die Dauer der HTTP-Client-Keepalive-Dauer verringern.

Der Application Load Balancer weist dem HTTP-Client bei der ersten Verbindung einmal eine Keepalive-Dauer zu. Wenn die Keepalive-Dauer des HTTP-Clients aktualisiert wird, kann dies zu gleichzeitigen Verbindungen mit unterschiedlichen Werten für die Keepalive-Dauer des HTTP-Clients führen. Bei bestehenden Verbindungen wird der Wert für die Keepalive-Dauer des HTTP-Clients beibehalten, der bei der ersten Verbindung angewendet wurde, während alle neuen Verbindungen den aktualisierten Wert für die Keepalive-Dauer des HTTP-Clients erhalten.

Um den Wert für die Keepalive-Dauer des Clients mithilfe der Konsole zu aktualisieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Geben Sie unter Verkehrskonfiguration einen Wert für die Keep-Alive-Dauer des HTTP-Clients ein. Der gültige Bereich liegt zwischen 60 und 604800 Sekunden.
6. Wählen Sie Änderungen speichern aus.

Um den Wert für die Keepalive-Dauer des Clients zu aktualisieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-load-balancer-attributes](#) mit dem Attribut `client_keep_alive.seconds`.

Zonenübergreifendes Load Balancing

Bei Application Load Balancern ist zonenübergreifendes Load Balancing standardmäßig aktiviert und kann nicht auf Load-Balancer-Ebene geändert werden. Weitere Informationen finden Sie im Abschnitt [Zonenübergreifender Load Balancing](#) im Benutzerhandbuch für Elastic Load Balancing.

Das Deaktivieren des zonenübergreifenden Load Balancings ist auf Zielgruppenebene möglich. Weitere Informationen finden Sie unter [the section called “Wenn zonenübergreifendes Load Balancing deaktivieren”](#).

Löschschutz

Um zu verhindern, dass der Load Balancer versehentlich gelöscht wird, können Sie den Löschschutz aktivieren. Standardmäßig ist der Löschschutz für Ihren Load Balancer deaktiviert.

Wenn Sie den Löschschutz für Ihren Load Balancer aktivieren, müssen Sie ihn deaktivieren, bevor Sie den Load Balancer löschen.

So aktivieren Sie mithilfe der Konsole den Löschschutz:

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Aktivieren Sie unter Konfiguration die Option Löschschutz.
6. Wählen Sie Änderungen speichern aus.

So deaktivieren Sie mithilfe der Konsole den Löschschutz:

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Schalten Sie auf der Seite Konfiguration den Löschschutz aus.
6. Wählen Sie Änderungen speichern aus.

Um den Löschschutz zu aktivieren oder zu deaktivieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-load-balancer-attributes](#) mit dem Attribut `deletion_protection.enabled`.

Desynchroner Mitigationsmodus

Der desynchrone Mitigationsmodus schützt Ihre Anwendung vor Problemen aufgrund von HTTP-Desync-Angriffen. Der Load Balancer klassifiziert jede Anforderung anhand ihrer Bedrohungsstufe, lässt sichere Anforderungen zu und mindert dann das Risiko gemäß dem von Ihnen angegebenen Mitigationsmodus. Die desynchronen Mitigationsmodi lauten „Überwachen“, „Defensiv“ und „Am strengsten“. Der Standardmodus ist „Defensiv“, der eine dauerhafte Abwehr gegen HTTP-Desync-Angriffe bietet und gleichzeitig die Verfügbarkeit Ihrer Anwendung gewährleistet. Sie können in den Modus „Am strengsten“ wechseln, um sicherzustellen, dass Ihre Anwendung nur Anforderungen empfängt, die [RFC 7230](#) entsprechen.

Die Bibliothek „http_desync_guardian“ analysiert HTTP-Anforderungen, um HTTP-Desync-Angriffe zu verhindern. Weitere Informationen finden Sie unter [HTTP Desync Guardian](#) auf GitHub

Klassifizierungen

Diese Klassifizierungen lauten wie folgt:

- Konform – Die Anforderung entspricht RFC 7230 und stellt keine bekannten Sicherheitsbedrohungen dar.
- Akzeptabel – Die Anforderung entspricht nicht RFC 7230, stellt jedoch keine bekannten Sicherheitsbedrohungen dar.
- Mehrdeutig – Die Anforderung entspricht nicht RFC 7230, stellt jedoch ein Risiko dar, da verschiedene Webserver und Proxys sie unterschiedlich behandeln könnten.
- Schwerwiegend – Die Anforderung stellt ein hohes Sicherheitsrisiko dar. Der Load Balancer blockiert die Anforderung, sendet dem Client eine 400-Antwort und schließt die Client-Verbindung.

Wenn eine Anforderung nicht RFC 7230 entspricht, erhöht der Load Balancer die `DesyncMitigationMode_NonCompliant_Request_Count`-Metrik. Weitere Informationen finden Sie unter [Application-Load-Balancer-Metriken](#).

Die Klassifizierung für jede Anforderung ist in den Load-Balancer-Zugriffsprotokollen enthalten. Wenn die Anforderung nicht entspricht, enthalten die Zugriffsprotokolle einen Ursachencode für die Klassifizierung. Weitere Informationen finden Sie unter [Gründe für die Klassifizierung](#).

Modi

In der folgenden Tabelle wird beschrieben, wie Application Load Balancer Anforderungen basierend auf Modus und Klassifizierung behandeln.

Klassifizierung	Modus „Überwachen“	Modus „Defensiv“	Modus „Am strengsten“
Konform	Zulässig	Zulässig	Zulässig
Akzeptabel	Zulässig	Zulässig	Blocked
Mehrdeutig	Zulässig	Zulässig ¹	Blocked
Schwerwiegend	Zulässig	Blocked	Blocked

¹ Leitet die Anforderungen weiter, schließt aber die Client- und Zielverbindungen. Es können zusätzliche Gebühren anfallen, wenn Ihr Load Balancer im Modus „Defensiv“ eine große Anzahl von mehrdeutigen Anforderungen empfängt. Dies liegt daran, dass die erhöhte Anzahl neuer Verbindungen pro Sekunde dazu beiträgt, dass die Load-Balancer-Kapazitätseinheiten (LCU) pro Stunde verwendet werden. Sie können die `NewConnectionCount`-Metrik verwenden, um zu vergleichen, wie Ihr Load Balancer im Modus „Überwachen“ und im Modus „Defensiv“ neue Verbindungen herstellt.

So aktualisieren Sie den desynchronen Mitigationsmodus über die Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Wählen Sie unter Paketverarbeitung für Desynchroner Mitigationsmodus die Option Defensiv, Am strengsten oder Überwachen aus.
6. Wählen Sie Änderungen speichern aus.

Um den Desync-Mitigationsmodus zu aktualisieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-load-balancer-attributes](#) mit dem Attribut `routing.http.desync_mitigation_mode`, das auf `monitor`, `defensive` oder `strictest` festgelegt ist.

Beibehalten der Host-Header

Wenn Sie das Attribut Beibehalten des Host-Headers aktivieren, behält Application Load Balancer den Host-Header der HTTP-Anforderung bei und sendet den Header ohne Änderung an Ziele.

Wenn der Application Load Balancer mehrere Host-Header empfängt, behält er sie alle bei. Listener-Regeln werden nur auf den ersten empfangenen Host-Header angewendet.

Wenn das Attribut Beibehalten des Host-Headers nicht aktiviert ist, ändert der Application Load Balancer den Host-Header standardmäßig wie folgt:

Wenn „Beibehalten des Host-Headers“ nicht aktiviert ist und der Listener-Port kein Standardport ist: Wenn die Standardports (Port 80 oder 443) nicht verwendet werden, hängen wir die Portnummer an den Host-Header an, sofern sie nicht bereits vom Client hinzugefügt wurde. Zum Beispiel würde der Host-Header in der HTTP-Anforderung mit Host: `www.example.com` in Host: `www.example.com:8080` geändert werden, wenn der Listener-Port ein nicht standardmäßiger Port ist, wie z. B. 8080.

Wenn „Beibehalten des Host-Headers“ nicht aktiviert ist und der Listener-Port ein Standardport ist (Port 80 oder 443): Bei Standard-Listener-Ports (entweder Port 80 oder 443) fügen wir die Portnummer nicht an den ausgehenden Host-Header an. Jede Portnummer, die sich bereits im eingehenden Host-Header befand, wird entfernt.

Die folgende Tabelle zeigt weitere Beispiele dafür, wie Application Load Balancer die Host-Header in der HTTP-Anforderung auf der Grundlage des Listener-Ports behandeln.

Listener-Port	Beispielanforderung	Host-Header der Anforderung	„Beibehaltung des Host-Headers“ ist deaktiviert (Standardverhalten)	„Beibehaltung des Host-Headers“ ist aktiviert
Die Anforderung wird über den standardmäßigen HTTP/HTTPS-Listener gesendet.	GET / index.html HTTP/1.1 Host: example.com	example.com	example.com	example.com

Listener-Port	Beispielanforderung	Host-Header der Anforderung	„Beibehaltung des Host-Headers“ ist deaktiviert (Standardverhalten)	„Beibehaltung des Host-Headers“ ist aktiviert
Die Anfrage wird über den Standard-HTTP-Listener gesendet und der Host-Header hat einen Port (z. B. 80 oder 443).	GET / index.html HTTP/1.1 Host: example.com:80	example.com:80	example.com	example.com:80
Die Anforderung hat einen absoluten Pfad.	GET https:// dns_name/ index.html HTTP/1.1 Host: example.com	example.com	dns_name	example.com
Die Anfrage wird über einen nicht standardmäßigen Listener-Port gesendet (z. B. 8080)	GET / index.html HTTP/1.1 Host: example.com	example.com	example.com:8080	example.com

Listener-Port	Beispielanforderung	Host-Header der Anforderung	„Beibehaltung des Host-Headers“ ist deaktiviert (Standardverhalten)	„Beibehaltung des Host-Headers“ ist aktiviert
Die Anforderung wird über einen nicht standardmäßigen Listener-Port gesendet und der Host-Header enthält einen Port (z. B. 8080).	GET / index.html HTTP/1.1 Host: example.com:8080	example.com:8080	example.com:8080	example.com:8080

Um die Aufbewahrung von Host-Headern mithilfe der Konsole zu aktivieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie im Navigationsbereich auf Load Balancers.
3. Wählen Sie den Load Balancer aus.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Aktivieren Sie unter Paketverarbeitung die Option Beibehaltung des Host-Headers.
6. Wählen Sie Änderungen speichern aus.

Um die Aufbewahrung von Host-Headern zu aktivieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-load-balancer-attributes](#) mit dem Attribut `routing.http.preserve_host_header.enabled`, das auf `true` festgelegt ist.

Application Load Balancers und 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 unter [Arbeiten mit Web-ACLs](#) im AWS WAF - Entwicklerhandbuch.

Wenn der Load Balancer keine Antwort von erhalten kann AWS WAF, gibt er standardmäßig einen HTTP 500-Fehler zurück und leitet die Anfrage nicht weiter. Wenn Sie möchten, dass Ihr Load Balancer Anfragen an Ziele weiterleitet, auch wenn er keinen Kontakt herstellen kann AWS WAF, können Sie die Integration aktivieren AWS WAF . Um zu überprüfen, ob Ihr Load Balancer integriert ist AWS WAF, wählen Sie Ihren Load Balancer auf der AWS Management Console Registerkarte Integrierte Dienste aus.

Vordefinierte Web-ACLs

Wenn Sie die AWS WAF Integration aktivieren, können Sie wählen, ob automatisch eine neue Web-ACL mit vordefinierten Regeln erstellt werden soll. Die vordefinierte Web-ACL umfasst drei AWS verwaltete Regeln, die Schutz vor den häufigsten Sicherheitsbedrohungen bieten.

- **AWSManagedRulesAmazonIpReputationList**- Die Regelgruppe der Amazon IP-Reputationsliste blockiert IP-Adressen, die typischerweise mit Bots oder anderen Bedrohungen in Verbindung stehen. Weitere Informationen finden Sie unter [verwaltete Regelgruppe der Amazon IP-Reputationsliste](#) im AWS WAF Entwicklerhandbuch.
- **AWSManagedRulesCommonRuleSet**- Die Regelgruppe Core Rule Set (CRS) bietet Schutz vor der Ausnutzung einer Vielzahl von Sicherheitslücken, darunter einige der risikoreichen und häufig auftretenden Sicherheitslücken, die in OWASP-Publikationen wie [OWASP](#) Top 10 beschrieben werden. Weitere Informationen finden Sie unter [verwaltete Regelgruppe Core Rule Set \(CRS\)](#) im Entwicklerhandbuch.AWS WAF
- **AWSManagedRulesKnownBadInputsRuleSet**- Die Regelgruppe Bekannte fehlerhafte Eingaben blockiert Anforderungsmuster, die bekanntermaßen ungültig sind und mit der Ausnutzung oder Entdeckung von Sicherheitslücken in Verbindung stehen. Weitere Informationen finden Sie unter [Verwaltete Regelgruppe „Bekannte fehlerhafte Eingaben“](#) im AWS WAF Entwicklerhandbuch.

Um die AWS WAF Verwendung der Konsole zu aktivieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Erweitern Sie auf der Registerkarte Integrationen die Option AWS Web Application Firewall (WAF) und wählen Sie Einer WAF-Web-ACL zuordnen aus.

5. Wählen Sie unter Web-ACL die Option Vordefinierte Web-ACL automatisch erstellen oder wählen Sie eine vorhandene Web-ACL aus.
6. Wählen Sie unter Regelaktion die Option Blockieren oder Zählen aus.
7. Wählen Sie Bestätigen aus.

Um AWS WAF Fail Open zu aktivieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-load-balancer-attributes](#) mit dem Attribut `waf.fail_open.enabled`, das auf `true` festgelegt ist.

Erstellen eines Application Load Balancers

Ein Load Balancer empfängt Anfragen von Clients und leitet sie über Ziele in einer Zielgruppe weiter.

Bevor Sie beginnen, stellen Sie sicher, dass Sie über eine Virtual Private Cloud (VPC) mit mindestens einem öffentlichen Subnetz in jeder der Zonen verfügen, die von Ihren Zielen verwendet werden. Weitere Informationen finden Sie unter [the section called “Subnetze für Ihren Load Balancer”](#).

Informationen zum Erstellen eines Load Balancers mit dem finden Sie AWS CLI unter [Tutorial: Erstellen eines Application Load Balancers mithilfe der AWS CLI](#).

Um einen Load Balancer mit dem zu erstellen AWS Management Console, führen Sie die folgenden Aufgaben aus.

Aufgaben

- [Schritt 1: Konfigurieren einer Zielgruppe](#)
- [Schritt 2: Registrieren von Zielen](#)
- [Schritt 3: Konfigurieren von Load Balancer und Listener](#)
- [Schritt 4: Testen des Load Balancers](#)

Schritt 1: Konfigurieren einer Zielgruppe

Durch Konfigurieren einer Zielgruppe können Sie Ziele wie EC2-Instances registrieren. Die Zielgruppe, die Sie in diesem Schritt konfigurieren, wird als Zielgruppe in der Listener-Regel verwendet, wenn Sie Ihren Load Balancer konfigurieren. Weitere Informationen finden Sie unter [Zielgruppen für Ihre Application Load Balancer](#).

Um Ihre Zielgruppe mithilfe der Konsole zu konfigurieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Target Groups aus.
3. Wählen Sie Zielgruppe erstellen aus.
4. Legen Sie im Abschnitt Grundlegende Konfiguration die folgenden Parameter fest:
 - a. Wählen Sie unter Zieltyp wählen die Option Instances aus, um Ziele nach Instance-ID anzugeben, oder wählen Sie IP-Adressen aus, um Ziele nur nach IP-Adresse anzugeben. Wenn der Zieltyp eine Lambda-Funktion ist, können Sie Zustandsprüfungen aktivieren, indem Sie im Abschnitt Zustandsprüfungen die Option Aktivieren auswählen.
 - b. Geben Sie unter Zielgruppenname einen Namen für die Zielgruppe ein.
 - c. Ändern Sie Port und Protokoll nach Bedarf.
 - d. Wenn der Zieltyp Instances oder IP-Adressen ist, wählen Sie IPv4 oder IPv6 als IP-Adresstyp aus. Fahren Sie andernfalls mit dem nächsten Schritt fort.

Beachten Sie, dass nur Ziele, die den ausgewählten IP-Adresstyp haben, in diese Zielgruppe eingefügt werden können. Der IP-Adresstyp kann nicht geändert werden, nachdem die Zielgruppe erstellt wurde.

- e. Wählen Sie für VPC eine Virtual Private Cloud (VPC) mit den Zielen aus, die Sie Ihrer Zielgruppe hinzufügen möchten.
 - f. Wählen Sie für Protokollversion die Option HTTP1 aus, wenn das Anforderungsprotokoll HTTP/1.1 oder HTTP/2 ist. Wählen Sie HTTP2 aus, wenn das Anforderungsprotokoll HTTP/2 oder gRPC ist; und wählen Sie gRPC aus, wenn das Anforderungsprotokoll gRPC ist.
5. Behalten Sie im Abschnitt Zustandsprüfungen die Standardeinstellungen bei. Wählen Sie unter Erweiterte Zustandsprüfungseinstellungen den Port, die Anzahl, das Timeout und das Intervall für die Zustandsprüfung aus und geben Sie die Erfolgscodes an. Überschreitet die Anzahl der Zustandsprüfungen nacheinander den Schwellenwert für fehlerhaften Zustand, nimmt der Load Balancer das Ziel außer Betrieb. Überschreitet die Anzahl der Zustandsprüfungen nacheinander den Schwellenwert für fehlerfreien Zustand, nimmt der Load Balancer das Ziel wieder in Betrieb. Weitere Informationen finden Sie unter [Zustandsprüfungen für Ihre Zielgruppen](#).
6. (Optional) Fügen Sie einen oder mehrere Tags wie folgt hinzu:
 - a. Erweitern Sie den Abschnitt Tags.
 - b. Wählen Sie Add tag.

- c. Geben Sie den Schlüssel und den Wert für das Tag ein. Erlaubte Zeichen sind Buchstaben, Leerzeichen und Zahlen (in UTF-8) sowie die folgenden Sonderzeichen: + - = _ : / @. Verwenden Sie keine führenden oder nachgestellten Leerzeichen. Bei Tag-Werten muss die Groß- und Kleinschreibung beachtet werden.

7. Wählen Sie Weiter aus.

Schritt 2: Registrieren von Zielen

Sie können EC2-Instances, IP-Adressen oder Lambda-Funktionen als Ziele in einer Zielgruppe registrieren. Dies ist ein optionaler Schritt zum Erstellen eines Load Balancers. Sie müssen Ihre Ziele jedoch registrieren, um sicherzustellen, dass Ihr Load Balancer den Datenverkehr an sie weiterleitet.

1. Auf der Seite Ziele registrieren fügen Sie ein oder mehrere Ziele wie folgt hinzu:
 - Wenn der Zieltyp Instances ist, wählen Sie eine oder mehrere Instances aus, geben Sie einen oder mehrere Ports ein und wählen Sie dann Schließen Sie die unten angeführten als ausstehend ein aus.
 - Wenn der Zieltyp IP-Adressen lautet, gehen Sie wie folgt vor:
 - a. Wählen Sie eine Netzwerk-VPC in der Liste aus oder wählen Sie Andere private IP-Adresse aus.
 - b. Geben Sie die IP-Adresse manuell ein oder suchen Sie die IP-Adresse anhand der Instance-Details. Sie können bis zu fünf IP-Adressen gleichzeitig eingeben.
 - c. Geben Sie die Ports für die Weiterleitung des Datenverkehrs an die angegebenen IP-Adressen ein.
 - d. Wählen Sie Schließen Sie die unten angeführten als ausstehend ein aus.
 - Wenn der Zieltyp Lambda ist, wählen Sie eine Lambda-Funktion aus oder geben Sie einen Lambda-Funktions-ARN ein und wählen Sie dann Schließen Sie die unten angeführten als ausstehend ein aus.
2. Wählen Sie Zielgruppe erstellen aus.

Schritt 3: Konfigurieren von Load Balancer und Listener

Um einen Application Load Balancer zu erstellen, müssen Sie zunächst grundlegende Konfigurationsinformationen für Ihren Load Balancer angeben, z. B. einen Namen, ein Schema und einen IP-Adresstyp. Geben Sie anschließend Informationen über Ihr Netzwerk und einen oder

mehrere Listener an. Ein Listener ist ein Prozess, der Verbindungsanfragen überprüft. Er wird mit einem Protokoll und einem Port für Verbindungen von Clients zum Load Balancer konfiguriert. Weitere Informationen zu unterstützten Protokollen und Ports finden Sie unter [Listener-Konfiguration](#).

Um Ihren Load Balancer und Listener mithilfe der Konsole zu konfigurieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie im Navigationsbereich auf Load Balancers.
3. Klicken Sie auf Load Balancer erstellen.
4. Wählen Sie unter Application Load Balancer Create (Erstellen) aus.
5. Basiskonfiguration
 - a. Geben Sie im Feld Name des Load Balancers einen Namen für Ihren Load Balancer ein. z. B. **my-alb**. Der Name Ihres Application Load Balancers muss innerhalb Ihrer Gruppe von Application Load Balancers und Network Load Balancers für die Region eindeutig sein. Namen dürfen maximal 32 Zeichen lang sein und dürfen nur alphanumerische Zeichen und Bindestriche enthalten. Sie dürfen nicht mit einem Bindestrich oder mit `internal-` beginnen oder enden. Der Name Ihres Application Load Balancers kann nicht mehr geändert werden, sobald er erstellt wurde.
 - b. Wählen Sie für Scheme (Schema) entweder Internet-facing (Mit dem Internet verbunden) oder Internal (Intern) aus. Ein mit dem Internet verbundener Load Balancer leitet Anfragen von Clients über das Internet an Ziele weiter. Ein interner Load Balancer leitet Anforderungen an Ziele unter Verwendung privater IP-Adressen weiter.
 - c. Wählen Sie als IP-Adresstyp IPv4, Dualstack oder Dualstack ohne öffentliches IPv4. Wählen Sie IPv4, wenn Ihre Clients IPv4-Adressen für die Kommunikation mit dem Load Balancer verwenden. Wählen Sie Dualstack, wenn Ihre Clients sowohl IPv4- als auch IPv6-Adressen verwenden, um mit dem Load Balancer zu kommunizieren. Wählen Sie Dualstack ohne öffentliches IPv4, wenn Ihre Clients nur IPv6-Adressen für die Kommunikation mit dem Load Balancer verwenden.
6. Netzwerkzuordnung
 - a. Wählen Sie im Feld VPC die VPC aus, die Sie für Ihre EC2-Instances verwendet haben. Wenn Sie Mit dem Internet verbunden für Schema ausgewählt haben, stehen nur VPCs mit einem Internet-Gateway zur Auswahl.
 - b. Aktivieren Sie für Zuordnungen Zonen für Ihren Load Balancer, indem Sie wie folgt Subnetze auswählen:

- Subnetze aus zwei oder mehr Availability Zones
- Subnetze aus einer oder mehreren Local Zones
- Outpost-Subnetz

Weitere Informationen finden Sie unter [the section called “Subnetze für Ihren Load Balancer”](#).

Für interne Load Balancer werden die IPv4- und IPv6-Adressen aus dem Subnetz-CIDR zugewiesen.

Wenn Sie den Modus Dualstack für den Load Balancer aktiviert haben, wählen Sie Subnetze mit sowohl IPv4- als auch IPv6-CIDR-Blöcken aus.

7. Für die Security groups (Sicherheitsgruppen) können Sie eine vorhandene Sicherheitsgruppe auswählen oder eine neue erstellen.

Die Sicherheitsgruppe für Ihren Load Balancer ermöglicht die Kommunikation mit registrierten Zielen sowohl auf dem Listener-Port als auch auf dem Zustandsprüfungs-Port. Die Konsole kann eine Sicherheitsgruppe für Ihren Load Balancer mit Regeln erstellen, die diese Kommunikation zulassen. Sie können auch eine Sicherheitsgruppe erstellen und diese stattdessen auswählen. Weitere Informationen finden Sie unter [Empfohlene Regeln](#).

(Optional) Um eine neue Sicherheitsgruppe für Ihren Load Balancer zu erstellen, wählen Sie Create a new security group (Neue Sicherheitsgruppe erstellen) aus.

8. Bei Listener und Routing ist standardmäßig ein Listener eingestellt, der über Port 80 HTTP-Datenverkehr annimmt. Sie können das Standardprotokoll und den Standardport beibehalten oder andere auswählen. Wählen Sie für Default action (Standardaktion) die von Ihnen erstellte Zielgruppe aus. Sie können optional Add listener (Listener hinzufügen) auswählen, um einen weiteren Listener hinzuzufügen (z. B. einen HTTPS-Listener).
9. (Optional) Wenn Sie einen HTTPS-Listener verwenden

Wir empfehlen, dass Sie für Sicherheitsrichtlinie immer die neueste vordefinierte Sicherheitsrichtlinie verwenden.

- a. Für Standard-SSL-/TLS-Zertifikat sind die folgenden Optionen verfügbar:

- Wenn Sie ein Zertifikat mit erstellt oder importiert haben AWS Certificate Manager, wählen Sie Aus ACM und dann das Zertifikat unter Zertifikat auswählen aus.

- Wenn Sie ein Zertifikat mit IAM importiert haben, wählen Sie Von ACM aus und wählen Sie dann das Zertifikat unter Zertifikat auswählen aus.
 - Wenn Sie ein Zertifikat importieren, aber ACM in Ihrer Region nicht verfügbar ist, wählen Sie Importieren und dann An IAM aus. Geben Sie im Feld Zertifikatname den Namen des Zertifikats ein. Kopieren Sie den Inhalt der privaten Schlüsseldatei (PEM-kodiert) und fügen Sie ihn in das Feld Privater Zertifikatsschlüssel ein. Kopieren Sie den Inhalt der öffentlichen Schlüsselzertifikatdatei (PEM-kodiert) und fügen Sie ihn in das Feld Zertifikatstext ein. Kopieren Sie den Inhalt der Zertifikatskettendatei (PEM-kodiert) und fügen Sie ihn in das Feld Certificate Chain (Zertifikats-Kette) ein, es sei denn, Sie verwenden ein selbst signiertes Zertifikat und es ist nicht wichtig, dass Browser das Zertifikat implizit akzeptieren.
- b. (Optional) Um die gegenseitige Authentifizierung zu aktivieren, aktivieren Sie unter Verwaltung von Client-Zertifikaten die Option Gegenseitige Authentifizierung (mTLS).

Wenn diese Option aktiviert ist, ist der standardmäßige Modus für gegenseitiges TLS Passthrough.

Wenn Sie „Mit Trust Store verifizieren“ auswählen:

- Standardmäßig werden Verbindungen mit abgelaufenen Client-Zertifikaten abgelehnt. Um dieses Verhalten zu ändern, erweitern Sie die erweiterten mTLS-Einstellungen und wählen Sie dann unter Ablauf des Client-Zertifikats die Option Abgelaufene Client-Zertifikate zulassen aus.
 - Wählen Sie unter Trust Store einen vorhandenen Trust Store aus, oder wählen Sie Neuer Trust Store aus.
 - Wenn Sie Neuer Vertrauensspeicher ausgewählt haben, geben Sie einen Namen für den Vertrauensspeicher, den Speicherort der S3-URI-Zertifizierungsstelle und optional einen Speicherort für die Sperrliste für S3-URI-Zertifikate an.
10. (Optional) Sie können während der Erstellung unter Mit Dienstintegrationen optimieren andere Dienste in Ihren Load Balancer integrieren.
- Sie können wählen, ob Sie AWS WAF Sicherheitsvorkehrungen für Ihren Load Balancer mit einer vorhandenen oder automatisch erstellten Web-ACL einbeziehen möchten. [Nach der Erstellung können Web-ACLs in der Konsole verwaltet werden.](#) [AWS WAF](#) Weitere Informationen finden Sie im [Entwicklerhandbuch unter Web-ACL mit einer AWS Ressource verknüpfen oder deren Zuordnung aufheben.](#) [AWS WAF](#)

- Sie können wählen, ob Sie einen Accelerator für Sie AWS Global Accelerator erstellen und Ihren Load Balancer mit dem Accelerator verknüpfen möchten. Der Name des Beschleunigers kann die folgenden Zeichen (bis zu 64 Zeichen) enthalten: a-z, A-Z, 0-9, . (Punkt) und - (Bindestrich). Nachdem der Accelerator erstellt wurde, können Sie ihn in der [AWS Global Accelerator Konsole](#) verwalten. Weitere Informationen finden Sie im [AWS Global Accelerator Entwicklerhandbuch unter Hinzufügen eines Accelerators beim Erstellen eines Load Balancers](#).

11. Hinzufügen von Tags und Erstellen

- (Optional) Sie können zwecks Kategorisierung Ihrem Load Balancer ein Tag hinzufügen. Tag-Schlüssel müssen für jeden Load Balancer eindeutig sein. Erlaubte Zeichen sind Buchstaben, Leerzeichen und Zahlen (in UTF-8) sowie die folgenden Sonderzeichen: + - = _ : / @. Verwenden Sie keine führenden oder nachgestellten Leerzeichen. Bei Tag-Werten muss die Groß- und Kleinschreibung beachtet werden.
- Überprüfen Sie Ihre Konfiguration und wählen Sie Load Balancer erstellen aus. Bei der Erstellung werden einige Standardattribute auf Ihren Load Balancer angewendet. Sie können sie nach der Erstellung des Load Balancers anzeigen und bearbeiten. Weitere Informationen finden Sie unter [Load Balancer-Attribute](#).

Schritt 4: Testen des Load Balancers

Nach der Erstellung Ihres Load Balancers können Sie sicherstellen, dass Ihre EC2-Instances die erste Zustandsprüfung bestehen. Anschließend können Sie überprüfen, ob der Load Balancer Datenverkehr an Ihre EC2-Instance sendet. Informationen zum Löschen des Load Balancers finden Sie unter [Löschen eines Application Load Balancers](#).

So testen Sie den Load Balancer

1. Nachdem der Load Balancer erstellt wurde, klicken Sie auf Close (Schließen).
2. Wählen Sie im Navigationsbereich Target Groups aus.
3. Wählen Sie die neu erstellte Zielgruppe aus.
4. Wählen Sie Targets und vergewissern Sie sich, dass die Instances bereit sind. Wenn der Status einer Instance `initial` lautet, liegt das in der Regel daran, dass sich die Instance noch im Registrierungsprozess befindet. Dieser Status kann auch darauf hindeuten, dass die Instance nicht die Mindestanzahl an Zustandsprüfungen nicht bestanden hat, um als fehlerfrei angesehen

zu werden. Wenn der Status von mindestens einer Instance fehlerfrei ist, können Sie Ihren Load Balancer testen. Weitere Informationen finden Sie unter [Zustandsstatus des Ziels](#).

5. Klicken Sie im Navigationsbereich auf Load Balancers.
6. Wählen Sie den neu erstellten Load Balancer aus.
7. Wählen Sie Beschreibung und kopieren Sie den DNS-Namen des mit dem Internet verbundenen oder internen Load Balancers (z. B. my-load-balancer -1234567890abcdef. elb.us-east-2.amazonaws.com).
 - Fügen Sie für mit dem Internet verbundene Load Balancer den DNS-Namen in das Adressfeld eines mit dem Internet verbundenen Webbrowsers ein.
 - Fügen Sie für interne Load Balancer den DNS-Namen in das Adressfeld eines Webbrowsers ein, der über private Konnektivität mit der VPC verbunden ist.

Wenn alles korrekt konfiguriert ist, zeigt der Browser die Standardseite Ihres Servers an.

8. Wenn die Webseite nicht angezeigt wird, finden Sie in den folgenden Dokumenten zusätzliche Hilfe für die Konfiguration und Schritte zur Fehlerbehebung.
 - Informationen zu Problemen in Bezug auf DNS finden Sie unter [Weiterleiten von Datenverkehr an einen ELB Load Balancer](#) im Entwicklerhandbuch von Amazon Route 53.
 - Informationen zu Problemen im Zusammenhang mit Load Balancer finden Sie unter [Fehlerbehebung bei Ihren Application Load Balancern](#).

Availability Zones für Ihren Application Load Balancer

Sie können die Availability Zones für Ihren Load Balancer jederzeit aktivieren oder deaktivieren. Nachdem Sie eine Availability Zone aktiviert haben, beginnt der Load Balancer, Anforderungen an die registrierten Ziele in der Availability Zone weiterzuleiten. Ihr Load Balancer ist am effektivsten, wenn Sie dafür sorgen, dass jede aktivierte Availability Zone mindestens ein registriertes Ziel hat.

Nachdem Sie eine Availability Zone deaktivieren, bleiben die Ziele in der Availability Zone beim Load Balancer registriert, aber der Load Balancer leitet keine Anforderungen an sie weiter.

So aktualisieren Sie Availability Zones mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.

3. Wählen Sie den Load Balancer aus.
4. Wählen Sie auf der Registerkarte Netzwerkzuordnung die Option Subnetze bearbeiten aus.
5. Um eine Availability Zone zu aktivieren, aktivieren Sie das entsprechende Kontrollkästchen und wählen Sie ein Subnetz aus. Wenn nur ein Subnetz verfügbar ist, ist es bereits für Sie ausgewählt.
6. Um das Subnetz für eine aktivierte Availability Zone zu ändern, wählen Sie eines der anderen Subnetze in der Liste aus.
7. Um eine Availability Zone zu deaktivieren, deaktivieren Sie das entsprechende Kontrollkästchen.
8. Wählen Sie Änderungen speichern aus.

Um Availability Zones zu aktualisieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [set-subnets](#).

Sicherheitsgruppen für Ihren Application Load Balancer

Die Sicherheitsgruppe für Ihren Application Load Balancer steuert den Datenverkehr, der den Load Balancer erreichen und verlassen darf. Sie müssen sicherstellen, dass der Load Balancer mit den registrierten Zielen sowohl auf dem Listener-Port als auch auf dem Zustandsprüfungs-Port kommunizieren kann. Wenn Sie einen Listener zum Load Balancer hinzufügen oder den Zustandsprüfungs-Port für eine Zielgruppe, die vom Load Balancer zum Weiterleiten von Anforderungen verwendet wird, aktualisieren, müssen Sie überprüfen, ob die Sicherheitsgruppen für den Load Balancer den Datenverkehr auf dem neuen Port in beide Richtungen zulassen. Falls nicht, können Sie die Regeln für die derzeit zugeordneten Sicherheitsgruppen ändern oder dem Load Balancer andere Sicherheitsgruppen zuordnen. Sie können die Ports und Protokolle auswählen, die zugelassen werden sollen. Sie können beispielsweise ICMP-Verbindungen (Internet Control Message Protocol) für den Load Balancer öffnen, um auf Ping-Anforderungen zu antworten (Ping-Anforderungen werden jedoch nicht an alle Instances übermittelt).

Empfohlene Regeln

Die folgenden Regeln werden für einen Load Balancer empfohlen, der mit dem Internet verbunden ist.

Inbound

Source	Port Range	Comment
0.0.0.0/0	<i>Listener</i>	Allen eingehenden Datenverkehr auf dem Load Balancer Listener-Port erlauben
Outbound		
Destination	Port Range	Comment
<i>Instance-Sicherheitsgruppe</i>	<i>Instance-Listener</i>	Ausgehenden Datenverkehr an Instances auf dem Instance-Listener-Port erlauben
<i>Instance-Sicherheitsgruppe</i>	<i>Zustandsprüfung</i>	Ausgehenden Datenverkehr an Instances auf dem Zustandsprüfungsport erlauben

Die folgenden Regeln werden für einen internen Load Balancer empfohlen.

Inbound		
Source	Port Range	Comment
<i>VPC-CIDR</i>	<i>Listener</i>	Eingehenden Datenverkehr aus dem VPC CIDR auf dem Load Balancer-Listener-Port erlauben
Outbound		
Destination	Port Range	Comment
<i>Instance-Sicherheitsgruppe</i>	<i>Instance-Listener</i>	Ausgehenden Datenverkehr an Instances auf dem

Instance-Listener-Port
erlauben

Ausgehenden Datenverkehr an Instances auf dem Zustandsprüfungsport erlauben

Instance-Sicherheitsgruppe *Zustandsprüfung*

Die folgenden Regeln werden für einen Application Load Balancer empfohlen, der als Ziel für einen Network Load Balancer verwendet wird.

Inbound

Source	Port Range	Comment
<i>Client-IP-Adressen/CIDR</i>	<i>alb Listener</i>	Eingehenden Client-Datenverkehr am Load-Balancer-Listener-Port erlauben.
<i>VPC-CIDR</i>	<i>alb Listener</i>	Lassen Sie eingehenden Client-Verkehr über AWS PrivateLink den Load Balancer-Listener-Port zu
<i>VPC-CIDR</i>	<i>alb Listener</i>	Eingehenden Zustandsdatenverkehr vom Network Load Balancer erlauben

Outbound

Destination	Port Range	Comment
<i>Instance-Sicherheitsgruppe</i>	<i>Instance-Listener</i>	Ausgehenden Datenverkehr an Instances auf dem Instance-Listener-Port erlauben
<i>Instance-Sicherheitsgruppe</i>	<i>Zustandsprüfung</i>	Ausgehenden Datenverkehr an Instances auf dem

Zustandsprüfungsport erlauben

Beachten Sie, dass die Sicherheitsgruppen für Ihren Application Load Balancer die Verbindungsverfolgung verwenden, um Informationen über den vom Network Load Balancer kommenden Datenverkehr zu verfolgen. Das geschieht unabhängig von den Sicherheitsgruppenregeln, die für Ihren Application Load Balancer festgelegt wurden. Weitere Informationen zur Amazon EC2 EC2-Verbindungsverfolgung finden Sie unter [Verbindungsverfolgung für Sicherheitsgruppen](#) im Amazon EC2 EC2-Benutzerhandbuch.

Um sicherzustellen, dass Ihre Ziele ausschließlich Traffic vom Load Balancer erhalten, beschränken Sie die mit Ihren Zielen verknüpften Sicherheitsgruppen so, dass sie ausschließlich Traffic vom Load Balancer akzeptieren. Dies kann erreicht werden, indem Sie die Sicherheitsgruppe des Load Balancers als Quelle in der Eingangsregel der Sicherheitsgruppe des Ziels festlegen.

Außerdem sollten Sie eingehenden ICMP-Datenverkehr zur Unterstützung von Path MTU Discovery erlauben. Weitere Informationen finden Sie unter [Path MTU Discovery](#) im Amazon EC2 EC2-Benutzerhandbuch.

Aktualisieren der zugeordneten Sicherheitsgruppen

Sie können die dem Load Balancer zugeordneten Sicherheitsgruppen jederzeit ändern.

So aktualisieren Sie Sicherheitsgruppen mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Wählen Sie auf der Registerkarte Sicherheit die Option Bearbeiten aus.
5. Um eine Sicherheitsgruppe mit Ihrem Load Balancer zu verknüpfen, wählen Sie sie aus. Um eine Sicherheitsgruppenverknüpfung zu entfernen, wählen Sie das X-Symbol für die Sicherheitsgruppe.
6. Wählen Sie Änderungen speichern aus.

Um Sicherheitsgruppen mit dem zu aktualisieren AWS CLI

Verwenden Sie den Befehl [set-security-groups](#).

IP-Adresstypen für Ihren Application Load Balancer

Sie können Ihren Application Load Balancer so konfigurieren, dass Clients mit dem Load Balancer nur über IPv4-Adressen oder sowohl über IPv4- als auch über IPv6-Adressen (Dualstack) kommunizieren können. Der Load Balancer kommuniziert mit Zielen auf der Grundlage des IP-Adresstyps der Zielgruppe. Weitere Informationen finden Sie unter [IP-Adresstyp](#).

Dualstack-Anforderungen

- Sie können den IP-Adresstyp bei der Erstellung des Load Balancers festlegen und jederzeit aktualisieren.
- Die Virtual Private Cloud (VPC) und die Subnetze, die Sie für den Load Balancer angeben, müssen IPv6-CIDR-Blöcken zugeordnet sein. Weitere Informationen finden Sie unter [IPv6-Adressen](#) im Amazon-EC2-Benutzerhandbuch.
- Die Routingtabellen für die Load Balancer-Subnetze müssen IPv6-Datenverkehr weiterleiten.
- Die Sicherheitsgruppen für den Load Balancer müssen IPv6-Datenverkehr zulassen.
- Die Netzwerk-ACLs für die Load Balancer-Subnetze müssen IPv6-Datenverkehr zulassen.

So legen Sie den IP-Adresstyp bei der Erstellung fest

Konfigurieren Sie die Einstellungen wie unter [???](#) beschrieben.

So aktualisieren Sie den IP-Adresstyp mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Wählen Sie auf der Registerkarte Netzwerkzuordnung die Option IP-Adresstyp bearbeiten aus.
5. Wählen Sie als IP-Adresstyp IPv4, um nur IPv4-Adressen zu unterstützen, Dualstack, um sowohl IPv4- als auch IPv6-Adressen zu unterstützen, oder Dualstack ohne öffentliches IPv4, um nur IPv6-Adressen zu unterstützen.
6. Wählen Sie Änderungen speichern aus.

Um den IP-Adresstyp mit dem zu aktualisieren AWS CLI

Verwenden Sie den Befehl [set-ip-address-type](#).

Tags für Ihren Application Load Balancer

Tags helfen Ihnen, Ihre Load Balancer auf unterschiedliche Weise zu kategorisieren, z. B. nach Zweck, Eigentümer oder Umgebung.

Sie können mehrere Tags für jeden Load Balancer hinzufügen. Wenn Sie ein Tag mit einem Schlüssel hinzufügen, der dem Load Balancer bereits zugeordnet ist, ändert sich der Wert dieses Tags.

Wenn Sie ein Tag nicht mehr benötigen, können Sie es von Ihrem Load Balancer entfernen.

Einschränkungen

- Maximale Anzahl von Tags pro Ressource: 50
- Maximale Schlüssellänge: 127 Unicode-Zeichen
- Maximale Wertlänge: 255 Unicode-Zeichen
- Bei Tag-Schlüsseln und -Werten muss die Groß- und Kleinschreibung beachtet werden. Erlaubte Zeichen sind Buchstaben, Leerzeichen und Zahlen, die in UTF-8 darstellbar sind, sowie die folgenden Sonderzeichen: + - = _ : / @. Verwenden Sie keine führenden oder nachgestellten Leerzeichen.
- Verwenden Sie das `aws :` Präfix nicht in Ihren Tagnamen oder -Werten, da es für die AWS Verwendung reserviert ist. Sie können keine Tag-Namen oder Werte mit diesem Präfix bearbeiten oder löschen. Tags mit diesem Präfix werden nicht als Ihre Tags pro Ressourcenlimit angerechnet.

So aktualisieren Sie die Tags für einen Load Balancer mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Wählen Sie auf der Registerkarte Tags die Option Tags verwalten aus und führen Sie einen oder mehrere der folgenden Schritte aus:
 - a. Um ein Tag zu aktualisieren, bearbeiten Sie die Werte von Schlüssel und Wert.
 - b. Um ein neues Tag hinzuzufügen, wählen Sie Tag hinzufügen aus und geben Sie Werte für Schlüssel und Wert ein.
 - c. Um ein Tag zu löschen, wählen Sie Entfernen neben dem zu löschenden Tag aus.

5. Wenn Sie die Aktualisierung der Tags abgeschlossen haben, wählen Sie Änderungen speichern.

Um die Tags für einen Load Balancer mit dem zu aktualisieren AWS CLI

Verwenden Sie die Befehle [add-tags](#) und [remove-tags](#).

Löschen eines Application Load Balancers

Sobald der Load Balancer verfügbar ist, wird Ihnen jede ganze oder angebrochene Stunde in Rechnung gestellt, in der er ausgeführt wird. Wenn Sie den Load Balancer nicht mehr benötigen, können Sie ihn löschen. Sobald der Load Balancer gelöscht wurde, fallen keine weiteren Kosten dafür mehr an.

Sie können einen Load Balancer nicht löschen, solange der Löschschutz aktiviert ist. Weitere Informationen finden Sie unter [Löschschutz](#).

Beachten Sie, dass das Löschen eines Load Balancers sich nicht auf seine registrierten Ziele auswirkt. Ihre EC2-Instances werden beispielsweise weiter ausgeführt und sind weiterhin bei ihren Zielgruppen registriert. Informationen zum Löschen Ihrer Zielgruppen finden Sie unter [Löschen einer Zielgruppe](#).

So löschen Sie einen Load Balancer mithilfe der Konsole

1. Wenn Sie einen DNS-Eintrag für Ihre Domain haben, der auf Ihren Load Balancer verweist, verweisen Sie ihn an den neuen Standort und warten Sie, bis die DNS-Änderungen wirksam werden, bevor Sie den Load Balancer löschen.

Beispiel:

- Wenn es sich bei dem Datensatz um einen CNAME-Eintrag mit einer Time To Live (TTL) von 300 Sekunden handelt, warten Sie mindestens 300 Sekunden, bevor Sie mit dem nächsten Schritt fortfahren.
 - Wenn es sich bei dem Datensatz um einen Route 53-Alias(A)-Eintrag handelt, warten Sie mindestens 60 Sekunden.
 - Wenn Sie Route 53 verwenden, dauert es 60 Sekunden, bis die Datensatzänderung an alle globalen Route 53-Nameserver weitergegeben wird. Fügen Sie diese Zeit zum TTL-Wert des Datensatzes hinzu, der aktualisiert wird.
2. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.

3. Wählen Sie im Navigationsbereich Load Balancers aus.
4. Wählen Sie den Load Balancer aus und klicken Sie dann auf Aktionen und auf Load Balancer löschen.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **confirm** ein und wählen Sie dann Löschen aus.

Um einen Load Balancer zu löschen, verwenden Sie AWS CLI

Verwenden Sie den Befehl [delete-load-balancer](#).

Zonenverschiebung

Die Zonenverschiebung ist eine Funktion in Amazon Route 53 Application Recovery Controller (Route 53 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. Route 53 ARC erstellt 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 die Zonenverschiebung: Zustandsprüfungen und zonale IP-Adressen](#) im Entwicklerhandbuch für Amazon Route 53 Application Recovery Controller.

Zonenverschiebungen werden nur auf Application Load Balancern und Network Load Balancern unterstützt, wenn das zonenübergreifende Load Balancing deaktiviert ist. Wenn Sie das zonenübergreifende Load Balancing aktivieren, können Sie keine Zonenverschiebungen starten. Weitere Informationen finden Sie unter [Für Zonenverschiebung unterstützte Ressourcen](#) im Entwicklerhandbuch für Amazon Route 53 Application Recovery Controller.

Bevor Sie die Zonenverschiebung verwenden, sollten Sie Folgendes beachten:

- Zonenübergreifendes Load Balancing wird bei Zonenverschiebungen nicht unterstützt. Sie müssen das zonenübergreifende Load Balancing deaktivieren, um diese Funktion nutzen zu können.
- Die Zonenverschiebung wird nicht unterstützt, wenn Sie einen Application Load Balancer als Accelerator-Endpunkt in AWS Global Accelerator 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 zonale Load-Balancer-IP-Adressen aus DNS, wenn sich mehrere Infrastrukturprobleme auf Services 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.
- Wenn ein Application Load Balancer das Ziel eines Network Load Balancers ist, starten Sie die Zonenverschiebung immer vom Network Load Balancer aus. Wenn Sie eine Zonenverschiebung vom Application Load Balancer aus starten, erkennt der Network Load Balancer die Verschiebung nicht und sendet weiterhin Datenverkehr an den Application Load Balancer.

Weitere Hinweise und Informationen finden Sie unter [Bewährte Methoden für Route-53-ARC-Zonenverschiebungen](#) im Entwicklerhandbuch für Amazon Route 53 Application Recovery Controller.

Starten einer Zonenverschiebung

Die Schritte in diesem Verfahren erläutern, wie Sie eine Zonenverschiebung mithilfe der Amazon-EC2-Konsole starten. Schritte zum Starten einer Zonenverschiebung mithilfe der Route-53-ARC-Konsole finden Sie unter [Starten einer Zonenverschiebung](#) im Entwicklerhandbuch für Amazon Route 53 Application Recovery Controller.

So starten Sie eine Zonenverschiebung mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-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 Namen eines Load Balancers aus.
4. Wählen Sie auf der Registerkarte Integrationen unter Route 53 Application Recovery Controller die Option Zonenverschiebung starten aus.
5. Wählen Sie die Availability Zone, von der Sie den Datenverkehr wegleiten möchten.
6. Wählen Sie ein Ablaufdatum für die Zonenverschiebung aus oder geben Sie es ein. Eine Zonenverschiebung kann zunächst auf eine Dauer von 1 Minute bis zu 3 Tagen (72 Stunden) festgelegt werden.

- Alle Zonenverschiebungen sind temporär. Sie müssen ein Ablaufdatum festlegen, aber Sie können aktive Verschiebungen später aktualisieren, um ein neues Ablaufdatum festzulegen.
7. Geben Sie einen Kommentar ein. Sie können die Zonenverschiebung später aktualisieren, um den Kommentar zu bearbeiten, wenn Sie möchten.
 8. Aktivieren Sie das Kontrollkästchen, um zu bestätigen, dass der Start einer Zonenverschiebung die Kapazität Ihrer Anwendung reduziert, da der Datenverkehr von der Availability Zone weg verlagert wird.
 9. Wählen Sie Starten.

So starten Sie eine Zonenverschiebung mithilfe der AWS CLI

Informationen zum programmgesteuerten Arbeiten mit Zonenverschiebungen finden Sie im [Zonenverschiebungs-API-Referenzhandbuch](#).

Aktualisieren einer Zonenverschiebung

Die Schritte in diesem Verfahren erläutern, wie Sie eine Zonenverschiebung mithilfe der Amazon-EC2-Konsole aktualisieren. Schritte zum Aktualisieren einer Zonenverschiebung mithilfe der Konsole von Amazon Route 53 Application Recovery Controller finden Sie unter [Aktualisieren einer Zonenverschiebung](#) im Entwicklerhandbuch für Amazon Route 53 Application Recovery Controller.

So aktualisieren Sie eine Zonenverschiebung mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-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 Namen eines Load Balancers mit einer aktiven Zonenverschiebung aus.
4. Wählen Sie auf der Registerkarte Integrationen unter Route 53 Application Recovery Controller die Option Zonenverschiebung aktualisieren aus.

Dadurch wird die Route-53-ARC-Konsole geöffnet, um die Aktualisierung fortzusetzen.

5. Wählen Sie für Ablauf der Zonenverschiebung festlegen optional ein Ablaufdatum aus oder geben Sie es ein.
6. Bearbeiten Sie unter Kommentar optional den vorhandenen Kommentar oder geben Sie einen neuen Kommentar ein.
7. Wählen Sie Aktualisieren aus.

So aktualisieren Sie eine Zonenverschiebung mithilfe der AWS CLI

Informationen zum programmgesteuerten Arbeiten mit Zonenverschiebungen finden Sie im [Zonenverschiebungs-API-Referenzhandbuch](#).

Abbrechen einer Zonenverschiebung

Die Schritte in diesem Verfahren erläutern, wie Sie eine Zonenverschiebung mithilfe der Amazon-EC2-Konsole abbrechen. Schritte zum Abbrechen einer Zonenverschiebung mithilfe der Konsole von Amazon Route 53 Application Recovery Controller finden Sie unter [Abbrechen einer Zonenverschiebung](#) im Entwicklerhandbuch für Amazon Route 53 Application Recovery Controller.

So brechen Sie eine Zonenverschiebung mithilfe der Konsole ab

1. Öffnen Sie die Amazon EC2-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 Namen eines Load Balancers mit einer aktiven Zonenverschiebung aus.
4. Wählen Sie auf der Registerkarte Integrationen unter Route 53 Application Recovery Controller die Option Zonenverschiebung abbrechen aus.

Dadurch wird die Route-53-ARC-Konsole geöffnet, um den Abbruch fortzusetzen.

5. Wählen Sie Zonenverschiebung abbrechen.
6. Wählen Sie im Bestätigungsdiaologfeld Bestätigen.

So brechen Sie eine Zonenverschiebung mithilfe der AWS CLI ab

Informationen zum programmgesteuerten Arbeiten mit Zonenverschiebungen finden Sie im [Zonenverschiebungs-API-Referenzhandbuch](#).

Listener für Ihre Application Load Balancer

Ein Listener ist ein Prozess, der mit dem Protokoll und dem Port, das bzw. den Sie konfigurieren, Verbindungsanforderungen prüft. Bevor Sie Ihren Application Load Balancer verwenden können, müssen Sie mindestens einen Listener hinzufügen. Wenn Ihr Load Balancer keine Listener hat, kann er keinen Datenverkehr von Clients empfangen. Die Regeln, die Sie für Listener definieren, bestimmen, wie der Load Balancer Anforderungen an die Ziele weiterleitet, die Sie registrieren, z. B. EC2-Instances.

Inhalt

- [Listener-Konfiguration](#)
- [Listener-Regeln](#)
- [Regelaktionstypen](#)
- [Regelbedingungstypen](#)
- [Erstellen eines HTTP-Listeners für Ihren Application Load Balancer](#)
- [Erstellen eines HTTPS-Listeners für Ihren Application Load Balancer](#)
- [Listener-Regeln für Ihren Application Load Balancer](#)
- [Aktualisieren eines HTTPS-Listeners für Ihren Application Load Balancer](#)
- [Gegenseitige Authentifizierung mit TLS im Application Load Balancer](#)
- [Authentifizieren von Benutzern mithilfe eines Application Load Balancers](#)
- [HTTP-Header und Application Load Balancer](#)
- [Tags für Ihre Listener und Regeln](#)
- [Löschen eines Listeners für Ihren Application Load Balancer](#)

Listener-Konfiguration

Listener unterstützen die folgenden Protokolle und Ports:

- Protocols (Protokolle): HTTP, HTTPS
- Ports: 1-65535

Sie können einen HTTPS-Listener verwenden, um die Ver- und Entschlüsselung auf Ihren Load Balancer auszulagern, damit sich Ihre Anwendungen auf die Geschäftslogik konzentrieren können.

Wenn das Listener-Protokoll HTTPS ist, müssen Sie auf dem Listener mindestens ein SSL-Serverzertifikat bereitstellen. Weitere Informationen finden Sie unter [Erstellen eines HTTPS-Listeners für Ihren Application Load Balancer](#).

Wenn Sie sicherstellen müssen, dass die Ziele den HTTPS-Verkehr anstelle des Load Balancers entschlüsseln, können Sie einen Network Load Balancer mit einem TCP-Listener an Port 443 erstellen. Bei einem TCP-Listener leitet der Load Balancer verschlüsselten Datenverkehr an die Ziele weiter, ohne ihn zu entschlüsseln. Weitere Informationen finden Sie im [Benutzerhandbuch für Network Load Balancers](#).

Application Load Balancers bieten native Unterstützung für WebSockets. Sie können eine bestehende HTTP/1.1-Verbindung in eine WebSocket (wsoderwss) -Verbindung umwandeln, indem Sie ein HTTP-Verbindungs-Upgrade verwenden. Wenn Sie ein Upgrade durchführen, wird die für Anfragen (sowohl zum Load Balancer als auch zum Ziel) verwendete TCP-Verbindung über den Load Balancer zu einer dauerhaften WebSocket Verbindung zwischen dem Client und dem Ziel. Sie können sie sowohl WebSockets mit HTTP- als auch mit HTTPS-Listnern verwenden. Die Optionen, die Sie für Ihren Listener auswählen, gelten sowohl für WebSocket Verbindungen als auch für HTTP-Verkehr. Weitere Informationen finden Sie unter [So funktioniert das WebSocket Protokoll](#) im Amazon CloudFront Developer Guide.

Application Load Balancer verfügen über native Unterstützung für HTTP/2 mit HTTPS-Listener. Sie können mit einer einzigen HTTP/2-Verbindung bis zu 128 Anforderungen parallel senden. Sie können die Protokollversion verwenden, um die Anforderung mit HTTP/2 an die Ziele zu senden. Weitere Informationen finden Sie unter [Protokollversion](#). Da HTTP/2 Frontend-Verbindungen effizienter verwendet, gibt es möglicherweise weniger Verbindungen zwischen Clients und dem Load Balancer. Sie können das Server-Push-Feature von HTTP/2 nicht verwenden.

Weitere Informationen finden Sie unter [Weiterleitung von Anforderungen](#) im Benutzerhandbuch zu Elastic Load Balancing.


Listener-Regeln

Jeder Listener hat eine Standardaktion, auch als Standardregel bezeichnet. Die Standardregel kann nicht gelöscht werden und wird immer zuletzt ausgeführt. Es können zusätzliche Regeln erstellt werden, die aus einer Priorität, mindestens einer Aktion und mindestens einer Bedingung bestehen. Sie können jederzeit Regel hinzufügen oder bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer Regel](#).

Standardregeln

Beim Erstellen eines Listeners definieren Sie Aktionen für die Standardregel. Standardregeln können keine Bedingungen aufweisen. Wenn für die Regeln eines Listeners keine Bedingungen erfüllt werden, wird die Aktion für die Standardregel durchgeführt.

Es folgt ein Beispiel für eine Standardregel, wie in der Konsole dargestellt:

Priority	Conditions (If)	Actions (Then) 
Last (default)	<i>If no other rule applies</i>	Forward to target group <ul style="list-style-type: none">• my-targets: 1 (100%)• Group-level stickiness: Off

Priorität der Regel

Jede Regel hat eine Priorität. Regeln werden in der Reihenfolge ihrer Prioritäten bewertet, ausgehend vom niedrigsten Wert hin zum höchsten Wert. Die Standardregel wird zuletzt ausgewertet. Sie können die Priorität einer nicht standardmäßigen Regel jederzeit ändern. Sie können die Priorität der Standardregel nicht ändern. Weitere Informationen finden Sie unter [Aktualisieren der Regelpriorität](#).

Regelaktionen

Jede Regelaktion verfügt über einen Typ, eine Priorität und die für die Durchführung der Aktion erforderlichen Informationen. Weitere Informationen finden Sie unter [Regelaktionstypen](#).

Regelbedingungen

Jede Regelbedingung weist einen Typ und Bedingungsinformationen auf. Wenn die Bedingungen für eine Regel erfüllt sind, wird die dazugehörige Aktion durchgeführt. Weitere Informationen finden Sie unter [Regelbedingungstypen](#).

Regelaktionstypen

Im Folgenden finden Sie die unterstützten Aktionstypen einer Listener-Regel:

authenticate-cognito

[HTTPS-Listener] Verwenden von Amazon Cognito zum Authentifizieren von Benutzern. Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mithilfe eines Application Load Balancers](#).

authenticate-oidc

[HTTPS-Listener] Verwenden eines Identitätsanbieters, der mit OpenID Connect (OIDC) konform ist, um Benutzer zu authentifizieren

fixed-response

Zurückgeben einer benutzerdefinierten HTTP-Antwort Weitere Informationen finden Sie unter [Aktionen mit feststehender Antwort](#).

forward

Weiterleiten von Anforderungen an die angegebenen Zielgruppen. Weitere Informationen finden Sie unter [Weiterleitungsaktionen](#).

redirect

Weiterleiten von Anforderungen von einer URL an eine andere Weitere Informationen finden Sie unter [Weiterleitungsaktionen](#).

Die Aktion mit der niedrigsten Priorität wird zuerst durchgeführt. Jede Regel muss genau eine der folgenden Aktionen enthalten: `forward`, `redirect` oder `fixed-response`. Außerdem muss es die letzte auszuführende Regel sein.

Wenn die Protokollversion gRPC oder HTTP/2 ist, sind `forward`-Aktionen die einzigen unterstützten Aktionen.

Aktionen mit feststehender Antwort

Verwenden Sie `fixed-response`-Aktionen, um Client-Anforderungen zu verwerfen und eine benutzerdefinierte HTTP-Antwort zurückzugeben. Sie können mit dieser Aktion einen 2XX-, 4XX- oder 5XX-Antwortcode und optional eine Nachricht zurückgeben.

Wenn eine `fixed-response`-Aktion ausgeführt wird, werden die Aktion und die URL des Weiterleitungsziels in den Zugriffsprotokollen aufgezeichnet. Weitere Informationen finden Sie unter [Zugriffsprotokolleinträge](#). Die Anzahl der erfolgreichen `fixed-response`-Aktionen wird in der Metrik

HTTP_Fixed_Response_Count erfasst. Weitere Informationen finden Sie unter [Application-Load-Balancer-Metriken](#).

Example Beispiel für eine Aktion mit einer festen Antwort für AWS CLI

Sie können eine Aktion angeben, wenn Sie eine Regel erstellen oder ändern. Weitere Informationen finden Sie bei den Befehlen [create-rule](#) und [modify-rule](#). Mit der folgenden Aktion wird mit dem angegebenen Statuscode und dem Textkörper eine festgelegte Antwort gesendet.

```
[
  {
    "Type": "fixed-response",
    "FixedResponseConfig": {
      "StatusCode": "200",
      "ContentType": "text/plain",
      "MessageBody": "Hello world"
    }
  }
]
```

Weiterleitungsaktionen

Sie können mithilfe von `forward`-Aktionen Anforderungen an eine oder mehrere Zielgruppen weiterleiten. Wenn Sie mehrere Zielgruppen für eine `forward`-Aktion angeben, müssen Sie für jede Zielgruppe eine Gewichtung angeben. Jede Zielgruppengewichtung ist ein Wert zwischen 0 und 999. Anforderungen, die einer Listener-Regel mit gewichteten Zielgruppen entsprechen, werden basierend auf ihren Gewichtungen an diese Zielgruppen verteilt. Wenn Sie beispielsweise zwei Zielgruppen mit einer Gewichtung von 10 angeben, erhält jede Zielgruppe die Hälfte der Anforderungen. Wenn Sie zwei Zielgruppen angeben, eine mit einer Gewichtung von 10 und die andere mit einer Gewichtung von 20, erhält die Zielgruppe mit der Gewichtung von 20 doppelt so viele Anforderungen wie die andere Zielgruppe.

Standardmäßig garantiert das Konfigurieren einer Regel für die Verteilung des Datenverkehrs zwischen gewichteten Zielgruppen nicht, dass Sticky Sessions eingehalten werden. Um sicherzustellen, dass Sticky Sessions eingehalten werden, aktivieren Sie die Klebrigkeit der Zielgruppe für die Regel. Wenn der Load Balancer eine Anfrage zum ersten Mal an eine gewichtete Zielgruppe weiterleitet, generiert er ein Cookie mit dem Namen `AWSALBTG`, das Informationen über die ausgewählte Zielgruppe kodiert, das Cookie verschlüsselt und das Cookie in die Antwort an den Client einbezieht. Der Client sollte das erhaltene Cookie in nachfolgende Anfragen an den Load

Balancer aufnehmen. Wenn der Load Balancer eine Anforderung empfängt, die mit einer Regel mit aktivierter Zielgruppenklebrigkeit übereinstimmt und das Cookie enthält, wird die Anforderung an die im Cookie angegebene Zielgruppe weitergeleitet.

Application Load Balancer unterstützen keine Cookie-Werte, die URL-codiert sind.

Bei CORS (Cross-Origin Resource Sharing)-Anforderungen benötigen einige Browser `SameSite=None; Secure` zum Aktivieren von Stickiness. In diesem Fall generiert Elastic Load Balancing ein zweites Cookie `AWSALBTGCORS`, das dieselben Informationen wie das ursprüngliche Stickiness-Cookie plus dieses `SameSite` Attribut enthält. Kunden erhalten beide Cookies.

Example Beispiel einer Weiterleitungsaktion mit einer Zielgruppe

Sie können eine Aktion angeben, wenn Sie eine Regel erstellen oder ändern. Weitere Informationen finden Sie bei den Befehlen [create-rule](#) und [modify-rule](#). Die folgende Aktion leitet die Anforderungen an die angegebene Zielgruppe weiter.

```
[
  {
    "Type": "forward",
    "ForwardConfig": {
      "TargetGroups": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067"
        }
      ]
    }
  }
]
```

Example Beispiel einer Weiterleitungsaktion mit zwei gewichteten Zielgruppen

Die folgende Aktion leitet Anforderungen an die beiden angegebenen Zielgruppen basierend auf der Gewichtung jeder Zielgruppe weiter.

```
[
  {
    "Type": "forward",
    "ForwardConfig": {
      "TargetGroups": [
```

```

    {
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/blue-targets/73e2d6bc24d8a067",
      "Weight": 10
    },
    {
      "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/green-targets/09966783158cda59",
      "Weight": 20
    }
  ]
}
]

```

Example Beispiel einer Weiterleitungsaktion mit aktivierter Stickiness

Wenn Sie über eine Weiterleitungsaktion mit mehreren Zielgruppen verfügen und für eine oder mehrere Zielgruppen [Sticky Sessions](#) aktiviert sind, müssen Sie die Stickiness der Zielgruppe aktivieren.

Die folgende Aktion leitet Anforderungen an die beiden angegebenen Zielgruppen weiter, wobei die Klebrigkeit der Zielgruppe aktiviert ist. Anforderungen, die die Stickiness-Cookies nicht enthalten, werden basierend auf der Gewichtung jeder Zielgruppe weitergeleitet.

```

[
  {
    "Type": "forward",
    "ForwardConfig": {
      "TargetGroups": [
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/blue-targets/73e2d6bc24d8a067",
          "Weight": 10
        },
        {
          "TargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/green-targets/09966783158cda59",
          "Weight": 20
        }
      ],
      "TargetGroupStickinessConfig": {
        "Enabled": true,

```

```
    "DurationSeconds": 1000
  }
}
]
```

Weiterleitungsaktionen

Mithilfe von `redirect`-Aktionen können Sie Client-Anforderungen von einer URL an eine andere weiterleiten. Konfigurieren Sie Weiterleitungen je nach Bedarf entweder als temporär (HTTP 302) oder permanent (HTTP 301).

eine URI umfasst folgende Komponenten:

```
protocol://hostname:port/path?query
```

Sie müssen mindestens eine der folgenden Komponenten modifizieren, um eine Weiterleitungsschleife zu verhindern: Protokoll, Hostname, Port oder Pfad. Für alle Komponenten, an denen Sie keine Änderungen vornehmen, wird der ursprüngliche Wert beibehalten.

Protokoll

Das Protokoll (HTTP oder HTTPS). Sie können von HTTP nach HTTP, von HTTP nach HTTPS und von HTTPS nach HTTPS weiterleiten. Eine Weiterleitung von HTTPS nach HTTP ist nicht möglich.

hostname

Der Hostname Bei einem Hostnamen wird die Groß- und Kleinschreibung nicht beachtet. Er kann bis zu 128 Zeichen lang sein und aus alphanumerischen Zeichen, Platzhaltern (* und ?) und Bindestrichen (-) bestehen.

port

Der Port (1 bis 65535)

Pfad

Der absolute Pfad, beginnend mit dem vorangestellten "/" Bei einem Pfad muss die Groß- und Kleinschreibung nicht beachtet werden. Er kann 128 Zeichen lang sein und aus alphanumerischen Zeichen, Platzhaltern (* und ?), & (mittels &) sowie die Sonderzeichen `$_~"@"` bestehen.

query

Die Abfrageparameter Die maximale Länge beträgt 128 Zeichen.

Sie können URI-Komponenten der ursprünglichen URL in der Ziel-URL weiter nutzen. Verwenden Sie dazu die folgenden reservierten Schlüsselwörter:

- `{protocol}` – zur Beibehaltung des Protokolls. In den Protokoll- und Abfragekomponenten zu verwenden.
- `{host}` – Zur Beibehaltung der Domain. In den Hostnamen-, Pfad- und Abfragekomponenten zu verwenden.
- `{port}` – Zur Beibehaltung des Ports. In den Port-, Pfad- und Abfragekomponenten zu verwenden.
- `{path}` – Zur Beibehaltung des Pfads. In den Pfad- und Abfragekomponenten zu verwenden.
- `{query}` – Zur Beibehaltung der Abfrageparameter. In der Abfragekomponente zu verwenden.

Wenn eine `redirect`-Aktion ausgeführt wird, wird diese in den Zugriffsprotokollen aufgezeichnet. Weitere Informationen finden Sie unter [Zugriffsprotokolleinträge](#). Die Anzahl der erfolgreichen `redirect`-Aktionen wird in der Metrik `HTTP_Redirect_Count` erfasst. Weitere Informationen finden Sie unter [Application-Load-Balancer-Metriken](#).

Example Beispiel für Weiterleitungsaktionen mithilfe der Konsole

Mit der folgenden Regel wird beispielsweise eine permanente Weiterleitung auf eine URL mit dem HTTPS-Protokoll und dem festgelegten Port 40443 eingerichtet. Beibehalten werden der ursprüngliche Hostname, der Pfad und die Abfrageparameter. Dieser Bildschirm entspricht "`https://{host}:40443/{path}?{query}`".

Action types

Forward to target groups

Redirect to URL

Return fixed response

Redirect to URL [Info](#)

Redirect client requests from one URL to another. You cannot redirect HTTPS to HTTP. To avoid a redirect loop, you must modify at least one of the following components: protocol, port, hostname or path. Components that you do not modify retain their original values.

URI parts | Full URL

Protocol : Port
To retain the original port enter #{port}.

HTTPS ▼ 40443
1-65535

Custom host, path, query
Select to modify host, path and query. If no changes are made, settings from the request URL are retained.

Status code

301 - Permanently moved ▼

Mit der folgenden Regel wird eine permanente Weiterleitung auf eine URL mit dem ursprünglichen Protokoll, Port, Hostnamen und Abfrageparametern eingerichtet. Der Pfad wird mit dem Schlüsselwort `#{path}` modifiziert. Dieser Bildschirm entspricht `"#{protocol}://#{host}:#{port}/new/#{path}?#{query}"`.

Action types Forward to target groups Redirect to URL Return fixed response**Redirect to URL** | [Info](#)

Redirect client requests from one URL to another. You cannot redirect HTTPS to HTTP. To avoid a redirect loop, you must modify at least one of the following components: protocol, port, hostname or path. Components that you do not modify retain their original values.

URI parts**Full URL****Protocol : Port**

To retain the original port enter #{port}.

#{protocol} ▼

#{port}

1-65535

 Custom host, path, query

Select to modify host, path and query. If no changes are made, settings from the request URL are retained.

Host

Specify a host or retain the original host by using #{host}. Not case sensitive.

#{host}

Maximum 128 characters. Allowed characters are a-z, A-Z, 0-9; the following special characters: -,; and wildcards (* and ?). At least one "." is required. Only alphabetical characters are allowed after the final "." character.

Path

Specify a path or retain the original path by using #{path}. Case sensitive.

/new/#{path}

Maximum 128 characters. Allowed characters are a-z, A-Z, 0-9; the following special characters: _-.\$/~'"@:~; & (using &); and wildcards (* and ?).

Query - optional

Specify a query or retain the original query by using #{query}. Not case sensitive.

#{query}

Maximum 128 characters.

Status code

301 - Permanently moved ▼

Example Beispiel für eine Umleitungsaktion für AWS CLI

Sie können eine Aktion angeben, wenn Sie eine Regel erstellen oder ändern. Weitere Informationen finden Sie bei den Befehlen [create-rule](#) und [modify-rule](#). Mit der folgenden Aktion wird eine HTTP-Anforderung an eine HTTPS-Anforderung an Port 443 weitergeleitet, die denselben Hostnamen, Pfad und die gleiche Abfragezeichenfolge wie die HTTP-Anforderung aufweist.

```
[
  {
    "Type": "redirect",
    "RedirectConfig": {
      "Protocol": "HTTPS",
      "Port": "443",
      "Host": "#{host}",
      "Path": "/#{path}",
      "Query": "#{query}",
      "StatusCode": "HTTP_301"
    }
  }
]
```

Regelbedingungstypen

Im Folgenden finden Sie die unterstützten Bedingungstypen für eine Regel:

host-header

Die Route basiert auf dem Hostnamen jeder Anforderung. Weitere Informationen finden Sie unter [Hostbedingungen](#).

http-header

Die Route basiert auf den HTTP-Headern für jede Anforderung. Weitere Informationen finden Sie unter [HTTP-Header-Bedingungen](#).

http-request-method

Die Route basiert auf der HTTP-Anforderungsmethode jeder Anforderung. Weitere Informationen finden Sie unter [Bedingungen für HTTP-Anforderungsmethoden](#).

path-pattern

Die Route basiert auf den Pfadmustern in den Anforderungs-URLs. Weitere Informationen finden Sie unter [Pfadbedingungen](#).

query-string

Die Route basiert auf Schlüssel/Wert-Paaren oder Werten in den Abfragezeichenfolgen. Weitere Informationen finden Sie unter [Abfragezeichenfolgebedingungen](#).

source-ip

Die Route basiert auf der Quell-IP-Adresse jeder Anforderung. Weitere Informationen finden Sie unter [Bedingungen für die Quell-IP-Adresse](#).

Jede Regel kann optional jeweils eine der folgenden Bedingungen umfassen: `host-header`, `http-request-method`, `path-pattern`, und `source-ip`. Jede Regel kann optional auch eine oder mehrere der folgenden Bedingungen enthalten: `http-header` und `query-string`.

Sie können bis zu drei Übereinstimmungsbewertungen pro Bedingung angeben. Beispiel: Für jede `http-header`-Bedingung können Sie bis zu drei Zeichenfolgen angeben, die mit dem Wert des HTTP-Headers in der Anforderung verglichen werden. Die Bedingung wird erfüllt, wenn eine der Zeichenfolgen dem Wert des HTTP-Headers entspricht. Wenn alle drei Zeichenfolgen eine Übereinstimmung aufweisen sollen, erstellen Sie eine Bedingung pro Übereinstimmungsbewertung.

Sie können bis zu fünf Übereinstimmungsbewertungen pro Regel angeben. Beispiel: Sie können eine Regel mit fünf Bedingungen erstellen, wobei jede Bedingung eine Übereinstimmungsbewertung aufweist.

Sie können Platzhalterzeichen in die Übereinstimmungsbewertung für die Bedingungen `http-header`, `host-header`, `path-pattern` und `query-string` einschließen. Die Anzahl der Platzhalterzeichen pro Bedingung ist auf 5 beschränkt.

Regeln werden nur auf sichtbare ASCII-Zeichen angewendet; Steuerzeichen (0x00 bis 0x1f und 0x7f) sind ausgeschlossen.

Demos finden Sie unter [Erweiterte Anfrageweiterleitung](#).

HTTP-Header-Bedingungen

Mit HTTP-Header-Bedingungen können Sie Regeln konfigurieren, mit denen Anforderungen auf Grundlage der HTTP-Header für die Anforderung weitergeleitet werden. Sie können die Namen der standardmäßigen oder benutzerdefinierten HTTP-Header-Felder angeben. Beim Headernamen und bei der Übereinstimmungsbewertung wird nicht zwischen die Groß- und Kleinschreibung unterschieden. Die folgenden Platzhalterzeichen werden in den Vergleichszeichenfolgen unterstützt: * (findet eine Übereinstimmung mit 0 oder mehr Zeichen) und ? (findet Übereinstimmungen für genau 1 Zeichen). Platzhalterzeichen werden im Header-Namen nicht unterstützt.

Example Beispiel für eine HTTP-Header-Bedingung für die AWS CLI

Sie können Bedingungen angeben, wenn Sie eine Regel erstellen oder ändern. Weitere Informationen finden Sie bei den Befehlen [create-rule](#) und [modify-rule](#). Die folgende Bedingung wird von Anforderungen mit einem User-Agent-Header erfüllt, der mindestens einer der angegebenen Zeichenfolgen entspricht.

```
[
  {
    "Field": "http-header",
    "HTTPHeaderConfig": {
      "HttpHeaderName": "User-Agent",
      "Values": ["*Chrome*", "*Safari*"]
    }
  }
]
```

Bedingungen für HTTP-Anforderungsmethoden

Mit Bedingungen für HTTP-Anforderungsmethoden können Sie Regeln konfigurieren, mit denen Anforderungen auf Grundlage der HTTP-Anforderungsmethode der Anforderung weitergeleitet werden. Sie können standardmäßige oder benutzerdefinierte HTTP-Methoden angeben. Bei der Übereinstimmungsbewertung wird die Groß- und Kleinschreibung nicht beachtet, Platzhalterzeichen werden nicht unterstützt. Der Methodename muss also eine genaue Übereinstimmung sein.

Wir empfehlen, dass Sie GET- und HEAD-Anforderungen auf die gleiche Weise weiterleiten, da die Antwort auf eine HEAD-Anforderung möglicherweise zwischengespeichert wird.

Example Beispiel für eine HTTP-Methodenbedingung für die AWS CLI

Sie können Bedingungen angeben, wenn Sie eine Regel erstellen oder ändern. Weitere Informationen finden Sie bei den Befehlen [create-rule](#) und [modify-rule](#). Die folgende Bedingung wird von Anforderungen erfüllt, bei der die angegebene Methode verwendet wird.

```
[
  {
    "Field": "http-request-method",
    "HttpRequestMethodConfig": {
      "Values": ["CUSTOM-METHOD"]
    }
  }
]
```

]

Hostbedingungen

Mit Hostbedingungen können Sie Regeln definieren, die Anforderungen basierend auf dem Hostnamen im Host-Header weiterleiten (auch als hostbasierte Weiterleitung bezeichnet). Auf diese Weise können Sie mit einem einzigen Load Balancer mehrere Unterdomains und verschiedene Top-Level-Domains unterstützen.

Beim Hostnamen wird die Groß-/Kleinschreibung nicht berücksichtigt, er kann maximal 128 Zeichen lang sein und kann folgende Zeichen enthalten:

- A-Z, a-z, 0-9
- - .
- * (entspricht 0 oder mehr Zeichen)
- ? (entspricht genau 1 Zeichen)

Sie müssen mindestens ein "."-Zeichen einschließen. Es können nur alphanumerische Zeichen nach dem letzten "."-Zeichen angegeben werden.

Beispiele für Hostnamen

- **example.com**
- **test.example.com**
- ***.example.com**

Die Regel ***.example.com** entspricht **test.example.com**, nicht jedoch **example.com**.

Example Beispiel für eine Host-Header-Bedingung für die AWS CLI

Sie können Bedingungen angeben, wenn Sie eine Regel erstellen oder ändern. Weitere Informationen finden Sie bei den Befehlen [create-rule](#) und [modify-rule](#). Die folgende Bedingung wird von Anforderungen mit einem Host-Header erfüllt, der mindestens einer der angegebenen Zeichenfolgen entspricht.

```
[
  {
    "Field": "host-header",
```

```
    "HostHeaderConfig": {  
      "Values": ["*.example.com"]  
    }  
  }  
]
```

Pfadbedingungen

Mit Pfadbedingungen können Sie Regeln definieren, mit denen Anforderungen auf Grundlage der URL in der Anforderung weitergeleitet werden (auch bekannt als pfadbasierte Weiterleitung)

Das Pfadmuster wird nur auf den Pfad der URL, nicht auf dessen Abfrageparameter, angewendet. Es wird nur auf sichtbare ASCII-Zeichen angewendet; Steuerzeichen (0x00 bis 0x1f und 0x7f) sind ausgeschlossen.

Die Regelauswertung wird erst durchgeführt, nachdem die URI-Normalisierung erfolgt ist.

Beim Pfadmuster wird die Groß-/Kleinschreibung berücksichtigt, es kann maximal 128 Zeichen lang sein und kann folgende Zeichen enthalten.

- A-Z, a-z, 0-9
- _ - . \$ / ~ ' ' @ : +
- & (Verwendung von &#x26;)
- * (entspricht 0 oder mehr Zeichen)
- ? (entspricht genau 1 Zeichen)

Wenn die Protokollversion gRPC ist, können Bedingungen für ein Paket, einen Dienst oder eine Methode spezifisch sein.

Beispiel für HTTP-Pfadmuster

- /img/*
- /img/*/pics

Beispiel für gRPC-Pfadmuster

- /paket
- /paket.dienst

- /paket.dienst/methode

Das Pfadmuster wird verwendet, um Anforderungen weiterzuleiten. Die Anforderungen werden bei diesem Vorgang aber nicht geändert. Wenn eine Regel beispielsweise das Pfadmuster `/img/*` aufweist, leitet die Regel eine Anforderung von `/img/picture.jpg` an die angegebene Zielgruppe als Anforderung von `/img/picture.jpg` weiter.

Example Beispiel für eine Pfadmusterbedingung für AWS CLI

Sie können Bedingungen angeben, wenn Sie eine Regel erstellen oder ändern. Weitere Informationen finden Sie bei den Befehlen [create-rule](#) und [modify-rule](#). Die folgende Bedingung wird von Anforderungen mit einer URL erfüllt, die die angegebene Zeichenfolge enthält.

```
[
  {
    "Field": "path-pattern",
    "PathPatternConfig": {
      "Values": ["/img/*"]
    }
  }
]
```

Abfragezeichenfolgebedingungen

Mit Abfragezeichenfolgebedingungen können Sie Regeln konfigurieren, mit denen Anforderungen auf Grundlage von Schlüssel/Wert-Paaren oder Werten in der Abfragezeichenfolge weitergeleitet werden. Bei der Übereinstimmungsbewertung wird die Groß- und Kleinschreibung nicht beachtet. Die folgenden Platzhalterzeichen werden unterstützt: `*` (findet Übereinstimmungen mit 0 oder mehr Zeichen) und `?` (findet Übereinstimmungen für genau 1 Zeichen).

Example Beispiel für eine Abfragezeichenfolge für die AWS CLI

Sie können Bedingungen angeben, wenn Sie eine Regel erstellen oder ändern. Weitere Informationen finden Sie bei den Befehlen [create-rule](#) und [modify-rule](#). Die folgende Bedingung wird von Abfragen mit einer Abfragezeichenfolge erfüllt, die entweder ein Schlüssel/Wert-Paar oder „version=v1“ enthält oder bei der ein beliebiger Schlüssel auf „example“ festgelegt ist.

```
[
  {
    "Field": "query-string",
```



```
"QueryStringConfig": {
  "Values": [
    {
      "Key": "version",
      "Value": "v1"
    },
    {
      "Value": "*example*"
    }
  ]
}
```

Bedingungen für die Quell-IP-Adresse

Mit Bedingungen für die Quell-IP-Adresse können Sie Regeln konfigurieren, mit denen Anforderungen auf Grundlage der Quell-IP-Adresse der Anforderung weitergeleitet werden. Die IP-Adresse muss im CIDR-Format angegeben werden. Sie können IPv4- und IPv6-IP-Adressen verwenden. Platzhalterzeichen werden nicht unterstützt. Sie können den 255.255.255.255/32-CIDR für die Quell-IP-Regelbedingung nicht angeben.

Befindet sich ein Client hinter einem Proxy, ist dies die IP-Adresse des Proxys, nicht die des Clients.

Diese Bedingung wird nicht durch Adressen im X-Forwarded-For-Header erfüllt. Sie können mit einer `http-header`-Bedingung im X-Forwarded-For-Header nach Adressen suchen.

Example Beispiel für eine Quell-IP-Bedingung für die AWS CLI

Sie können Bedingungen angeben, wenn Sie eine Regel erstellen oder ändern. Weitere Informationen finden Sie bei den Befehlen [create-rule](#) und [modify-rule](#). Die Folgende Bedingung wird von Anforderungen mit einer Quell-IP-Adresse in einem der angegebenen CIDR-Blöcken erfüllt.

```
[
  {
    "Field": "source-ip",
    "SourceIpConfig": {
      "Values": ["192.0.2.0/24", "198.51.100.10/32"]
    }
  }
]
```

Erstellen eines HTTP-Listeners für Ihren Application Load Balancer

Ein Listener überprüft Verbindungsanforderungen. Sie definieren einen Listener, wenn Sie Ihren Load Balancer erstellen, und Sie können Listener jederzeit zu Ihrem Load Balancer hinzufügen.

Die Informationen auf dieser Seite helfen Ihnen bei der Erstellung eines HTTP-Listeners für Ihren Load Balancer. Informationen zum Hinzufügen eines HTTPS-Listeners zu Ihrem Load Balancer finden Sie unter [Erstellen eines HTTPS-Listeners für Ihren Application Load Balancer](#).

Voraussetzungen

- Um eine Weiterleitungsaktion zur Standard-Listener-Regel hinzuzufügen, müssen Sie eine verfügbare Zielgruppe angeben. Weitere Informationen finden Sie unter [Erstellen einer Zielgruppe](#).
- Sie können dieselbe Zielgruppe in mehreren Listenern angeben, aber diese Listener müssen demselben Load Balancer angehören. Um eine Zielgruppe mit einem Load Balancer zu verwenden, müssen Sie sicherstellen, dass sie nicht von einem Listener verwendet wird, der einem anderen Load Balancer angehört.

Hinzufügen eines HTTP-Listeners

Sie konfigurieren einen Listener mit einem Protokoll und einem Port für Verbindungen von Clients zum Load Balancer und einer Zielgruppe für die standardmäßige Listener-Regel. Weitere Informationen finden Sie unter [Listener-Konfiguration](#).

So fügen Sie einen HTTP-Listener mithilfe der Konsole hinzu

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Wählen Sie auf der Registerkarte Listener und Regeln die Option Listener hinzufügen aus.
5. Wählen Sie unter Protokoll : Port die Option HTTP aus und belassen Sie den Standardport bei oder wählen Sie einen anderen Port aus.
6. Wählen Sie unter Standardaktionen eine der folgenden Optionen:
 - Zu Zielgruppen weiterleiten – Wählen Sie eine oder mehrere Zielgruppen aus, an die der Datenverkehr weitergeleitet werden soll. Um Zielgruppen hinzuzufügen, wählen Sie Zielgruppe hinzufügen aus. Wenn Sie mehr als eine Zielgruppe verwenden, wählen Sie für

jede Zielgruppe eine Gewichtung aus und überprüfen Sie den zugehörigen Prozentsatz. Sie müssen die Stickiness auf Gruppenebene für eine Regel aktivieren, wenn Sie die Stickiness für eine oder mehrere Zielgruppen aktiviert haben.

- Umleitung zu URL – Geben Sie die URL an, an die Client-Anforderungen umgeleitet werden. Das ist möglich, indem Sie jeden Teil einzeln auf der Registerkarte URI-Teile eingeben oder indem Sie die vollständige Adresse auf der Registerkarte Vollständige URL eingeben. Für Statuscode können Sie Weiterleitungen je nach Bedarf entweder als temporär (HTTP 302) oder permanent (HTTP 301) konfigurieren.
- Feststehende Antwort zurückgeben – Geben Sie den Antwortcode an, der bei verworfenen Client-Anforderungen zurückgegeben wird. Darüber hinaus können Sie den Inhaltstyp und den Antworttext angeben. Diese sind jedoch nicht erforderlich.

7. Wählen Sie Hinzufügen aus.

Um einen HTTP-Listener hinzuzufügen, verwenden Sie den AWS CLI

Verwenden Sie den Befehl [create-listener](#), um den Listener und die Standardregel zu erstellen, und den Befehl [create-rule](#), um zusätzliche Listener-Regeln zu definieren.

Erstellen eines HTTPS-Listeners für Ihren Application Load Balancer

Ein Listener überprüft Verbindungsanforderungen. Sie definieren einen Listener, wenn Sie Ihren Load Balancer erstellen, und Sie können Listener jederzeit zu Ihrem Load Balancer hinzufügen.

Um einen HTTPS-Listener zu erstellen, müssen Sie im Load Balancer mindestens ein SSL-Serverzertifikat bereitstellen. Der Load Balancer verwendet ein Serverzertifikat, um die Frontend-Verbindung zu beenden und dann Anfragen von Clients zu entschlüsseln, bevor er sie an die Ziele sendet. Sie müssen außerdem eine Sicherheitsrichtlinie angeben, die verwendet wird, um sichere Verbindungen zwischen Clients und dem Load Balancer auszuhandeln.

Wenn Sie verschlüsselten Datenverkehr an Ziele weiterleiten müssen, ohne dass der Load Balancer diesen Datenverkehr entschlüsselt, können Sie einen Network Load Balancer oder Classic Load Balancer mit einem TCP-Listener an Port 443 erstellen. Bei einem TCP-Listener leitet der Load Balancer verschlüsselten Datenverkehr an die Ziele weiter, ohne ihn zu entschlüsseln.

Application Load Balancer unterstützen keine ED25519-Schlüssel.

Die Informationen auf dieser Seite helfen Ihnen bei der Erstellung eines HTTPS-Listeners für Ihren Load Balancer. Informationen zum Hinzufügen eines HTTP-Listeners zu Ihrem Load Balancer finden Sie unter [Erstellen eines HTTP-Listeners für Ihren Application Load Balancer](#).

Inhalt

- [SSL-Zertifikate](#)
 - [Standardzertifikat](#)
 - [Zertifikatliste](#)
 - [Zertifikatserneuerung](#)
- [Sicherheitsrichtlinien](#)
 - [TLS-1.3-Sicherheitsrichtlinien](#)
 - [FIPS-Sicherheitsrichtlinien](#)
 - [Von FS unterstützte Richtlinien](#)
 - [Sicherheitsrichtlinien für TLS 1.0 — 1.2](#)
 - [TLS-Protokolle und Chiffren](#)
- [Hinzufügen eines HTTPS-Listeners](#)

SSL-Zertifikate

Der Load Balancer erfordert X.509-Zertifikate (SSL/TLS-Serverzertifikate). Zertifikate sind eine digitale Methode zur Identifizierung. Sie werden von einer Zertifizierungsstelle (Certificate Authority, CA) ausgestellt. Ein Zertifikat enthält Identifizierungsdaten, einen Gültigkeitszeitraum, den öffentlichen Schlüssel, eine Seriennummer und die digitale Signatur des Ausstellers.

Wenn Sie ein Zertifikat zur Verwendung mit Ihrem Load Balancer erstellen, müssen Sie einen Domainnamen angeben. Der Domainname auf dem Zertifikat muss mit dem Datensatz für den benutzerdefinierten Domainnamen übereinstimmen, damit wir die TLS-Verbindung überprüfen können. Stimmen sie nicht überein, wird der Datenverkehr nicht verschlüsselt.

Sie müssen einen vollqualifizierten Domainnamen (Fully Qualified Domain Name, FQDN) für Ihr Zertifikat wie `www.example.com` oder einen Apex-Domainnamen wie `example.com` angeben. Sie können auch ein Sternchen (*) als Platzhalter verwenden, um mehrere Webseitenamen in derselben Domain zu schützen. Wenn Sie ein Platzhalter-Zertifikat anfordern, muss sich das Sternchen (*) ganz links im Domainnamen befinden und es kann nur eine Subdomain-Ebene geschützt werden. `*.example.com` schützt beispielsweise `corp.example.com` und `images.example.com`,

aber es kann `test.login.example.com` nicht schützen. Beachten Sie außerdem, dass `*.example.com` nur die Subdomains von `example.com` schützt, jedoch nicht die bare- oder apex-Domain (`example.com`). Der Platzhaltername wird im Feld Subjekt und in der Erweiterung Alternativer Subjekt-Name des ACM-Zertifikats angezeigt. Weitere Informationen zum Anfordern öffentlicher Zertifikate finden Sie unter [Anfordern eines öffentlichen Zertifikats](#) im AWS Certificate Manager -Benutzerhandbuch.

Wir empfehlen, Zertifikate für Ihren Load Balancer mit [AWS Certificate Manager \(ACM\)](#) zu erstellen. ACM unterstützt RSA-Zertifikate mit Schlüssellängen von 2 048, 3 072 und 4 096 Bit sowie alle ECDSA-Zertifikate. ACM lässt sich in Elastic Load Balancing integrieren, sodass Sie das Zertifikat in Ihrem Load Balancer bereitstellen können. Weitere Informationen finden Sie im [AWS Certificate Manager -Benutzerhandbuch](#).

Alternativ können Sie SSL/TLS-Tools verwenden, um eine Zertifikatssignieranforderung (CSR) zu erstellen, die CSR dann von einer Zertifizierungsstelle signieren zu lassen, um ein Zertifikat zu erstellen, und das Zertifikat anschließend in ACM importieren oder das Zertifikat in (IAM) hochladen. AWS Identity and Access Management Weitere Informationen zum Importieren von Zertifikaten in ACM finden Sie unter [Importieren von Zertifikaten](#) im AWS Certificate Manager -Leitfaden. Weitere Informationen zum Hochladen von Zertifikaten in IAM finden Sie unter [Arbeiten mit Serverzertifikaten](#) im IAM-Benutzerhandbuch.

Standardzertifikat

Wenn Sie einen HTTPS-Listener erstellen, müssen Sie genau ein Zertifikat angeben. Dieses Zertifikat wird als Standardzertifikat bezeichnet. Sie können das Standardzertifikat ersetzen, nachdem Sie den HTTPS-Listener erstellt haben. Weitere Informationen finden Sie unter [Ersetzen des Standardzertifikats](#).

Wenn Sie weitere Zertifikate in einer [Zertifikatliste](#) angeben, wird das Standardzertifikat nur verwendet, wenn ein Client eine Verbindung ohne SNI- (Server Name Indication)-Protokoll herstellt, um einen Hostnamen anzugeben, oder falls keine passenden Zertifikate in der Zertifikatliste gefunden werden.

Wenn Sie keine weiteren Zertifikate angeben, aber mehrere sichere Anwendungen über einen einzelnen Load Balancer hosten müssen, können Sie ein Platzhalterzertifikat verwenden oder Ihrem Zertifikat einen SAN (Subject Alternative Name) für jede weitere Domain hinzufügen.

Zertifikatliste

Nach der Erstellung eines HTTPS-Listeners verfügt dieser über ein Standardzertifikat und eine leere Zertifikatliste. Optional können Sie der Zertifikatliste für den Listener Zertifikate hinzufügen. Ein Load Balancer kann dann mehrere Domains über denselben Port unterstützen und ein anderes Zertifikat für jede Domain bereitstellen. Weitere Informationen finden Sie unter [Hinzufügen von Zertifikaten zu einer Zertifikatliste](#).

Der Load Balancer verwendet einen intelligenten Algorithmus für die Zertifikatsauswahl, bei dem SNI unterstützt wird. Wenn der von einem Client bereitgestellte Hostname nur mit einem Zertifikat in der Zertifikatliste übereinstimmt, wählt der Load Balancer das entsprechende Zertifikat aus. Wenn ein von einem Client bereitgestellter Hostname mehreren Zertifikaten in der Zertifikatliste entspricht, wählt der Load Balancer das beste vom Client unterstützte Zertifikat. Die Auswahl des Zertifikats basiert auf den folgenden Kriterien in der angegebenen Reihenfolge:

- Algorithmus für öffentlichen Schlüssel (ECDSA gegenüber RSA bevorzugt)
- Hash-Algorithmus (SHA gegenüber MD5 bevorzugt)
- Schlüssellänge (der längste Schlüssel wird bevorzugt)
- Gültigkeitszeitraum

Die Load Balancer-Zugriffsprotokolleinträge enthalten den vom Client angegebenen Hostnamen und das dem Client präsentierte Zertifikat. Weitere Informationen finden Sie unter [Zugriffsprotokolleinträge](#).

Zertifikatserneuerung

Jedes Zertifikat verfügt über einen Gültigkeitszeitraum. Sie müssen sicherstellen, dass Sie jedes Zertifikat für Ihren Load Balancer vor dem Ablauf des Gültigkeitszeitraum erneuern oder ersetzen. Dies schließt das Standardzertifikat und Zertifikate in der Zertifikatliste ein. Das Verlängern oder Ersetzen eines Zertifikats wirkt sich nicht auf Anforderungen aus, die bereits verarbeitet werden, von einem Load Balancer-Knoten empfangen wurden und deren Weiterleitung an ein fehlerfreies Ziel aussteht. Nachdem ein Zertifikat verlängert wurde, verwenden neue Anforderungen das verlängerte Zertifikat. Nachdem ein Zertifikat ersetzt wurde, verwenden neue Anforderungen das neue Zertifikat.

Sie können das Verlängern und Ersetzen von Zertifikaten folgendermaßen verwalten:

- Zertifikate, die von Ihrem Load AWS Certificate Manager Balancer bereitgestellt und dort bereitgestellt werden, können automatisch erneuert werden. ACM versucht, die Zertifikate zu

verlängern, bevor sie ablaufen. Weitere Informationen finden Sie unter [Verwaltete Erneuerung](#) im AWS Certificate Manager -Benutzerhandbuch.

- Wenn Sie ein Zertifikat in ACM importiert haben, müssen Sie das Ablaufdatum des Zertifikats überwachen und es vor dem Ablauf verlängern. Weitere Informationen finden Sie unter [Importieren von Zertifikaten](#) im AWS Certificate Manager -Benutzerhandbuch.
- Wenn Sie ein Zertifikat in IAM importiert haben, müssen Sie ein neues Zertifikat erstellen, das neue Zertifikat in ACM oder IAM importieren, es dem Load Balancer hinzufügen und das abgelaufene Zertifikat aus dem Load Balancer entfernen.

Sicherheitsrichtlinien

Elastic Load Balancing verwendet eine Secure Socket Layer (SSL)-Aushandlungskonfiguration, die als Sicherheitsrichtlinie bezeichnet wird, um SSL-Verbindungen zwischen einem Client und dem Load Balancer auszuhandeln. Eine Sicherheitsrichtlinie ist eine Kombination aus Protokollen und Verschlüsselungen. Das Protokoll stellt eine sichere Verbindung zwischen einem Client und einem Server her und stellt sicher, dass alle Daten, die zwischen dem Client und Ihres Load Balancers übertragen werden, privat sind. Ein Verschlüsselungsverfahren ist ein Algorithmus, der eine kodierte Nachricht mithilfe von Verschlüsselungsschlüsseln erstellt. Protokolle verwenden mehrere Verschlüsselungsverfahren zum Verschlüsseln von Daten über das Internet. Während der Verbindungsaushandlung präsentieren der Client und der Load Balancer eine Liste von Verschlüsselungsverfahren und Protokollen, die sie jeweils unterstützen, nach Priorität sortiert. Standardmäßig wird für die sichere Verbindung die erste Verschlüsselung auf der Liste des Servers ausgewählt, die mit einem der Verschlüsselungsverfahren des Clients übereinstimmt.

Überlegungen:

- Application Load Balancer unterstützen die erneute SSL-Aushandlung nur für Zielverbindungen.
- Application Load Balancer unterstützen keine benutzerdefinierten Sicherheitsrichtlinien.
- Die `ELBSecurityPolicy-TLS13-1-2-2021-06` Richtlinie ist die Standardsicherheitsrichtlinie für HTTPS-Listener, die mit dem erstellt wurden. AWS Management Console
- Die `ELBSecurityPolicy-2016-08` Richtlinie ist die Standardsicherheitsrichtlinie für HTTPS-Listener, die mit dem erstellt wurden. AWS CLI
- Wenn Sie einen HTTPS-Listener erstellen, müssen Sie eine Sicherheitsrichtlinie auswählen.
 - Wir empfehlen die `ELBSecurityPolicy-TLS13-1-2-2021-06` Sicherheitsrichtlinie, die TLS 1.3 beinhaltet und mit TLS 1.2 abwärtskompatibel ist.

- Sie können die Sicherheitsrichtlinie wählen, die für Front-End-Verbindungen verwendet wird, aber nicht für Back-End-Verbindungen.
 - Wenn Ihr HTTPS-Listener für Backend-Verbindungen eine TLS-1.3-Sicherheitsrichtlinie verwendet, wird die `ELBSecurityPolicy-TLS13-1-0-2021-06`-Sicherheitsrichtlinie verwendet. Andernfalls wird die `ELBSecurityPolicy-2016-08`-Sicherheitsrichtlinie für Backend-Verbindungen verwendet.
- Um die Compliance- und Sicherheitsstandards zu erfüllen, die die Deaktivierung bestimmter TLS-Protokollversionen erfordern, oder um ältere Clients zu unterstützen, die veraltete Verschlüsselungen benötigen, können Sie eine der Sicherheitsrichtlinien verwenden. `ELBSecurityPolicy-TLS-` Um die TLS-Protokollversion für Anfragen an Ihren Application Load Balancer anzuzeigen, aktivieren Sie die Zugriffsprotokollierung für Ihren Load Balancer und überprüfen Sie die entsprechenden Zugriffsprotokolleinträge. Weitere Informationen finden Sie unter [Zugriffsprotokolle für Ihren Application Load Balancer](#).
- Sie können einschränken, welche Sicherheitsrichtlinien Benutzern in Ihrem AWS-Konten Land zur Verfügung stehen, AWS Organizations indem Sie die [Elastic Load Balancing Balancing-Bedingungsschlüssel](#) in Ihren IAM- bzw. Service Control Policies (SCPs) verwenden. Weitere Informationen finden Sie unter [Service Control Policies \(SCPs\)](#) im Benutzerhandbuch AWS Organizations

TLS-1.3-Sicherheitsrichtlinien

Elastic Load Balancing bietet die folgenden TLS 1.3-Sicherheitsrichtlinien für Application Load Balancer:

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

FIPS-Sicherheitsrichtlinien

Important

Alle sicheren Listener, die an einen Application Load Balancer angeschlossen sind, müssen entweder FIPS-Sicherheitsrichtlinien oder Nicht-FIPS-Sicherheitsrichtlinien verwenden; sie können nicht gemischt werden. Wenn ein vorhandener Application Load Balancer über zwei oder mehr Listener verfügt, die Nicht-FIPS-Richtlinien verwenden, und Sie möchten, dass die Listener stattdessen FIPS-Sicherheitsrichtlinien verwenden, entfernen Sie alle Listener, bis nur noch einer vorhanden ist. Ändern Sie die Sicherheitsrichtlinie des Listeners in FIPS und erstellen Sie dann weitere Listener mithilfe von FIPS-Sicherheitsrichtlinien. Alternativ können Sie einen neuen Application Load Balancer mit neuen Listenern erstellen, die nur FIPS-Sicherheitsrichtlinien verwenden.

Der Federal Information Processing Standard (FIPS) ist ein US-amerikanischer und kanadischer Regierungsstandard, der die Sicherheitsanforderungen für kryptografische Module zum Schutz vertraulicher Informationen festlegt. Weitere Informationen finden Sie unter [Federal Information Processing Standard \(FIPS\) 140](#) auf der Seite AWS Cloud Security Compliance.

Alle FIPS-Richtlinien nutzen das FIPS-validierte kryptografische Modul von AWS-LC. Weitere Informationen finden Sie auf der Seite [AWS-LC Cryptographic Module](#) auf der Website des NIST Cryptographic Module Validation Program.

Elastic Load Balancing bietet die folgenden FIPS-Sicherheitsrichtlinien für Application Load Balancer:

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

Von FS unterstützte Richtlinien

Elastic Load Balancing bietet die folgenden von FS (Forward Secrecy) unterstützten Sicherheitsrichtlinien für Application Load Balancer:

- ELBSecurityPolicy-FS-1-2-Res-2020-10
- ELBSecurityPolicy-FS-1-2-Res-2019-08
- ELBSecurityPolicy-FS-1-2-2019-08
- ELBSecurityPolicy-FS-1-1-2019-08
- ELBSecurityPolicy-FS-2018-06

Sicherheitsrichtlinien für TLS 1.0 — 1.2

Elastic Load Balancing bietet die folgenden Sicherheitsrichtlinien für TLS 1.0 bis 1.2 für Application Load Balancer:

- ELBSecurityPolicy-TLS-1-2-Ext-2018-06
- ELBSecurityPolicy-TLS-1-2-2017-01
- ELBSecurityPolicy-TLS-1-1-2017-01
- ELBSecurityPolicy-2016-08
- ELBSecurityPolicy-TLS-1-0-2015-04
- ELBSecurityPolicy-2015-05(identisch mit) **ELBSecurityPolicy-2016-08**

TLS-Protokolle und Chiffren

TLS 1.3

In der folgenden Tabelle werden die unterstützten TLS-Protokolle und Chiffren für die verfügbaren TLS 1.3-Sicherheitsrichtlinien beschrieben.

Hinweis: Das ELBSecurityPolicy- Präfix wurde aus den Richtliniennamen in der Zeile mit den Sicherheitsrichtlinien entfernt.

Beispiel: Die Sicherheitsrichtlinie ELBSecurityPolicy-TLS13-1-2-2021-06 wird als angezeigt TLS13-1-2-2021-06.

Sicherheitsrichtlinien	TLS13-1-2-2021-06	TLS13-1-3-2021-06	TLS13-1-2-Res-2021-06	TLS13-1-2-Ext2-2021-06	TLS13-1-2-Ext1-2021-06	TLS13-1-1-2021-06	TLS13-1-0-2021-06
TLS-Protokolle							
Protocol-TLSv1							✓
Protocol-TLSv1.1						✓	✓
Protocol-TLSv1.2	✓		✓	✓	✓	✓	✓
Protokoll-TLSv1.3	✓	✓	✓	✓	✓	✓	✓
TLS-Verschlüsselungsverfahren							
TLS_AES_128_GCM_SHA256	✓	✓	✓	✓	✓	✓	✓
TLS_AES_256_GCM_SHA384	✓	✓	✓	✓	✓	✓	✓
TLS_CHACHA20_POLY1305_SHA256	✓	✓	✓	✓	✓	✓	✓
ECDHE-ECDSA-AES128	✓		✓	✓	✓	✓	✓

Sicherheitsrichtlinien	TLS13-1-2-2021-06	TLS13-1-3-2021-06	TLS13-1-2-Res-2021-06	TLS13-1-2-Ext2-2021-06	TLS13-1-2-Ext1-2021-06	TLS13-1-1-2021-06	TLS13-1-0-2021-06
-GCM- SHA256							
ECDHE- RSA- AES128- GCM- SHA256	✓		✓	✓	✓	✓	✓
ECDHE- ECDSA- AES128- SHA256	✓			✓	✓	✓	✓
ECDHE- RSA- AES128- SHA256	✓			✓	✓	✓	✓
ECDHE- ECDSA- AES128- SHA				✓		✓	✓
ECDHE- RSA- AES128- SHA				✓		✓	✓

Sicherheitsrichtlinien	TLS13-1-2-2021-06	TLS13-1-3-2021-06	TLS13-1-2-Res-2021-06	TLS13-1-2-Ext2-2021-06	TLS13-1-2-Ext1-2021-06	TLS13-1-1-2021-06	TLS13-1-0-2021-06
ECDHE- ECDSA- AES256- -GCM- SHA384	✓		✓	✓	✓	✓	✓
ECDHE- RSA- AES256- GCM- SHA384	✓		✓	✓	✓	✓	✓
ECDHE- ECDSA- AES256- SHA384	✓			✓	✓	✓	✓
ECDHE- RSA- AES256- SHA384	✓			✓	✓	✓	✓
ECDHE- RSA- AES256- SHA				✓		✓	✓
ECDHE- ECDSA- AES256- SHA				✓		✓	✓

Sicherheitsrichtlinien	TLS13-1-2-2021-06	TLS13-1-3-2021-06	TLS13-1-2-Res-2021-06	TLS13-1-2-Ext2-2021-06	TLS13-1-2-Ext1-2021-06	TLS13-1-1-2021-06	TLS13-1-0-2021-06
AES128-GCM-SHA256				✓	✓	✓	✓
AES128-SHA256				✓	✓	✓	✓
AES128-SHA				✓		✓	✓
AES256-GCM-SHA384				✓	✓	✓	✓
AES256-SHA256				✓	✓	✓	✓
AES256-SHA				✓		✓	✓

So erstellen Sie mit der CLI einen HTTPS-Listener, der eine TLS 1.3-Richtlinie verwendet

Verwenden Sie den Befehl [create-listener](#) mit einer beliebigen [TLS](#) 1.3-Sicherheitsrichtlinie.

Das Beispiel verwendet die Sicherheitsrichtlinie. `ELBSecurityPolicy-TLS13-1-2-2021-06`

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

So ändern Sie einen HTTPS-Listener für die Verwendung einer TLS 1.3-Richtlinie mithilfe der CLI

[Verwenden Sie den Befehl `modify-listener` mit einer beliebigen TLS 1.3-Sicherheitsrichtlinie.](#)

Das Beispiel verwendet die Sicherheitsrichtlinie. `ELBSecurityPolicy-TLS13-1-2-2021-06`

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

So zeigen Sie die von einem Listener verwendeten Sicherheitsrichtlinien mithilfe der CLI an

Verwenden Sie den Befehl [describe-listeners](#) zusammen mit dem Ihres Listenersarn.

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

So zeigen Sie die Konfiguration einer TLS 1.3-Sicherheitsrichtlinie mit der CLI an

Verwenden Sie den Befehl [describe-ssl-policies](#) mit einer beliebigen [TLS](#) 1.3-Sicherheitsrichtlinie.

Das Beispiel verwendet die Sicherheitsrichtlinie. `ELBSecurityPolicy-TLS13-1-2-2021-06`

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

FIPS

Important

Richtlinien `ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04` und `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04` werden nur aus Gründen der Kompatibilität mit älteren Versionen bereitgestellt. Sie verwenden zwar FIPS-Kryptografie unter Verwendung des FIPS140-Moduls, entsprechen aber möglicherweise nicht den neuesten NIST-Richtlinien für die TLS-Konfiguration.

In der folgenden Tabelle werden die unterstützten TLS-Protokolle und Chiffren für die verfügbaren FIPS-Sicherheitsrichtlinien beschrieben.

Hinweis: Das ELBSecurityPolicy- Präfix wurde aus den Richtliniennamen in der Zeile mit den Sicherheitsrichtlinien entfernt.

Beispiel: Die Sicherheitsrichtlinie ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04 wird als angezeigt TLS13-1-2-FIPS-2023-04.

Sicherheitsrichtlinien	TLS13-1-3-FIPS-2023-04	TLS13-1-2-Res-FIPS-2023-04	TLS13-1-2-FIPS-2023-04	TLS13-1-2-Ext0-FIPS-2023-04	TLS13-1-2-Ext1-FIPS-2023-04	TLS13-1-2-Ext2-FIPS-2023-04	TLS13-1-1-FIPS-2023-04	TLS13-1-0-FIPS-2023-04
TLS-Protokolle								
Protocol-TLSv1								✓
Protocol-TLSv1.1							✓	✓
Protocol-TLSv1.2		✓	✓	✓	✓	✓	✓	✓
Protokoll - TLSv1.3	✓	✓	✓	✓	✓	✓	✓	✓
TLS-Verschlüsselungsverfahren								
TLS_AES_128_GCM_SHA256		✓	✓	✓	✓	✓	✓	✓
TLS_AES_256_GCM_SHA384		✓	✓	✓	✓	✓	✓	✓

Sicherheitsrichtlinien	TLS13-1-3-FIPS-2023-04	TLS13-1-2-Res-FIPS-2023-04	TLS13-1-2-FIPS-2023-04	TLS13-1-2-Ext0-FIPS-2023-04	TLS13-1-2-Ext1-FIPS-2023-04	TLS13-1-2-Ext2-FIPS-2023-04	TLS13-1-1-FIPS-2023-04	TLS13-1-0-FIPS-2023-04
ECDHE-ECD SA-AES128 -GCM-SHA256		✓	✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES128-GCM-SHA256	✓		✓	✓	✓	✓	✓	✓
ECDHE-ECD SA-AES128 -SHA256			✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES128-SHA256			✓	✓	✓	✓	✓	✓

Sicherheitsrichtlinien	TLS13-1-3-FIPS-2023-04	TLS13-1-2-Res-FIPS-2023-04	TLS13-1-2-FIPS-2023-04	TLS13-1-2-Ext0-FIPS-2023-04	TLS13-1-2-Ext1-FIPS-2023-04	TLS13-1-2-Ext2-FIPS-2023-04	TLS13-1-1-FIPS-2023-04	TLS13-1-0-FIPS-2023-04
ECDHE-ECD SA- AES128 -SHA				✓		✓	✓	✓
ECDHE-RSA- AES128- SHA			✓			✓	✓	✓
ECDHE-ECD SA- AES256 -GCM- SHA3 84	✓	✓	✓	✓	✓	✓	✓	✓
ECDHE-RSA- AES256- GCM- SHA384	✓	✓	✓	✓	✓	✓	✓	✓

Sicherheitsrichtlinien	TLS13-1-3-FIPS-2023-04	TLS13-1-2-Res-FIPS-2023-04	TLS13-1-2-FIPS-2023-04	TLS13-1-2-Ext0-FIPS-2023-04	TLS13-1-2-Ext1-FIPS-2023-04	TLS13-1-2-Ext2-FIPS-2023-04	TLS13-1-1-FIPS-2023-04	TLS13-1-0-FIPS-2023-04
ECDHE-ECD SA- AES256- - SHA384			✓	✓	✓	✓	✓	✓
ECDHE-RSA- AES256- S HA384		✓	✓	✓	✓	✓	✓	✓
ECDHE-RSA- AES256- SHA				✓		✓	✓	✓
ECDHE-ECD SA- AES256 -SHA				✓		✓	✓	✓
AES128- GCM- SHA256					✓	✓	✓	✓

Sicherheitsrichtlinien	TLS13-1-3-FIPS-2023-04	TLS13-1-2-Res-FIPS-2023-04	TLS13-1-2-FIPS-2023-04	TLS13-1-2-Ext0-FIPS-2023-04	TLS13-1-2-Ext1-FIPS-2023-04	TLS13-1-2-Ext2-FIPS-2023-04	TLS13-1-1-FIPS-2023-04	TLS13-1-0-FIPS-2023-04
AES128-SHA256					✓	✓	✓	✓
AES128-SHA						✓	✓	✓
AES256-GCM-SHA384					✓	✓	✓	✓
AES256-SHA256					✓	✓	✓	✓
AES256-SHA						✓	✓	✓

So erstellen Sie mit der CLI einen HTTPS-Listener, der eine FIPS-Richtlinie verwendet

[Verwenden Sie den Befehl create-listener mit einer beliebigen FIPS-Sicherheitsrichtlinie.](#)

Das Beispiel verwendet die Sicherheitsrichtlinie. `ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04`

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

So ändern Sie einen HTTPS-Listener für die Verwendung einer FIPS-Richtlinie mithilfe der CLI

[Verwenden Sie den Befehl modify-listener mit einer beliebigen FIPS-Sicherheitsrichtlinie.](#)

Das Beispiel verwendet die Sicherheitsrichtlinie. `ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04`

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

So zeigen Sie die von einem Listener verwendeten Sicherheitsrichtlinien mithilfe der CLI an

Verwenden Sie den Befehl [describe-listeners](#) zusammen mit dem Ihres Listenersarn.

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

So zeigen Sie die Konfiguration einer FIPS-Sicherheitsrichtlinie mit der CLI an

[Verwenden Sie den Befehl describe-ssl-policies mit einer beliebigen FIPS-Sicherheitsrichtlinie.](#)

Das Beispiel verwendet die Sicherheitsrichtlinie. `ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04`

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

FS

In der folgenden Tabelle werden die unterstützten TLS-Protokolle und Chiffren für die verfügbaren FS-unterstützten Sicherheitsrichtlinien beschrieben.

Hinweis: Das `ELBSecurityPolicy-` Präfix wurde aus den Richtliniennamen in der Zeile mit den Sicherheitsrichtlinien entfernt.

Beispiel: Die Sicherheitsrichtlinie `ELBSecurityPolicy-FS-2018-06` wird als `FS-2018-06` angezeigt.

Sicherheitsrichtlinien	Default	FS-1-2-Res-2020-10	FS-1-2-Res-2019-08	FS-1-2-2019-08	FS-1-1-2019-08	FS-2018-06
TLS-Protokolle						
Protocol-TLSv1	✓					✓
Protocol-TLSv1.1	✓				✓	✓
Protocol-TLSv1.2	✓	✓	✓	✓	✓	✓
TLS-Verschlüsselungsverfahren						
ECDHE-ECDSA-AES128-GCM-SHA256	✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES128-GCM-SHA256	✓	✓	✓	✓	✓	✓
ECDHE-ECDSA-AES128-SHA256	✓		✓	✓	✓	✓

Sicherheitsrichtlinien	Default	FS-1-2-Res-2020-10	FS-1-2-Res-2019-08	FS-1-2-2019-08	FS-1-1-2019-08	FS-2018-06
ECDHE-RSA-AES128-SHA256	✓		✓	✓	✓	✓
ECDHE-ECDSA-AES128-SHA	✓			✓	✓	✓
ECDHE-RSA-AES128-SHA	✓			✓	✓	✓
ECDHE-ECDSA-AES256-GCM-SHA384	✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES256-GCM-SHA384	✓	✓	✓	✓	✓	✓

Sicherheitsrichtlinien	Default	FS-1-2-Res-2020-10	FS-1-2-Res-2019-08	FS-1-2-2019-08	FS-1-1-2019-08	FS-2018-06
ECDHE-ECDSA-AES256-SHA384	✓		✓	✓	✓	✓
ECDHE-RSA-AES256-SHA384	✓		✓	✓	✓	✓
ECDHE-RSA-AES256-SHA	✓			✓	✓	✓
ECDHE-ECDSA-AES256-SHA	✓			✓	✓	✓
AES128-GCM-SHA256	✓					
AES128-SHA256	✓					
AES128-SHA	✓					

Sicherheitsrichtlinien	Default	FS-1-2-Res-2020-10	FS-1-2-Res-2019-08	FS-1-2-2019-08	FS-1-1-2019-08	FS-2018-06
AES256-GCM-SHA384	✓					
AES256-SHA256	✓					
AES256-SHA	✓					

So erstellen Sie mit der CLI einen HTTPS-Listener, der eine von FS unterstützte Richtlinie verwendet

Verwenden Sie den Befehl [create-listener](#) mit jeder von [FS](#) unterstützten Sicherheitsrichtlinie.

Das Beispiel verwendet die Sicherheitsrichtlinie. `ELBSecurityPolicy-FS-2018-06`

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

So ändern Sie einen HTTPS-Listener mithilfe der CLI so, dass er eine von FS unterstützte Richtlinie verwendet

[Verwenden Sie den Befehl modify-listener mit jeder von FS unterstützten Sicherheitsrichtlinie.](#)

Das Beispiel verwendet die Sicherheitsrichtlinie. `ELBSecurityPolicy-FS-2018-06`

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

```
--ssl-policy ELBSecurityPolicy-FS-2018-06
```

So zeigen Sie die von einem Listener verwendeten Sicherheitsrichtlinien mithilfe der CLI an. Verwenden Sie den Befehl [describe-listeners](#) zusammen mit dem Namen Ihres Listeners.

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

So zeigen Sie die Konfiguration einer FS-unterstützten Sicherheitsrichtlinie mit der CLI an. Verwenden Sie den Befehl [describe-ssl-policies](#) mit jeder von [FS](#) unterstützten Sicherheitsrichtlinie.

Das Beispiel verwendet die Sicherheitsrichtlinie `ELBSecurityPolicy-FS-2018-06`.

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

TLS 1.0 - 1.2

In der folgenden Tabelle werden die unterstützten TLS-Protokolle und Chiffren für die verfügbaren TLS 1.0-1.2-Sicherheitsrichtlinien beschrieben.

Hinweis: Das `ELBSecurityPolicy-` Präfix wurde aus den Richtliniennamen in der Zeile mit den Sicherheitsrichtlinien entfernt.

Beispiel: Die Sicherheitsrichtlinie `ELBSecurityPolicy-TLS-1-2-Ext-2018-06` wird als `TLS-1-2-Ext-2018-06` angezeigt.

Sicherheitsrichtlinien	Default	TLS-1-2-Ext-2018-06	TLS-1-2-2017-01	TLS-1-1-2017-01	TLS-1-0-2015-04*
TLS-Protokolle					

Sicherheitsrichtlinien	Default	TLS-1-2-Ext-2018-06	TLS-1-2-2017-01	TLS-1-1-2017-01	TLS-1-0-2015-04*
Protocol-TLSv1	✓				✓
Protocol-TLSv1.1	✓			✓	✓
Protocol-TLSv1.2	✓	✓	✓	✓	✓
TLS-Verschlüsselungsverfahren					
ECDHE-ECD SA-AES128 -GCM-SHA2 56	✓	✓	✓	✓	✓
ECDHE-RSA -AES128-G CM-SHA256	✓	✓	✓	✓	✓
ECDHE-ECD SA-AES128- SHA256	✓	✓	✓	✓	✓
ECDHE-RSA -AES128-S HA256	✓	✓	✓	✓	✓

Sicherheitsrichtlinien	Default	TLS-1-2-Ext-2018-06	TLS-1-2-2017-01	TLS-1-1-2017-01	TLS-1-0-2015-04*
ECDHE-ECD SA-AES128- SHA	✓	✓		✓	✓
ECDHE-RSA -AES128-S HA	✓	✓		✓	✓
ECDHE-ECD SA-AES256 -GCM-SHA3 84	✓	✓	✓	✓	✓
ECDHE-RSA -AES256-G CM-SHA384	✓	✓	✓	✓	✓
ECDHE-ECD SA-AES256- SHA384	✓	✓	✓	✓	✓
ECDHE-RSA -AES256-S HA384	✓	✓	✓	✓	✓
ECDHE-RSA -AES256-S HA	✓	✓		✓	✓

Sicherheitsrichtlinien	Default	TLS-1-2-Ext-2018-06	TLS-1-2-2017-01	TLS-1-1-2017-01	TLS-1-0-2015-04*
ECDHE-ECD SA-AES256- SHA	✓	✓		✓	✓
AES128-GC M-SHA256	✓	✓	✓	✓	✓
AES128-SH A256	✓	✓	✓	✓	✓
AES128-SH A	✓	✓		✓	✓
AES256-GC M-SHA384	✓	✓	✓	✓	✓
AES256-SH A256	✓	✓	✓	✓	✓
AES256-SH A	✓	✓		✓	✓
DES-CBC3- SHA					✓

* Verwenden Sie diese Richtlinie nicht, es sei denn, Sie müssen einen veralteten Client unterstützen, der das DES-CBC3-SHA-Verschlüsselungsverfahren, ein schwächeres Verschlüsselungsverfahren, erfordert.

So erstellen Sie mit der CLI einen HTTPS-Listener, der eine TLS 1.0-1.2-Richtlinie verwendet

[Verwenden Sie den Befehl `create-listener` mit einer beliebigen Sicherheitsrichtlinie, die von TLS 1.0-1.2 unterstützt wird.](#)

Das Beispiel verwendet die Sicherheitsrichtlinie. `ELBSecurityPolicy-2016-08`

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

So ändern Sie einen HTTPS-Listener für die Verwendung einer TLS 1.0-1.2-Richtlinie mithilfe der CLI

[Verwenden Sie den Befehl `modify-listener` mit einer beliebigen Sicherheitsrichtlinie, die von TLS 1.0-1.2 unterstützt wird.](#)

Das Beispiel verwendet die Sicherheitsrichtlinie. `ELBSecurityPolicy-2016-08`

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

So zeigen Sie die von einem Listener verwendeten Sicherheitsrichtlinien mithilfe der CLI an

Verwenden Sie den Befehl [describe-listeners](#) zusammen mit dem Ihres Listenersarn.

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

So zeigen Sie die Konfiguration einer TLS 1.0-1.2-Sicherheitsrichtlinie mit der CLI an

[Verwenden Sie den Befehl `describe-ssl-policies` mit jeder Sicherheitsrichtlinie, die von TLS 1.0-1.2 unterstützt wird.](#)

Das Beispiel verwendet die Sicherheitsrichtlinie. `ELBSecurityPolicy-2016-08`

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

Hinzufügen eines HTTPS-Listeners

Sie konfigurieren einen Listener mit einem Protokoll und einem Port für Verbindungen von Clients zum Load Balancer und einer Zielgruppe für die standardmäßige Listener-Regel. Weitere Informationen finden Sie unter [Listener-Konfiguration](#).

Voraussetzungen

- Um einen HTTPS-Listener zu erstellen, müssen Sie ein Zertifikat und eine Sicherheitsrichtlinie angeben. Der Load Balancer verwendet ein Zertifikat, um die Verbindung zu beenden und Anforderungen von Clients zu entschlüsseln, bevor er sie an Ziele weiterleitet. Der Load Balancer verwendet die Sicherheitsrichtlinie beim Aushandeln von SSL-Verbindungen mit den Clients.
- Um eine Weiterleitungsaktion zur Standard-Listener-Regel hinzuzufügen, müssen Sie eine verfügbare Zielgruppe angeben. Weitere Informationen finden Sie unter [Erstellen einer Zielgruppe](#).
- Sie können dieselbe Zielgruppe in mehreren Listenern angeben, aber diese Listener müssen demselben Load Balancer angehören. Um eine Zielgruppe mit einem Load Balancer zu verwenden, müssen Sie sicherstellen, dass sie nicht von einem Listener verwendet wird, der einem anderen Load Balancer angehört.

So fügen Sie einen HTTPS-Listeners mithilfe der Konsole hinzu

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Wählen Sie auf der Registerkarte Listener und Regeln die Option Listener hinzufügen aus.
5. Wählen Sie unter Protokoll : Port die Option HTTPS aus und belassen Sie den Standardport bei oder wählen Sie einen anderen Port aus.
6. (Optional) Um die Authentifizierung zu aktivieren, wählen Sie unter Authentifizierung die Option OpenID oder Amazon Cognito verwenden aus und geben Sie die angeforderten Informationen ein. Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mithilfe eines Application Load Balancers](#).
7. Führen Sie für Default actions (Standardaktionen) einen der folgenden Schritte aus:
 - Zu Zielgruppen weiterleiten – Wählen Sie eine oder mehrere Zielgruppen aus, an die der Datenverkehr weitergeleitet werden soll. Um Zielgruppen hinzuzufügen, wählen Sie Zielgruppe hinzufügen aus. Wenn Sie mehr als eine Zielgruppe verwenden, wählen Sie für

jede Zielgruppe eine Gewichtung aus und überprüfen Sie den zugehörigen Prozentsatz. Sie müssen die Stickiness auf Gruppenebene für eine Regel aktivieren, wenn Sie die Stickiness für eine oder mehrere Zielgruppen aktiviert haben.

- Umleitung zu URL – Geben Sie die URL an, an die Client-Anforderungen umgeleitet werden. Das ist möglich, indem Sie jeden Teil einzeln auf der Registerkarte URI-Teile eingeben oder indem Sie die vollständige Adresse auf der Registerkarte Vollständige URL eingeben. Für Statuscode können Sie Weiterleitungen je nach Bedarf entweder als temporär (HTTP 302) oder permanent (HTTP 301) konfigurieren.
 - Feststehende Antwort zurückgeben – Geben Sie den Antwortcode an, der bei verworfenen Client-Anforderungen zurückgegeben wird. Darüber hinaus können Sie den Inhaltstyp und den Antworttext angeben. Diese sind jedoch nicht erforderlich.
8. Wir empfehlen, dass Sie für Sicherheitsrichtlinie immer die neueste vordefinierte Sicherheitsrichtlinie verwenden.
9. Für Standard-SSL-/TLS-Zertifikat sind die folgenden Optionen verfügbar:
- Wenn Sie ein Zertifikat mit erstellt oder importiert haben AWS Certificate Manager, wählen Sie Aus ACM und dann das Zertifikat unter Zertifikat auswählen aus.
 - Wenn Sie ein Zertifikat mit IAM importiert haben, wählen Sie Von ACM aus und wählen Sie dann das Zertifikat unter Zertifikat auswählen aus.
 - Wenn Sie ein Zertifikat importieren, aber ACM in Ihrer Region nicht verfügbar ist, wählen Sie Importieren und dann An IAM aus. Geben Sie im Feld Zertifikatname den Namen des Zertifikats ein. Kopieren Sie den Inhalt der privaten Schlüsseldatei (PEM-kodiert) und fügen Sie ihn in das Feld Privater Zertifikatsschlüssel ein. Kopieren Sie den Inhalt der öffentlichen Schlüsselzertifikatdatei (PEM-kodiert) und fügen Sie ihn in das Feld Zertifikatstext ein. Kopieren Sie den Inhalt der Zertifikatskettendatei (PEM-kodiert) und fügen Sie ihn in das Feld Certificate Chain (Zertifikats-Kette) ein, es sei denn, Sie verwenden ein selbst signiertes Zertifikat und es ist nicht wichtig, dass Browser das Zertifikat implizit akzeptieren.
10. (Optional) Um die gegenseitige Authentifizierung zu aktivieren, aktivieren Sie unter Verwaltung von Client-Zertifikaten die Option Gegenseitige Authentifizierung (mTLS).

Wenn diese Option aktiviert ist, ist der standardmäßige Modus für gegenseitiges TLS Passthrough.

Wenn Sie „Mit Trust Store verifizieren“ auswählen:

- Standardmäßig werden Verbindungen mit abgelaufenen Client-Zertifikaten abgelehnt. Um dieses Verhalten zu ändern, erweitern Sie die erweiterten mTLS-Einstellungen und wählen Sie dann unter Ablauf des Client-Zertifikats die Option Abgelaufene Client-Zertifikate zulassen aus.
- Wählen Sie unter Trust Store einen vorhandenen Trust Store aus, oder wählen Sie Neuer Trust Store aus.
 - Wenn Sie Neuer Vertrauensspeicher ausgewählt haben, geben Sie einen Namen für den Vertrauensspeicher, den Speicherort der S3-URI-Zertifizierungsstelle und optional einen Speicherort für die Sperrliste für S3-URI-Zertifikate an.

11. Wählen Sie Speichern.

Um einen HTTPS-Listener hinzuzufügen, verwenden Sie AWS CLI

Verwenden Sie den Befehl [create-listener](#), um den Listener und die Standardregel zu erstellen, und den Befehl [create-rule](#), um zusätzliche Listener-Regeln zu definieren.

Listener-Regeln für Ihren Application Load Balancer

Die Regeln, die Sie für Ihren Listener definieren, bestimmen, wie der Load Balancer Anforderungen an die Ziele in einer oder mehreren Zielgruppen weiterleitet.

Jede Rolle besteht aus einer Priorität, mindestens einer Aktion und mindestens einer Bedingung. Weitere Informationen finden Sie unter [Listener-Regeln](#).

Voraussetzungen

- Regeln können nur sicheren Listenern angehängt werden.
- Jede Regel muss genau eine der folgenden Aktionen enthalten: `forward`, `redirect` oder `fixed-response`. Außerdem muss es die letzte auszuführende Regel sein.
- Jede Rolle kann keine oder eine der folgenden Bedingungen enthalten: `host-header`, `http-request-method`, `path-pattern` und `source-ip` sowie keine oder mehrere der folgenden Bedingungen; `http-header` und `query-string`.
- Sie können bis zu drei Vergleichszeichenfolgen pro Bedingung und bis zu fünf pro Regel angeben.
- Eine `forward`-Aktion leitet Anforderungen an ihre Zielgruppe weiter. Bevor Sie eine `forward`-Aktion hinzufügen, erstellen Sie ihre Zielgruppe und fügen Sie Ziele hinzu. Weitere Informationen finden Sie unter [Erstellen einer Zielgruppe](#).

Hinzufügen einer Regel

Sie definieren eine Standardregel, wenn Sie einen Listener erstellen, und Sie können jederzeit zusätzliche nicht standardmäßige Regeln definieren.

Hinzufügen einer Regel mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus, um die dazugehörigen Details anzuzeigen.
4. Führen Sie auf der Registerkarte Listener und Regeln einen der folgenden Schritte durch:
 - a. Wählen Sie den Text in der Spalte Protokoll: Port aus, um die Detailseite für den Listener zu öffnen.

Wählen Sie auf der Registerkarte Regeln die Option Regel hinzufügen aus.

- b. Wählen Sie den Listener aus, dem Sie eine Regel hinzufügen möchten.

Wählen Sie Regeln verwalten und dann Regel hinzufügen aus.
5. Sie können unter Name und Tags einen Namen für Ihre Regel angeben, obwohl dies nicht erforderlich ist.

Um zusätzliche Tags hinzuzufügen, wählen Sie Hinzufügen zusätzlicher Tags aus.

6. Wählen Sie Weiter aus.
7. Klicken Sie auf Bedingung hinzufügen.
8. Fügen Sie mindestens eine der folgenden Bedingungen hinzu:
 - Host-Header – Definieren Sie den Host-Header. Zum Beispiel: `*.example.com`. Wählen Sie Bestätigen aus, um die Bedingung zu speichern.

Maximal 128 Zeichen. Die Groß- und Kleinschreibung muss nicht berücksichtigt werden. Zulässige Zeichen sind a-z, A-Z, 0-9; die folgenden Sonderzeichen: `-_.`; und Platzhalter (`*` und `?`).

- Pfad – Definieren Sie den Pfad. Zum Beispiel: `/item/*`. Wählen Sie Bestätigen aus, um die Bedingung zu speichern.

Maximal 128 Zeichen. Groß-/Kleinschreibung ist zu beachten. Zulässige Zeichen sind a-z, A-Z, 0-9; die folgenden Sonderzeichen: `_-.$/~"@"+;` &; und Platzhalter (`*` und `?`).

- HTTP-Anforderungsmethode – Definieren Sie die HTTP-Anforderungsmethode. Wählen Sie Bestätigen aus, um die Bedingung zu speichern.

Maximal 40 Zeichen. Groß-/Kleinschreibung ist zu beachten. Zulässige Zeichen sind A-Z und die folgenden Sonderzeichen: -_. Platzhalter werden nicht unterstützt.

- Quell-IP – Definieren Sie die Quell-IP-Adresse im CIDR-Format. Wählen Sie Bestätigen aus, um die Bedingung zu speichern.

Sowohl IPv4- als auch IPv6-CIDRs sind zulässig. Platzhalter werden nicht unterstützt.

- HTTP-Header – Geben Sie den Namen des Headers ein und fügen Sie mindestens eine Vergleichszeichenfolge ein. Wählen Sie Bestätigen aus, um die Bedingung zu speichern.
- HTTP-Header-Name – Die Regel bewertet Anforderungen, die diesen Header enthalten, um zu bestätigen, dass die Werte übereinstimmen.

Maximal 40 Zeichen. Die Groß- und Kleinschreibung muss nicht berücksichtigt werden. Zulässige Zeichen sind a-z, A-Z, 0-9 und die folgenden Sonderzeichen: *? !# \$ % & ' + . ^ _ ` | ~. Platzhalter werden nicht unterstützt.

- HTTP-Header-Wert – Geben Sie Zeichenfolgen ein, die mit dem HTTP-Header-Wert verglichen werden sollen.

Maximal 128 Zeichen. Die Groß- und Kleinschreibung muss nicht berücksichtigt werden. Zulässige Zeichen sind a-z, A-Z, 0-9; Leerzeichen; die folgenden Sonderzeichen: ! " # \$ % & ' () + , . / : ; # = > @ [] ^ _ { } ~ - ; und Platzhalter (* und ?).

- Abfragezeichenfolge – Sie können Anforderungen basierend auf Schlüssel/Wert-Paaren oder Werten in den Abfragezeichenfolgen weiterleiten. Wählen Sie Bestätigen aus, um die Bedingung zu speichern.

Maximal 128 Zeichen. Die Groß- und Kleinschreibung muss nicht berücksichtigt werden. Zulässige Zeichen sind a-z, A-Z, 0-9; die folgenden Sonderzeichen: _ . \$ / ~ " @ : + & () ! , ; = ; und Platzhalter (* und ?).

9. Wählen Sie Weiter aus.

10. Definieren Sie eine der folgenden Aktionen für Ihre Regel:

- Zu Zielgruppen weiterleiten – Wählen Sie eine oder mehrere Zielgruppen aus, an die der Datenverkehr weitergeleitet werden soll. Um Zielgruppen hinzuzufügen, wählen Sie Zielgruppe hinzufügen aus. Wenn Sie mehr als eine Zielgruppe verwenden, wählen Sie für jede Zielgruppe eine Gewichtung aus und überprüfen Sie den zugehörigen Prozentsatz. Sie

müssen die Stickiness auf Gruppenebene für eine Regel aktivieren, wenn Sie die Stickiness für eine oder mehrere Zielgruppen aktiviert haben.

- Umleitung zu URL – Geben Sie die URL an, an die Client-Anforderungen umgeleitet werden. Das ist möglich, indem Sie jeden Teil einzeln auf der Registerkarte URI-Teile eingeben oder indem Sie die vollständige Adresse auf der Registerkarte Vollständige URL eingeben. Für Statuscode können Sie Weiterleitungen je nach Bedarf entweder als temporär (HTTP 302) oder permanent (HTTP 301) konfigurieren.
- Feststehende Antwort zurückgeben – Geben Sie den Antwortcode an, der bei verworfenen Client-Anforderungen zurückgegeben wird. Darüber hinaus können Sie den Inhaltstyp und den Antworttext angeben. Diese sind jedoch nicht erforderlich.

11. Wählen Sie Weiter aus.

12. Geben Sie die Priorität Ihrer Regel an, indem Sie einen Wert zwischen 1 und 50000 eingeben.

13. Wählen Sie Weiter aus.

14. Überprüfen Sie alle Details und Einstellungen, die derzeit für Ihre neue Regel konfiguriert sind. Wenn Sie mit Ihrer Auswahl zufrieden sind, wählen Sie Erstellen aus.

Um eine Regel hinzuzufügen, verwenden Sie AWS CLI

Verwenden Sie den Befehl [create-rule](#) zum Erstellen der Regel. Verwenden Sie den Befehl [describe-rules](#), um Informationen zur Regel anzuzeigen.

Bearbeiten einer Regel

Sie können die Aktion und Bedingungen für eine Regel jederzeit bearbeiten. Regelaktualisierungen werden nicht sofort wirksam, so dass Anforderungen nach dem Aktualisieren einer Regel für kurze Zeit mit der vorherigen Regelkonfiguration weitergeleitet werden können. Alle in der Übertragung befindlichen Anforderungen sind abgeschlossen.

Bearbeiten einer Regel mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Führen Sie auf der Registerkarte Listener und Regeln einen der folgenden Schritte durch:

- Wählen Sie den Text in der Spalte Protokoll: Port aus, um die Detailseite für den Listener zu öffnen.
 - i. Wählen Sie auf der Registerkarte Regeln im Abschnitt Listener-Regeln den Text in der Spalte Namens-Tag für die Regel aus, die Sie bearbeiten möchten.

Wählen Sie Aktionen und dann Regel bearbeiten aus.
 - ii. Wählen Sie auf der Registerkarte Regeln im Abschnitt Listener-Regeln die Regel aus, die Sie bearbeiten möchten.

Wählen Sie Aktionen und dann Regel bearbeiten aus.
- 5. Ändern Sie den Namen und die Tags nach Bedarf. Um zusätzliche Tags hinzuzufügen, wählen Sie Hinzufügen zusätzlicher Tags aus.
- 6. Wählen Sie Weiter
- 7. Ändern Sie die Bedingungen nach Bedarf. Sie können Bedingungen hinzufügen, bestehende bearbeiten oder löschen.
- 8. Wählen Sie Weiter
- 9. Ändern Sie die Aktionen nach Bedarf.
- 10. Wählen Sie Weiter
- 11. Ändern Sie die Regelpriorität nach Bedarf. Sie können einen Wert zwischen 1 und 50000 eingeben.
- 12. Wählen Sie Weiter
- 13. Überprüfen Sie alle Details und aktualisierten Einstellungen, die für Ihre Regel konfiguriert wurden. Wenn Sie mit Ihrer Auswahl zufrieden sind, wählen Sie Änderungen speichern.

Um eine Regel mit dem zu bearbeiten AWS CLI

Verwenden Sie den Befehl [modify-rule](#).

Aktualisieren der Regelpriorität

Regeln werden in der Reihenfolge ihrer Prioritäten bewertet, ausgehend vom niedrigsten Wert hin zum höchsten Wert. Die Standardregel wird zuletzt ausgewertet. Sie können die Priorität einer nicht standardmäßigen Regel jederzeit ändern. Sie können die Priorität der Standardregel nicht ändern.

Um die Regelpriorität mithilfe der Konsole zu aktualisieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Führen Sie auf der Registerkarte Listener und Regeln einen der folgenden Schritte durch:
 - a. Wählen Sie den Text in den Spalten Protokoll: Port oder Regeln aus, um die Detailseite für den Listener zu öffnen.
 - i. Wählen Sie Aktionen und anschließend Regeln neu priorisieren aus.
 - ii. Wählen Sie auf der Registerkarte Regeln im Abschnitt Listener-Regeln die Option Aktionen und dann Regeln neu priorisieren aus.
 - b. Wählen Sie den Listener aus.
 - Wählen Sie Regeln verwalten und dann Regeln neu priorisieren aus.
5. Im Bereich Listener-Regeln wird in der Spalte Priorität die aktuelle Regelpriorität angezeigt. Sie können die Priorität einer Regel aktualisieren, indem Sie einen Wert zwischen 1 und 50000 eingeben.
6. Wenn Sie mit Ihren Änderungen zufrieden sind, klicken Sie auf Änderungen speichern.

Um die Regelprioritäten zu aktualisieren, verwenden Sie den AWS CLI

Verwenden Sie den Befehl [set-rule-priorities](#).

Löschen einer Regel

Sie können die nicht standardmäßigen Regeln für einen Listener jederzeit ändern. Sie können die Standardregel für einen Listener nicht löschen. Wenn Sie einen Listener löschen, werden all seine Regeln gelöscht.

Löschen einer Regel mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Führen Sie auf der Registerkarte Listener und Regeln einen der folgenden Schritte durch:

- a. Wählen Sie den Text in den Spalten Protokoll: Port oder Regeln aus, um die Detailseite für den Listener zu öffnen.
 - i. Wählen Sie die Regel aus, die Sie löschen möchten.
 - ii. Wählen Sie Aktionen und dann Regel löschen aus.
 - iii. Geben Sie `confirm` in das Texteingabefeld ein und wählen Sie dann Löschen aus.
- b. Wählen Sie den Text in der Spalte Namens-Tag aus, um die Detailseite für die Regel zu öffnen.
 - i. Wählen Sie Aktionen und dann Regel löschen aus.
 - ii. Geben Sie `confirm` in das Texteingabefeld ein und wählen Sie dann Löschen aus.

Um eine Regel mit dem zu löschen AWS CLI

Verwenden Sie den Befehl [delete-rule](#).

Aktualisieren eines HTTPS-Listeners für Ihren Application Load Balancer

Nach der Erstellung eines HTTPS-Listeners können Sie das Standardzertifikat ersetzen, die Zertifikatliste aktualisieren oder die Sicherheitsrichtlinie ersetzen.

Aufgaben

- [Ersetzen des Standardzertifikats](#)
- [Hinzufügen von Zertifikaten zu einer Zertifikatliste](#)
- [Entfernen eines Zertifikats aus der Zertifikatliste](#)
- [Aktualisieren der Sicherheitsrichtlinie](#)

Ersetzen des Standardzertifikats

Sie können das Standardzertifikat für den Listener mit den folgenden Schritten ersetzen. Weitere Informationen finden Sie unter [SSL-Zertifikate](#).

So ändern Sie das Standardzertifikat mit der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Wählen Sie auf der Registerkarte Listener und Regeln den Text in der Spalte Protokoll: Port aus, um die Detailseite für den Listener zu öffnen.
5. Wählen Sie auf der Registerkarte Zertifikate die Option Standard ändern aus.
6. Wählen Sie in der Tabelle ACM- und IAM-Zertifikate ein neues Standardzertifikat aus.
7. Wählen Sie Als Standard speichern aus.

Um das Standardzertifikat mit dem zu ändern AWS CLI

Verwenden Sie den Befehl [modify-listener](#).

Hinzufügen von Zertifikaten zu einer Zertifikatliste

Sie können der Zertifikatliste für den Listener mit den folgenden Schritten Zertifikate hinzufügen. Wenn Sie zum ersten Mal einen HTTPS-Listener erstellen, ist die Zertifikatliste leer. Sie können mindestens ein Zertifikat hinzufügen. Optional können Sie das Standardzertifikat hinzufügen, um sicherzustellen, dass dieses Zertifikat auch dann mit dem SNI-Protokoll verwendet wird, wenn es durch als das Standardzertifikat ersetzt wird. Weitere Informationen finden Sie unter [SSL-Zertifikate](#).

So ändern Sie das Standardzertifikat mit der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Wählen Sie auf der Registerkarte Listener und Regeln den Text in der Spalte Protokoll: Port aus, um die Detailseite für den Listener zu öffnen.
5. Wählen Sie auf der Registerkarte Zertifikate die Option Zertifikat hinzufügen aus.
6. Wählen Sie in der Tabelle ACM- und IAM-Zertifikate die Zertifikate aus, die Sie hinzufügen möchten, und klicken Sie unten auf Schließen Sie die unten angeführten als ausstehend ein.
7. Wenn Sie über ein Zertifikat verfügen, das nicht von ACM oder IAM verwaltet wird, wählen Sie Zertifikat importieren, füllen Sie das Formular aus und wählen Sie Importieren aus.
8. Wählen Sie Ausstehende Zertifikate hinzufügen aus.

Um der Zertifikatsliste ein Zertifikat hinzuzufügen, verwenden Sie AWS CLI

Verwenden Sie den Befehl [add-listener-certificates](#).

Entfernen eines Zertifikats aus der Zertifikatsliste

Sie können mit den folgenden Schritten Zertifikate aus der Zertifikatsliste für einen HTTPS-Listener entfernen. Informationen zum Entfernen des Standardzertifikats für einen HTTPS-Listener finden Sie unter [Ersetzen des Standardzertifikats](#).

So entfernen Sie Zertifikate über die Konsole aus der Zertifikatsliste

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Wählen Sie auf der Registerkarte Listener und Regeln den Text in der Spalte Protokoll: Port aus, um die Detailseite für den Listener zu öffnen.
5. Aktivieren Sie auf der Registerkarte Zertifikate die Kontrollkästchen für die Zertifikate und wählen Sie Entfernen aus.
6. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **confirm** ein und wählen Sie dann Entfernen.

Um ein Zertifikat aus der Zertifikatsliste zu entfernen, verwenden Sie AWS CLI

Verwenden Sie den Befehl [remove-listener-certificates](#).

Aktualisieren der Sicherheitsrichtlinie

Wenn Sie einen HTTPS-Listener erstellen, können Sie die Sicherheitsrichtlinie auswählen, die Ihre Anforderungen erfüllt. Wenn eine neue Sicherheitsrichtlinie hinzugefügt wird, können Sie Ihren HTTPS-Listener aktualisieren, sodass die neue Sicherheitsrichtlinie verwendet wird. Application Load Balancer unterstützen keine benutzerdefinierten Sicherheitsrichtlinien. Weitere Informationen finden Sie unter [Sicherheitsrichtlinien](#).

Verwenden von FIPS-Richtlinien auf Ihrem Application Load Balancer:

Alle sicheren Listener, die an einen Application Load Balancer angeschlossen sind, müssen entweder FIPS-Sicherheitsrichtlinien oder Nicht-FIPS-Sicherheitsrichtlinien verwenden; sie können nicht gemischt werden. Wenn ein vorhandener Application Load Balancer über zwei oder mehr Listener

verfügt, die Nicht-FIPS-Richtlinien verwenden, und Sie möchten, dass die Listener stattdessen FIPS-Sicherheitsrichtlinien verwenden, entfernen Sie alle Listener, bis nur noch einer vorhanden ist. Ändern Sie die Sicherheitsrichtlinie des Listeners in FIPS und erstellen Sie dann weitere Listener mithilfe von FIPS-Sicherheitsrichtlinien. Alternativ können Sie einen neuen Application Load Balancer mit neuen Listenern erstellen, die nur FIPS-Sicherheitsrichtlinien verwenden.

Aktualisieren der Sicherheitsrichtlinie mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Wählen Sie auf der Registerkarte Listener und Regeln den Text in der Spalte Protokoll: Port aus, um die Detailseite für den Listener zu öffnen.
5. Wählen Sie dann auf der Seite Details zuerst Aktionen und dann Listener bearbeiten aus.
6. Wählen Sie im Abschnitt Einstellungen für sichere Listener unter Sicherheitsrichtlinie eine neue Sicherheitsrichtlinie aus.
7. Wählen Sie Änderungen speichern aus.

Um die Sicherheitsrichtlinie zu aktualisieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-listener](#).

Gegenseitige Authentifizierung mit TLS im Application Load Balancer

Die gegenseitige TLS-Authentifizierung ist eine Variante von Transport Layer Security (TLS). Herkömmliches TLS ermöglicht eine sichere Kommunikation zwischen einem Server und einem Client, wobei der Server seinen Clients seine Identität mitteilen muss. Bei Mutual TLS handelt es sich um einen Load Balancer, bei dem die Aushandlung von TLS die gegenseitige Authentifizierung zwischen dem Client und dem Server aus. Wenn Sie Mutual TLS mit Application Load Balancer verwenden, vereinfachen Sie das Authentifizierungsmanagement und reduzieren die Belastung Ihrer Anwendungen.

Durch die Verwendung von Mutual TLS mit Application Load Balancer kann Ihr Load Balancer die Client-Authentifizierung verwalten, um sicherzustellen, dass nur vertrauenswürdige Clients mit Ihren Backend-Anwendungen kommunizieren. Wenn Sie diese Funktion verwenden, authentifiziert Application Load Balancer Clients mit Zertifikaten von einer Zertifizierungsstelle (CA) eines

Drittanbieters oder mithilfe der AWS Private Certificate Authority (PCA), optional, mit Sperrprüfungen. Application Load Balancer leitet Client-Zertifikatsinformationen an das Backend weiter, die Ihre Anwendungen für die Autorisierung verwenden können. Durch die Verwendung von Mutual TLS in Application Load Balancer erhalten Sie eine integrierte, skalierbare, verwaltete Authentifizierung für zertifikatsbasierte Entitäten, die etablierte Bibliotheken verwendet.

Mutual TLS for Application Load Balancers bietet die folgenden zwei Optionen für die Validierung Ihrer X.509v3-Client-Zertifikate:

Hinweis: X.509v1-Clientzertifikate werden nicht unterstützt.

- **Gegenseitiger TLS-Passthrough:** Wenn Sie den Mutual TLS-Passthrough-Modus verwenden, sendet Application Load Balancer die gesamte Client-Zertifikatskette mithilfe von HTTP-Headern an das Ziel. Mithilfe der Client-Zertifikatskette können Sie dann die entsprechende Authentifizierungs- und Autorisierungslogik in Ihrer Anwendung implementieren.
- **Gegenseitige TLS-Überprüfung:** Wenn Sie den Modus für die gegenseitige TLS-Überprüfung verwenden, führt Application Load Balancer die X.509-Client-Zertifikatsauthentifizierung für Clients durch, wenn ein Load Balancer TLS-Verbindungen aushandelt.

Um mit Mutual TLS in Application Load Balancer mithilfe von Passthrough zu beginnen, müssen Sie den Listener nur so konfigurieren, dass er Zertifikate von Clients akzeptiert. Um Mutual TLS mit Verifizierung zu verwenden, müssen Sie wie folgt vorgehen:

- Erstellen Sie eine neue Trust Store-Ressource.
- Laden Sie Ihr Zertifizierungsstellenpaket (CA) und optional Sperrlisten hoch.
- Hängen Sie den Trust Store an den Listener an, der für die Überprüfung von Client-Zertifikaten konfiguriert ist.

step-by-step Verfahren zur Konfiguration des gegenseitigen TLS-Überprüfungsmodus mit Ihrem Application Load Balancer finden Sie unter [Konfiguration von Mutual TLS auf einem Application Load Balancer](#).

Bevor Sie mit der Konfiguration von Mutual TLS auf Ihrem Application Load Balancer beginnen

Bevor Sie mit der Konfiguration von Mutual TLS auf Ihrem Application Load Balancer beginnen, sollten Sie Folgendes beachten:

Kontingente

Application Load Balancers enthalten bestimmte Beschränkungen, die sich auf die Anzahl der in Ihrem AWS Konto verwendeten Trust Stores, CA-Zertifikate und Zertifikatssperllisten beziehen.

Weitere Informationen finden Sie unter [Kontingente für Ihre Application Load Balancers](#).

Anforderungen für Zertifikate

Application Load Balancers unterstützen Folgendes für Zertifikate, die mit gegenseitiger TLS-Authentifizierung verwendet werden:

- Unterstütztes Zertifikat: X.509v3
- Unterstützte öffentliche Schlüssel: RSA 2K — 8K oder ECDSA secp256r1, secp384r1, secp521r1
- Unterstützte Signaturalgorithmen: SHA256, 384, 512 mit RSA/SHA256, 384, 512 mit EC/SHA256,384,512 Hash mit RSASSA-PSS mit MGF1

CA-Zertifikatspakete

Folgendes gilt für Zertifizierungsstellen-Pakete (CA):

- Application Load Balancers laden jedes Zertifikatspaket der Zertifizierungsstelle (CA) als Batch hoch. Application Load Balancers unterstützen das Hochladen einzelner Zertifikate nicht. Wenn Sie neue Zertifikate hinzufügen müssen, müssen Sie die Zertifikatspaketdatei hochladen.
- Verwenden Sie die [ModifyTrustStore-API](#), um ein CA-Zertifikatspaket zu ersetzen.

Reihenfolge der Zertifikate für Passthrough

Wenn Sie den gegenseitigen TLS-Passthrough verwenden, fügt der Application Load Balancer Header ein, um den Backend-Zielen die Zertifikatskette des Clients zu präsentieren. Die Reihenfolge der Präsentation beginnt mit den Leaf-Zertifikaten und endet mit dem Stammzertifikat.

Wiederaufnahme der Sitzung

Die Sitzungswiederaufnahme wird nicht unterstützt, wenn der Modus Mutual TLS Passthrough oder Verify mit einem Application Load Balancer verwendet wird.

HTTP-Header

Application Load Balancers verwenden X-Amzn-Mtls Header, um Zertifikatsinformationen zu senden, wenn es Clientverbindungen mit gegenseitigem TLS aushandelt. Weitere Informationen und Beispiel-Header finden Sie unter [HTTP-Header und gegenseitiges TLS](#)

CA-Zertifikatsdateien

CA-Zertifikatsdateien müssen die folgenden Anforderungen erfüllen:

- Die Zertifikatsdatei muss das PEM-Format (Privacy Enhanced Mail) verwenden.
- Der Inhalt des Zertifikats muss innerhalb der -----END CERTIFICATE----- Grenzen -----BEGIN CERTIFICATE----- und liegen.
- Den Kommentaren muss ein # Zeichen vorangestellt werden.
- Es dürfen keine Leerzeilen vorhanden sein.

Beispielzertifikat, das nicht akzeptiert wird (ungültig):

```
# comments

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 01
  Signature Algorithm: ecdsa-with-SHA384
  Issuer: C=US, O=EXAMPLE, OU=EXAMPLE, CN=EXAMPLE
  Validity
    Not Before: Jan 11 23:57:57 2024 GMT
    Not After : Jan 10 00:57:57 2029 GMT
  Subject: C=US, O=EXAMPLE, OU=EXAMPLE, CN=EXAMPLE
  Subject Public Key Info:
    Public Key Algorithm: id-ecPublicKey
    Public-Key: (384 bit)
    pub:
      00:01:02:03:04:05:06:07:08
    ASN1 OID: secp384r1
    NIST CURVE: P-384
  X509v3 extensions:
    X509v3 Key Usage: critical
      Digital Signature, Key Encipherment, Certificate Sign, CRL Sign
    X509v3 Basic Constraints: critical
      CA:TRUE
    X509v3 Subject Key Identifier:
      00:01:02:03:04:05:06:07:08
    X509v3 Subject Alternative Name:
      URI:EXAMPLE.COM
  Signature Algorithm: ecdsa-with-SHA384
    00:01:02:03:04:05:06:07:08
```

```
-----BEGIN CERTIFICATE-----  
Base64-encoded certificate  
-----END CERTIFICATE-----
```

Beispielzertifikate, die akzeptiert werden (gültig):

1. Einzelzertifikat (PEM-kodiert):

```
# comments  
-----BEGIN CERTIFICATE-----  
Base64-encoded certificate  
-----END CERTIFICATE-----
```

2. Mehrere Zertifikate (PEM-kodiert):

```
# comments  
-----BEGIN CERTIFICATE-----  
Base64-encoded certificate  
-----END CERTIFICATE-----  
# comments  
-----BEGIN CERTIFICATE-----  
Base64-encoded certificate  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
Base64-encoded certificate  
-----END CERTIFICATE-----
```

HTTP-Header und gegenseitiges TLS

In diesem Abschnitt werden die HTTP-Header beschrieben, die Application Load Balancers verwenden, um Zertifikatsinformationen zu senden, wenn Verbindungen mit Clients, die gegenseitiges TLS verwenden, aushandeln. Die spezifischen X-Amzn-Mtls Header, die der Application Load Balancer verwendet, hängen vom von Ihnen angegebenen gegenseitigen TLS-Modus ab: Passthrough-Modus oder Verifizierungsmodus.

Hinweise zu anderen HTTP-Headern, die von Application Load Balancers unterstützt werden, finden Sie unter [HTTP-Header und Application Load Balancer](#)

HTTP-Header für den Passthrough-Modus

Für Mutual TLS im Passthrough-Modus verwenden Application Load Balancer den folgenden Header.

X-Amzn-Mtls-Clientcert

Dieser Header enthält das URL-kodierte PEM-Format der gesamten Client-Zertifikatskette, die in der Verbindung dargestellt wird, mit sicheren Zeichen. +=/

Inhalt eines Beispiel-Headers:

```
X-Amzn-Mtls-Clientcert: -----BEGIN%20CERTIFICATE-----%0AMIID<...reduced...>do0g%3D%3D%0A-----END%20CERTIFICATE-----%0A-----BEGIN%20CERTIFICATE-----%0AMIID1<...reduced...>3eZlyKA%3D%3D%0A-----END%20CERTIFICATE-----%0A
```

HTTP-Header für den Überprüfungsmodus

Für gegenseitiges TLS im Überprüfungsmodus verwenden Application Load Balancer die folgenden Header.

X-Amzn-Mtls-Clientcert-Seriennummer

Dieser Header enthält eine hexadezimale Darstellung der Seriennummer des Leaf-Zertifikats.

Beispiel für den Inhalt der Kopfzeile:

```
X-Amzn-Mtls-Clientcert-Serial-Number: 03A5B1
```

X-Amzn-Mtls-Clientcert-Issuer

Dieser Header enthält eine RFC2253-String-Darstellung des definierten Namens (DN) des Emittenten.

Beispiel für einen Header-Inhalt:

```
X-Amzn-Mtls-Clientcert-Issuer:  
CN=rootcamtls.com,OU=rootCA,O=mTLS,L=Seattle,ST=Washington,C=US
```

X-Amzn-Mtls-Clientcert-Betreff

Dieser Header enthält eine RFC2253-String-Darstellung des definierten Namens (DN) des Subjekts.

Beispiel für den Inhalt einer Kopfzeile:

```
X-Amzn-Mtls-Clientcert-Subject: CN=client_.com,OU=client-3,O=mTLS,ST=Washington,C=US
```

X-Amzn-Mtls-Clientcert-Validität

Dieser Header enthält das Datum und im ISO8601-Format. `notBefore` `notAfter`

Beispiel für den Inhalt einer Kopfzeile:

```
X-Amzn-Mtls-Clientcert-Validity:  
NotBefore=2023-09-21T01:50:17Z;NotAfter=2024-09-20T01:50:17Z
```

X-Amzn-Mtls-Clientcert-Leaf

Dieser Header enthält ein URL-codiertes PEM-Format des Leaf-Zertifikats mit sicheren Zeichen. `+/=`

Beispiel für den Inhalt einer Kopfzeile:

```
X-Amzn-Mtls-Clientcert-Leaf: -----BEGIN%20CERTIFICATE-----%0AMIIG<...reduced...>NmULw  
%0A-----END%20CERTIFICATE-----%0A
```

Konfiguration von Mutual TLS auf einem Application Load Balancer

Dieser Abschnitt enthält die Verfahren zur Konfiguration des Modus für die gegenseitige TLS-Überprüfung für die Authentifizierung auf Application Load Balancern.

Um den Mutual TLS-Passthrough-Modus zu verwenden, müssen Sie den Listener nur so konfigurieren, dass er Zertifikate von Clients akzeptiert. Wenn Sie gegenseitiges TLS-Passthrough verwenden, sendet der Application Load Balancer die gesamte Client-Zertifikatskette mithilfe von HTTP-Headern an das Ziel, sodass Sie die entsprechende Authentifizierungs- und Autorisierungslogik in Ihrer Anwendung implementieren können. Weitere Informationen finden Sie unter [Einen HTTPS-Listener für Ihren Application Load Balancer erstellen](#).

Wenn Sie Mutual TLS im Überprüfungsmodus verwenden, führt der Application Load Balancer eine X.509-Client-Zertifikatsauthentifizierung für Clients durch, wenn ein Load Balancer TLS-Verbindungen aushandelt.

Gehen Sie wie folgt vor, um den Modus für die gegenseitige TLS-Überprüfung zu verwenden:

- Erstellen Sie eine neue Trust Store-Ressource.
- Laden Sie Ihr Zertifizierungsstellenpaket (CA) und optional Sperrlisten hoch.
- Hängen Sie den Trust Store an den Listener an, der für die Überprüfung von Client-Zertifikaten konfiguriert ist.

Folgen Sie den Verfahren in diesem Abschnitt, um den Modus für die gegenseitige TLS-Überprüfung auf Ihrem Application Load Balancer in der AWS Management Console zu konfigurieren.

Informationen zur Konfiguration von Mutual TLS mithilfe von API-Vorgängen anstelle der Konsole finden Sie im [Application Load Balancer API-Referenzhandbuch](#).

Aufgaben

- [Erstellen Sie einen Trust Store](#)
- [Ordnen Sie einen Trust Store zu](#)
- [Details zum Trust Store anzeigen](#)
- [Ändern Sie einen Trust Store](#)
- [Löschen Sie einen Trust Store](#)

Erstellen Sie einen Trust Store

Es gibt drei Möglichkeiten, einen Trust Store zu erstellen: wenn Sie einen Application Load Balancer erstellen, wenn Sie einen sicheren Listener erstellen und indem Sie die Trust Store-Konsole verwenden. Wenn Sie bei der Erstellung eines Load Balancers oder Listeners einen Trust Store hinzufügen, wird der Trust Store automatisch dem neuen Listener zugeordnet. Wenn Sie mithilfe der Trust Store-Konsole einen Trust Store erstellen, müssen Sie ihn selbst einem Listener zuordnen.

In diesem Abschnitt wird das Erstellen eines Trust Stores mithilfe der Trust Store-Konsole behandelt. Die Schritte beim Erstellen eines Application Load Balancer oder Listeners sind jedoch dieselben. Weitere Informationen finden [Sie unter Einen Load Balancer und einen Listener konfigurieren und Einen HTTPS-Listener hinzufügen](#).

Voraussetzungen:

- Um einen Trust Store zu erstellen, benötigen Sie ein Zertifikatspaket von Ihrer Zertifizierungsstelle (CA).

Um einen Trust Store mit der Konsole zu erstellen

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie Trust Store erstellen aus.
4. Konfiguration des Vertrauensspeichers

- a. Geben Sie unter Trust Store Name einen Namen für Ihren Trust Store ein.
- b. Geben Sie für Certificate Authority Bundle den Amazon S3 S3-Pfad zu dem CA-Zertifikatspaket ein, das Ihr Trust Store verwenden soll.

Optional: Verwenden Sie die Objektversion, um eine frühere Version des CA-Zertifikatspakets auszuwählen. Andernfalls wird die aktuelle Version verwendet.

5. Für Widerrufe können Sie optional eine Zertifikatssperrliste zu Ihrem Trust Store hinzufügen.
 - Geben Sie unter Certificate Revocation List den Amazon S3 S3-Pfad zu der Zertifikatssperrliste ein, die Ihr Trust Store verwenden soll.

Optional: Verwenden Sie Objektversion, um eine frühere Version der Zertifikatssperrliste auszuwählen. Andernfalls wird die aktuelle Version verwendet.

6. Für Trust Store-Tags können Sie optional bis zu 50 Tags eingeben, die auf Ihren Trust Store angewendet werden sollen.
7. Wählen Sie Trust Store erstellen aus.

Ordnen Sie einen Trust Store zu

Nachdem Sie einen Trust Store erstellt haben, müssen Sie ihn einem Listener zuordnen, bevor Ihr Application Load Balancer den Trust Store verwenden kann. Sie können jedem Ihrer sicheren Listener nur einen Trust Store zuordnen, aber ein Trust Store kann mehreren Listenern zugeordnet werden.

In diesem Abschnitt wird das Zuordnen eines Vertrauensspeichers zu einem vorhandenen Listener beschrieben. Alternativ können Sie beim Erstellen eines Application Load Balancer oder Listeners einen Trust Store zuordnen. Weitere Informationen finden [Sie unter Einen Load Balancer und einen Listener konfigurieren und Einen HTTPS-Listener hinzufügen](#).

So ordnen Sie mithilfe der Konsole einen Trust Store zu

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus, um seine Detailseite aufzurufen.
4. Wählen Sie auf der Registerkarte Listener und Regeln den Link in der Spalte Protokoll:Port, um die Detailseite für den sicheren Listener zu öffnen.

5. Wählen Sie auf der Registerkarte Sicherheit die Option Einstellungen für sicheren Listener bearbeiten aus.
6. (Optional) Wenn Mutual TLS nicht aktiviert ist, wählen Sie Mutual Authentication (mTLS) unter Behandlung von Client-Zertifikaten aus und wählen Sie dann Verify with Trust Store aus.
7. Wählen Sie unter Trust Store den Trust Store aus, den Sie erstellt haben.
8. Wählen Sie Änderungen speichern aus.

Details zum Trust Store anzeigen

CA-Zertifikatspakete

Das CA-Zertifikatspaket ist eine erforderliche Komponente des Trust Store. Es handelt sich um eine Sammlung vertrauenswürdiger Stamm- und Zwischenzertifikate, die von einer Zertifizierungsstelle validiert wurden. Diese validierten Zertifikate stellen sicher, dass der Client darauf vertrauen kann, dass das vorgelegte Zertifikat dem Load Balancer gehört.

Sie können den Inhalt des aktuellen CA-Zertifikatspakets jederzeit in Ihrem Trust Store einsehen.

Ein CA-Zertifikatspaket anzeigen

So zeigen Sie ein CA-Zertifikatspaket mit der Konsole an

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie den Trust Store aus, um die Detailseite anzuzeigen.
4. Wählen Sie Actions und dann Get CA Bundle aus.
5. Wählen Sie Link teilen oder Herunterladen.

Listen zum Widerruf von Zertifikaten

Optional können Sie eine Zertifikatssperrliste für einen Vertrauensspeicher erstellen. Sperrlisten werden von Zertifizierungsstellen veröffentlicht und enthalten Daten für Zertifikate, die gesperrt wurden. Application Load Balancers unterstützen nur Zertifikatssperrlisten im PEM-Format.

Wenn eine Zertifikatssperrliste zu einem Vertrauensspeicher hinzugefügt wird, erhält sie eine Sperr-ID. Die Sperr-IDs erhöhen sich mit jeder Sperrliste, die dem Trust Store hinzugefügt wird, und sie können nicht geändert werden. Wenn eine Zertifikatssperrliste aus einem Vertrauensspeicher gelöscht

wird, wird auch ihre Sperr-ID gelöscht und für die gesamte Lebensdauer des Vertrauensspeichers nicht wiederverwendet.

Note

Application Load Balancers können in einer Zertifikatssperrliste keine Zertifikate mit einer negativen Seriennummer widerrufen.

Eine Zertifikatssperrliste anzeigen

So zeigen Sie eine Sperrliste mit der Konsole an

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie den Trust Store aus, um die Detailseite anzuzeigen.
4. Wählen Sie auf der Registerkarte Zertifikatssperrlisten die Optionen Aktionen und dann Sperrliste abrufen aus.
5. Wählen Sie Link teilen oder Herunterladen aus.

Ändern Sie einen Trust Store

Ein Trust Store kann jeweils nur ein CA-Zertifikatpaket enthalten, aber Sie können das CA-Zertifikatpaket jederzeit ersetzen, nachdem der Trust Store erstellt wurde.

Ersetzen Sie ein CA-Zertifikatpaket

Um ein CA-Zertifikatpaket mithilfe der Konsole zu ersetzen

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie den Trust Store aus, um die Detailseite anzuzeigen.
4. Wählen Sie „Aktionen“ und dann „CA-Bundle ersetzen“.
5. Geben Sie auf der Seite CA-Bundle ersetzen unter Certificate Authority Bundle den Amazon S3 S3-Standort des gewünschten CA-Bundles ein.
6. (Optional) Verwenden Sie Objektversion, um eine frühere Version der Zertifikatssperrliste auszuwählen. Andernfalls wird die aktuelle Version verwendet.

7. Wählen Sie CA-Bundle ersetzen aus.

Fügen Sie eine Sperrliste für Zertifikate hinzu

Um mit der Konsole eine Sperrliste hinzuzufügen

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie den Trust Store aus, um seine Detailseite anzuzeigen.
4. Wählen Sie auf der Registerkarte Zertifikatssperrlisten die Option Aktionen und dann Sperrliste hinzufügen aus.
5. Geben Sie auf der Seite Sperrliste hinzufügen unter Zertifikatssperrliste den Amazon S3 S3-Speicherort der gewünschten Zertifikatssperrliste ein.
6. (Optional) Verwenden Sie Objektversion, um eine frühere Version der Zertifikatssperrliste auszuwählen. Andernfalls wird die aktuelle Version verwendet.
7. Wählen Sie Sperrliste hinzufügen

Löschen Sie eine Zertifikatssperrliste

Um eine Sperrliste mit der Konsole zu löschen

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie den Trust Store aus, um die Detailseite anzuzeigen.
4. Wählen Sie auf der Registerkarte Zertifikatssperrlisten die Optionen Aktionen und dann Sperrliste löschen aus.
5. Bestätigen Sie den Löschvorgang, indem Sie Folgendes eingeben `confirm`.
6. Wählen Sie Löschen aus.

Löschen Sie einen Trust Store

Wenn Sie einen Trust Store nicht mehr benötigen, können Sie ihn löschen.

Hinweis: Sie können keinen Trust Store löschen, der derzeit einem Listener zugeordnet ist.

Um einen Trust Store mit der Konsole zu löschen

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Trust Stores aus.
3. Wählen Sie den Trust Store aus, um seine Detailseite anzuzeigen.
4. Wählen Sie Aktionen und dann Trust Store löschen aus.
5. Bestätigen Sie den Löschvorgang, indem Sie Folgendes eingeben `confirm`.
6. Wählen Sie Löschen

Verbindungsprotokolle für Application Load Balancer

Elastic Load Balancing stellt Verbindungsprotokolle bereit, die Attribute der an Ihre Application Load Balancer gesendeten Anfragen erfassen. Verbindungsprotokolle enthalten Informationen wie die Client-IP-Adresse und den Port, Informationen zum Client-Zertifikat, die Verbindungsergebnisse und die verwendeten TLS-Chiffren. Diese Verbindungsprotokolle können dann verwendet werden, um Anforderungsmuster und andere Trends zu überprüfen.

Weitere Informationen zu Verbindungsprotokollen finden Sie unter [Verbindungsprotokolle für Ihren Application Load Balancer](#)

Authentifizieren von Benutzern mithilfe eines Application Load Balancers

Sie können einen Application Load Balancer so konfigurieren, dass Benutzer auf sichere Weise authentifiziert werden, wenn sie auf ihre Anwendungen zugreifen. So können Sie den Aufwand für das Authentifizieren der Benutzer auf Ihren Load Balancer verlagern, damit sich Ihre Anwendungen auf die Geschäftslogik konzentrieren können.

Die folgenden Anwendungsfälle werden unterstützt:

- Authentifizieren von Benutzern über einen Identitätsanbieter (IdP), der mit OpenID Connect (OIDC) konform ist
- Authentifizieren Sie Benutzer über soziale Netzwerke IdPs wie Amazon oder Google über die von Amazon Cognito unterstützten Benutzerpools. Facebook
- Authentifizieren Sie Benutzer über Unternehmensidentitäten mithilfe von SAML, OpenID Connect (OIDC) oder OAuth über die von Amazon Cognito unterstützten Benutzerpools.

Vorbereiten der Nutzung eines OIDC-konformen Identitätsanbieters

Gehen Sie wie folgt vor, wenn Sie für Ihren Application Load Balancer einen OIDC-konformen Identitätsanbieter verwenden:

- Erstellen Sie unter Ihrem Identitätsanbieter eine neue OIDC-App. Das DNS des Identitätsanbieters muss öffentlich auflösbar sein.
- Sie müssen eine Client-ID und einen Clientschlüssel konfigurieren.
- Rufen Sie die folgenden Arten von Endpunkten ab, die vom Identitätsanbieter veröffentlicht werden: Autorisierung, Token und Benutzerinformationen. Sie finden diese Informationen in der Konfiguration.
- Die Identitätsanbieter-Endpunktzertifikate sollten von einer vertrauenswürdigen öffentlichen Zertifizierungsstelle ausgestellt werden.
- Die DNS-Datensätze für die Endpunkte müssen öffentlich auflösbar sein, auch wenn sie in private IP-Adressen aufgelöst werden.
- Erlauben Sie je nach Nutzung durch Ihre Benutzer eine der folgenden Umleitungs-URLs in Ihrer Identitätsanbieter-App. Hierbei ist DNS der Domainname Ihres Load Balancers und CNAME der DNS-Alias für Ihre Anwendung:
 - <https://DNS/oauth2/idpresponse>
 - <https://CNAME/oauth2/idpresponse>

Vorbereitung für die Verwendung von Amazon Cognito

Verfügbare Regionen

Die Amazon Cognito Cognito-Integration für Application Load Balancers ist in den folgenden Regionen verfügbar:

- USA Ost (Nord-Virginia)
- USA Ost (Ohio)
- USA West (Nordkalifornien)
- USA West (Oregon)
- Kanada (Zentral)
- Europa (Stockholm)
- Europa (Milan)

- Europa (Frankfurt)
- Europa (Zürich)
- Europa (Irland)
- Europe (London)
- Europe (Paris)
- Südamerika (São Paulo)
- Asien-Pazifik (Tokio)
- Asien-Pazifik (Seoul)
- Asien-Pazifik (Osaka)
- Asia Pacific (Mumbai)
- Asien-Pazifik (Singapur)
- Asien-Pazifik (Sydney)
- Asien-Pazifik (Jakarta)
- Naher Osten (VAE)
- Naher Osten (Bahrain)
- Afrika (Kapstadt)
- Israel (Tel Aviv)

Gehen Sie wie folgt vor, wenn Sie Amazon-Cognito-Benutzerpools für Ihren Application Load Balancer verwenden:

- Erstellen Sie einen Benutzerpool. Weitere Informationen finden Sie unter [Amazon-Cognito-Benutzerpools](#) im Amazon-Cognito-Entwicklerhandbuch.
- Erstellen Sie einen Benutzerpool-Client. Sie müssen den Client so konfigurieren, dass ein Clientschlüssel generiert wird, der Ablauf zur Code-Erteilung verwendet wird und die gleichen OAuth-Bereiche wie für den Load Balancer unterstützt werden. Weitere Informationen finden Sie unter [Konfigurieren eines Benutzerpool-App-Clients](#) im Amazon-Cognito-Entwicklerhandbuch.
- Erstellen Sie eine Benutzerpool-Domain. Weitere Informationen finden Sie unter [Hinzufügen eines Domainnamens für Ihren Benutzerpool](#) im Amazon-Cognito-Entwicklerhandbuch.
- Überprüfen Sie, ob der angeforderte Bereich ein ID-Token zurückgibt.
Beispiel: Der Standardbereich `openid` gibt ein ID-Token zurück, der Bereich `aws.cognito.signin.user.admin` hingegen nicht.

Hinweis: Application Load Balancers unterstützen keine von Amazon Cognito ausgegebenen benutzerdefinierten Zugriffstoken. Weitere Informationen finden Sie unter [Pre-Token-Generierung](#) im Amazon Cognito Developer Guide.

- Aktivieren Sie den Identitätsanbieter im Verbundabschnitt, um einen Verbund mit einem Anbieter von Social Identities bzw. Unternehmensidentitäten einzurichten. Weitere Informationen finden Sie unter [Hinzufügen von Social Sign-in zu einem Benutzerpool](#) oder [Hinzufügen der Anmeldung mit einem SAML-Identitätsanbieter zu einem Benutzerpool](#) im Amazon-Cognito-Entwicklerhandbuch.
- Erlauben Sie die folgenden Umleitungs-URLs im Feld für die Rückruf-URL für Amazon Cognito. Hierbei ist DNS der Domainname Ihres Load Balancers und CNAME der DNS-Alias für Ihre Anwendung (falls zutreffend):
 - <https://DNS/oauth2/idpresponse>
 - <https://CNAME/oauth2/idpresponse>
- Erlauben Sie Ihre Benutzerpool-Domain für die Rückruf-URL Ihrer Identitätsanbieter-App. Verwenden Sie das für Ihren Identitätsanbieter erforderliche Format. Beispielsweise:
 - <https://domain-prefix.auth.region.amazoncognito.com/saml2/idpresponse>
 - <https://user-pool-domain/oauth2/idpresponse>

Die Callback-URL in den App-Client-Einstellungen darf ausschließlich Kleinbuchstaben enthalten.

Damit ein Benutzer einen Load Balancer für die Verwendung von Amazon Cognito zum Authentifizieren von Benutzern konfigurieren kann, müssen Sie dem Benutzer die Berechtigung zum Aufrufen der Aktion `cognito-idp:DescribeUserPoolClient` gewähren.

Bereiten Sie sich auf die Nutzung von Amazon vor CloudFront

Aktivieren Sie die folgenden Einstellungen, wenn Sie eine CloudFront Distribution vor Ihrem Application Load Balancer verwenden:

- Anforderungsheader weiterleiten (alle) — Stellt sicher, dass Antworten für authentifizierte Anfragen CloudFront nicht zwischengespeichert werden. Damit wird vermieden, dass sie nach Ablauf der Authentifizierungssitzung aus dem Zwischenspeicher geladen werden. Um dieses Risiko zu verringern, wenn das Caching aktiviert ist, können Besitzer einer CloudFront Distribution alternativ festlegen, dass der time-to-live (TTL-) Wert abläuft, bevor das Authentifizierungscookie abläuft.

- Abfragezeichenfolge weiterleiten und zwischenspeichern (alle) – Stellt sicher, dass der Load Balancer Zugriff auf die Abfragezeichenfolgenparameter hat, die für die Authentifizierung des Benutzers beim IdP erforderlich sind.
- Cookie-Weiterleitung (alle) — Stellt sicher, dass alle Authentifizierungs-Cookies an den Load Balancer CloudFront weitergeleitet werden.

Konfigurieren der Benutzerauthentifizierung

Sie konfigurieren die Benutzerauthentifizierung, indem Sie eine Authentifizierungsaktion für eine oder mehrere Listener-Regeln erstellen. Die Aktionstypen `authenticate-cognito` und `authenticate-oidc` werden nur mit HTTPS-Listnern unterstützt. Eine Beschreibung der entsprechenden Felder finden Sie unter [AuthenticateCognitoActionConfig](#) und [AuthenticateOidcActionConfig](#) in der Elastic Load Balancing API-Referenzversion 2015-12-01.

Der Load Balancer sendet ein Session-Cookie an den Client, um den Authentifizierungsstatus beizubehalten. Dieses Cookie enthält immer das Attribut `secure`, da die Benutzerauthentifizierung einen HTTPS-Listener erfordert. Dieses Cookie enthält das Attribut `SameSite=None` mit CORS-Anforderungen (Cross-Origin Resource Sharing).

Für einen Load Balancer, der mehrere Anwendungen unterstützt, die eine unabhängige Client-Authentifizierung verlangen, sollte jede Listener-Regel mit einer Authentifizierungsaktion einen eindeutigen Cookie-Namen haben. Dadurch wird sichergestellt, dass Clients immer beim IdP authentifiziert werden, bevor sie an die in der Regel angegebene Zielgruppe weitergeleitet werden.

Application Load Balancer unterstützen keine Cookie-Werte, die URL-codiert sind.

Das Feld `SessionTimeout` ist standardmäßig auf 7 Tage festgelegt. Wenn Sie kürzere Sitzungen benötigen, können Sie auch eine Sitzungs-Zeitbeschränkung von bis zu lediglich einer Sekunde konfigurieren. Weitere Informationen finden Sie unter [Sitzungs-Timeout](#).

Legen Sie das Feld `OnUnauthenticatedRequest` je nach Bedarf für Ihre Anwendung fest. Beispielsweise:

- Anwendungen, bei denen sich der Benutzer mit einer Social Identity oder Unternehmensidentität anmelden muss: Dies wird von der Standardoption `authenticate` unterstützt. Wenn der Benutzer nicht angemeldet ist, leitet der Load Balancer die Anforderung an den Identitätsanbieter-Autorisierungsendpunkt weiter. Der Benutzer wird dann vom Identitätsanbieter aufgefordert, sich über die entsprechende Benutzeroberfläche anzumelden.

- Anwendungen mit einer personalisierten Ansicht für angemeldete Benutzer und einer allgemeinen Ansicht für nicht angemeldete Benutzer: Verwenden Sie die Option `allow`, um diese Art von Anwendung zu unterstützen. Wenn der Benutzer angemeldet ist, stellt der Load Balancer die Benutzeransprüche bereit und die Anwendung kann eine personalisierte Ansicht anzeigen. Wenn der Benutzer nicht angemeldet ist, leitet der Load Balancer die Anforderung ohne die Benutzeransprüche weiter und die Anwendung kann die allgemeine Ansicht anzeigen.
- Einseitige Anwendungen JavaScript, die alle paar Sekunden geladen werden — Wenn Sie die `deny` Option verwenden, gibt der Load Balancer bei AJAX-Aufrufen, die keine Authentifizierungsinformationen enthalten, den Fehler HTTP 401 Unauthorized zurück. Wenn die Authentifizierungsinformationen des Benutzers jedoch abgelaufen sind, wird der Client zum IdP-Autorisierungsendpunkt weitergeleitet.

Der Load Balancer muss mit dem Identitätsanbieter-Tokenendpunkt (`TokenEndpoint`) und dem Endpunkt mit den Benutzerinformationen (`UserInfoEndpoint`) des Identitätsanbieters kommunizieren können. Application Load Balancer unterstützen IPv4 nur bei der Kommunikation mit diesen Endpunkten. Wenn Ihr IdP öffentliche Adressen verwendet, stellen Sie sicher, dass die Sicherheitsgruppen für Ihren Load Balancer und die Netzwerk-ACLs für Ihre VPC den Zugriff auf die Endpunkte zulassen. Wenn Sie einen internen Load Balancer oder den IP-Adresstyp `use-elastic-load-balancing-without-public-ipv4` verwendendualstack-without-public-ipv4, kann ein NAT-Gateway dem Load Balancer die Kommunikation mit den Endpunkten ermöglichen. Weitere Informationen finden Sie unter [Grundlagen zu NAT-Gateways](#) im Amazon-VPC-Benutzerhandbuch.

Verwenden Sie den folgenden [create-rule](#)-Befehl, um die Benutzerauthentifizierung zu konfigurieren.

```
aws elbv2 create-rule --listener-arn listener-arn --priority 10 \  
--conditions Field=path-pattern,Values="/login" --actions file://actions.json
```

Im Folgenden finden Sie ein Beispiel für die `actions.json`-Datei, die eine `authenticate-oidc`-Aktion und eine `forward`-Aktion angibt. `AuthenticationRequestExtraParams` ermöglicht es Ihnen, während der Authentifizierung zusätzliche Parameter an einen IdP zu übergeben. In der Dokumentation Ihres Identitätsanbieters ist angegeben, welche Felder unterstützt werden.

```
[{  
  "Type": "authenticate-oidc",  
  "AuthenticateOidcConfig": {  
    "Issuer": "https://idp-issuer.com",  
    "AuthorizationEndpoint": "https://authorization-endpoint.com",  
    "TokenEndpoint": "https://token-endpoint.com",
```

```

    "UserInfoEndpoint": "https://user-info-endpoint.com",
    "ClientId": "abcdefghijklmnopqrstuvxyz123456789",
    "ClientSecret": "123456789012345678901234567890",
    "SessionCookieName": "my-cookie",
    "SessionTimeout": 3600,
    "Scope": "email",
    "AuthenticationRequestExtraParams": {
      "display": "page",
      "prompt": "login"
    },
    "OnUnauthenticatedRequest": "deny"
  },
  "Order": 1
},
{
  "Type": "forward",
  "TargetGroupArn": "arn:aws:elasticloadbalancing:region-code:account-
id:targetgroup/target-group-name/target-group-id",
  "Order": 2
}]

```

Unten ist ein Beispiel für die Datei `actions.json` angegeben, in der die Aktion `authenticate-cognito` und die Aktion `forward` enthalten sind.

```

[
  {
    "Type": "authenticate-cognito",
    "AuthenticateCognitoConfig": {
      "UserPoolArn": "arn:aws:cognito-idp:region-code:account-id:userpool/user-pool-
id",
      "UserPoolClientId": "abcdefghijklmnopqrstuvxyz123456789",
      "UserPoolDomain": "userPoolDomain1",
      "SessionCookieName": "my-cookie",
      "SessionTimeout": 3600,
      "Scope": "email",
      "AuthenticationRequestExtraParams": {
        "display": "page",
        "prompt": "login"
      },
      "OnUnauthenticatedRequest": "deny"
    },
    "Order": 1
  },
  {

```

```

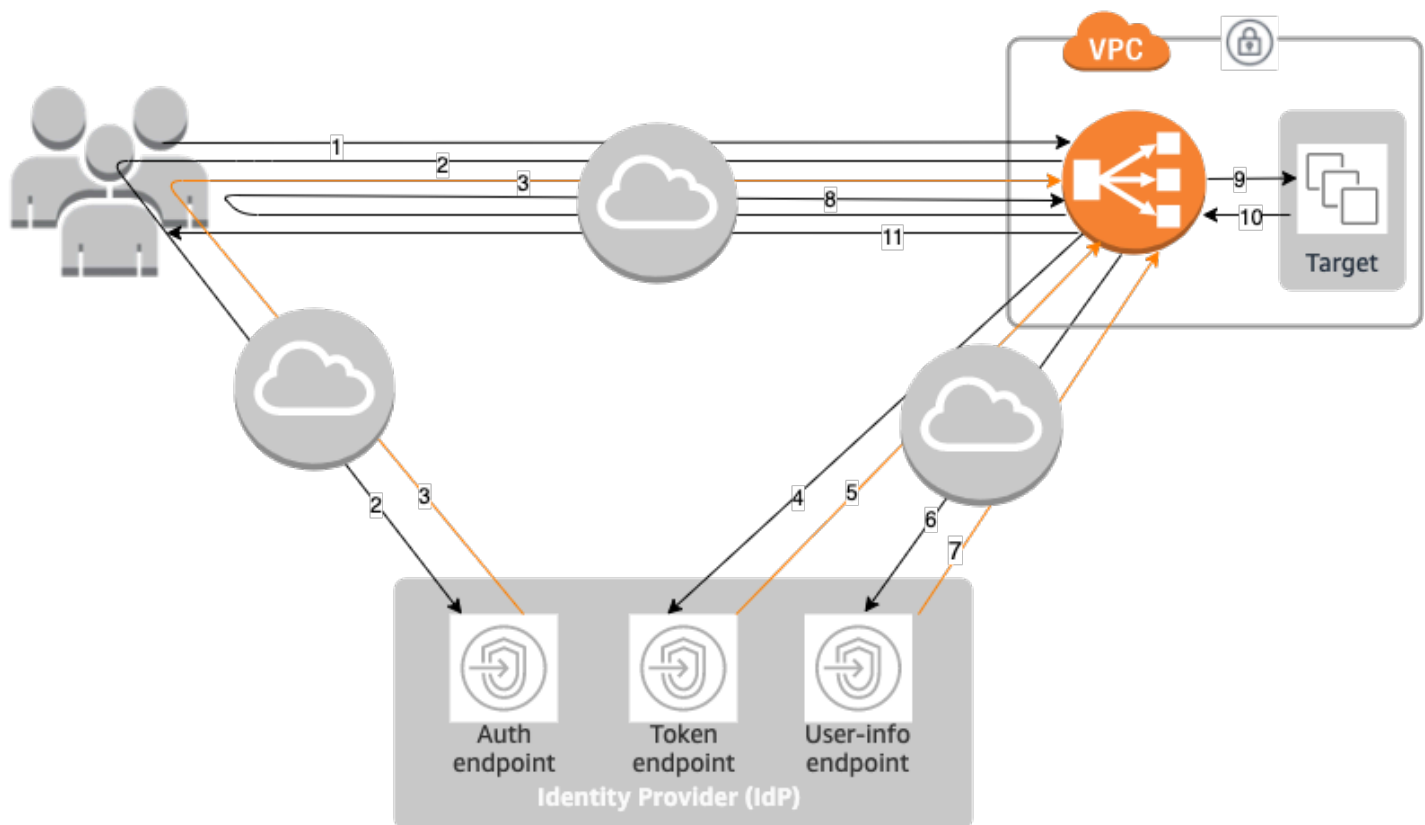
    "Type": "forward",
    "TargetGroupArn": "arn:aws:elasticloadbalancing:region-code:account-
id:targetgroup/target-group-name/target-group-id",
    "Order": 2
  }]

```

Weitere Informationen finden Sie unter [Listener-Regeln](#).

Authentifizierungsfluss

Das folgende Netzwerkdiagramm ist eine visuelle Darstellung dazu, wie ein Application Load Balancer OIDC zur Benutzerauthentifizierung verwendet.



Die folgenden nummerierten Punkte heben die im vorangehenden Netzwerkdiagramm gezeigten Elemente hervor und erläutern sie.

1. Der Benutzer sendet eine HTTPS-Anforderung an eine Website, die hinter einem Application Load Balancer gehostet wird. Wenn die Bedingungen für eine Regel mit einer Authentifizierungsaktion erfüllt sind, führt der Load Balancer in den Headern der Anforderungen eine Prüfung auf ein Cookie für eine Authentifizierungssitzung durch.

2. Falls das Cookie nicht vorhanden ist, leitet der Load Balancer den Benutzer an den Identitätsanbieter-Autorisierungsendpunkt um, damit der Benutzer vom Identitätsanbieter authentifiziert werden kann.
3. Nachdem der Benutzer authentifiziert wurde, wird er vom Identitätsanbieter mit einem Code für die Gewährung der Autorisierung zurück an den Load Balancer gesendet.
4. Der Load Balancer präsentiert dem IdP-Token-Endpunkt den Code für die Gewährung der Autorisierung.
5. Nach Erhalt eines gültigen Codes für die Gewährung der Autorisierung stellt der IdP dem Application Load Balancer das ID-Token und das Zugriffstoken zur Verfügung.
6. Der Application Load Balancer sendet dann das Zugriffstoken an den Endpunkt für Benutzerinformationen.
7. Der Endpunkt der Benutzerinformationen tauscht das Zugriffstoken gegen Benutzeransprüche aus.
8. Der Application Load Balancer leitet den Benutzer mit dem AWSELB-Cookie für die Authentifizierungssitzung zur ursprünglichen URI weiter. Da für die meisten Browser in Bezug auf Cookies eine Größenbeschränkung von 4 KB gilt, unterteilt der Load Balancer Anwendungs-Cookie, die größer als 4 KB sind, in mehrere Cookies. Wenn die Gesamtgröße der Benutzeransprüche und des Zugriffstokens, die vom Identitätsanbieter eingehen, über 11 KB liegt, gibt der Load Balancer die HTTP 500-Fehlermeldung an den Client zurück und erhöht die Metrik `ELBAuthUserClaimsSizeExceeded`.
9. Der Application Load Balancer validiert das Cookie und leitet die Benutzerinformationen an Ziele in den festgelegten `X-AMZN-OIDC-*`-HTTP-Headern weiter. Weitere Informationen finden Sie unter [Codierung von Benutzeransprüchen und Signaturverifizierung](#).
10. Das Ziel sendet eine Antwort zurück an den Application Load Balancer.
11. Der Application Load Balancer sendet die endgültige Antwort an den Benutzer.

Jede neue Anforderung durchläuft die Schritte 1 bis 11, während nachfolgende Anforderungen die Schritte 9 bis 11 durchlaufen. Das heißt, jede nachfolgende Anforderung beginnt bei Schritt 9, solange das Cookie nicht abgelaufen ist.

Das `AWSALBAuthNonce`-Cookie wird dem Anforderungsheader hinzugefügt, nachdem sich der Benutzer beim IdP authentifiziert hat. Dies ändert nichts daran, wie der Application Load Balancer Anfragen zur Umleitung vom IdP verarbeitet.

Wenn der Identitätsanbieter ein gültiges Aktualisierungstoken im ID-Token bereitstellt, speichert der Load Balancer das Aktualisierungstoken und nutzt es jeweils zum Aktualisieren der Benutzeransprüche, sobald das Zugriffstoken abgelaufen ist. Dieser Vorgang wird fortgesetzt, bis für die Sitzung der Wert für die Zeitüberschreitung erreicht ist oder die Identitätsanbieter-Aktualisierung fehlschlägt. Wenn sich der Benutzer abmeldet, tritt für die Aktualisierung ein Fehler auf und der Load Balancer leitet den Benutzer an den Identitätsanbieter-Autorisierungsendpunkt weiter. Auf diese Weise kann der Load Balancer Sitzungen verwerfen, nachdem sich der Benutzer abgemeldet hat. Weitere Informationen finden Sie unter [Sitzungs-Timeout](#).

Note

Der Ablauf des Cookies unterscheidet sich vom Ablauf der Authentifizierungssitzung. Der Ablauf des Cookies ist ein Attribut des Cookies, das auf 7 Tage festgelegt ist. Die tatsächliche Länge der Authentifizierungssitzung wird durch die Sitzungs-Zeitüberschreitung bestimmt, das im Application Load Balancer für das Authentifizierungsfeature konfiguriert wurde. Dieses Sitzungs-Timeout ist im Wert des Authentifizierungs-Cookies enthalten, der ebenfalls verschlüsselt ist.

Codierung von Benutzeransprüchen und Signaturverifizierung

Nachdem der Load Balancer einen Benutzer erfolgreich authentifiziert hat, sendet er die vom Identitätsanbieter erhaltenen Benutzeransprüche an das Ziel. Der Load Balancer signiert den Benutzeranspruch, damit diese Anwendungen die Signatur prüfen und sicherstellen können, dass die Ansprüche vom Load Balancer gesendet wurden.

Der Load Balancer fügt die folgenden HTTP-Header hinzu:

`x-amzn-oidc-accesstoken`

Das Zugriffstoken des Tokenendpunkts als Klartext

`x-amzn-oidc-identity`

Das Betrefffeld (sub) vom Endpunkt mit den Benutzerinformationen als Klartext

Hinweis: Der sub-Anspruch ist der beste Weg, um einen bestimmten Benutzer zu identifizieren..

`x-amzn-oidc-data`

Die Benutzeransprüche im JWT-Format (JSON-Web-Tokens)

Zugriffstoken und Benutzeransprüche unterscheiden sich von den ID-Token. Zugriffstoken und Benutzeransprüche ermöglichen nur den Zugriff auf Serverressourcen, während ID-Token zusätzliche Informationen zur Authentifizierung eines Benutzers enthalten. Der Application Load Balancer erstellt bei der Authentifizierung eines Benutzers ein neues Zugriffstoken und leitet nur die Zugriffstoken und Ansprüche an das Backend weiter, nicht jedoch die ID-Token-Informationen.

Diese Token weisen zwar das JWT-Format auf, sind aber keine ID-Token. Das JWT-Format umfasst einen Header, eine Nutzlast und eine Signatur (jeweils mit base64-URL-Verschlüsselung) und beinhalten Padding-Zeichen am Ende. Ein Application Load Balancer verwendet ES256 (ECDSA verwendet P-256 und SHA256), um die JWT-Signatur zu generieren.

Der JWT-Header ist ein JSON-Objekt mit den folgenden Feldern:

```
{
  "alg": "algorithm",
  "kid": "12345678-1234-1234-1234-123456789012",
  "signer": "arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/
app/load-balancer-name/load-balancer-id",
  "iss": "url",
  "client": "client-id",
  "exp": "expiration"
}
```

Die JWT-Nutzlast ist ein JSON-Objekt mit den Benutzeransprüchen, die vom Identitätsanbieterendpunkt mit den Benutzerinformationen empfangen wurden.

```
{
  "sub": "1234567890",
  "name": "name",
  "email": "alias@example.com",
  ...
}
```

Da der Load Balancer die Benutzeransprüche nicht verschlüsselt, empfehlen wir Ihnen, für die Zielgruppe die Nutzung von HTTPS zu konfigurieren. Wenn Sie für Ihre Zielgruppe die Nutzung von HTTP konfigurieren, sollten Sie darauf achten, den Datenverkehr mithilfe von Sicherheitsgruppen auf Ihren Load Balancer zu beschränken.

Um die Sicherheit zu gewährleisten, müssen Sie die Signatur überprüfen, bevor Sie eine Autorisierung auf der Grundlage der Ansprüche vornehmen, und überprüfen, ob das `signer` Feld im JWT-Header den erwarteten Application Load Balancer Balancer-ARN enthält.

Sie erhalten den öffentlichen Schlüssel, indem Sie die Schlüssel-ID aus dem JWT-Header verwenden, um den öffentlichen Schlüssel aus dem Endpunkt zu suchen. Der Endpunkt für jede AWS-Region lautet wie folgt:

```
https://public-keys.auth.elb.region.amazonaws.com/key-id
```

Denn AWS GovCloud (US) die Endpunkte lauten wie folgt:

```
https://s3-us-gov-west-1.amazonaws.com/aws-elb-public-keys-prod-us-gov-west-1/key-id  
https://s3-us-gov-east-1.amazonaws.com/aws-elb-public-keys-prod-us-gov-east-1/key-id
```

Im folgenden Beispiel wird veranschaulicht, wie Sie die Schlüssel-ID, den öffentlichen Schlüssel und die Nutzlast in Python 3.x abrufen:

```
import jwt
import requests
import base64
import json

# Step 1: Validate the signer
expected_alb_arn = 'arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/  
app/load-balancer-name/load-balancer-id'

encoded_jwt = headers.dict['x-amzn-oidc-data']
jwt_headers = encoded_jwt.split('.')[0]
decoded_jwt_headers = base64.b64decode(jwt_headers)
decoded_jwt_headers = decoded_jwt_headers.decode("utf-8")
decoded_json = json.loads(decoded_jwt_headers)
received_alb_arn = decoded_json['signer']

assert expected_alb_arn == received_alb_arn, "Invalid Signer"

# Step 2: Get the key id from JWT headers (the kid field)
kid = decoded_json['kid']

# Step 3: Get the public key from regional endpoint
url = 'https://public-keys.auth.elb.' + region + '.amazonaws.com/' + kid
req = requests.get(url)
pub_key = req.text

# Step 4: Get the payload
```

```
payload = jwt.decode(encoded_jwt, pub_key, algorithms=['ES256'])
```

Im folgenden Beispiel wird veranschaulicht, wie Sie die Schlüssel-ID, den öffentlichen Schlüssel und die Nutzlast in Python 2.7 abrufen:

```
import jwt
import requests
import base64
import json

# Step 1: Validate the signer
expected_alb_arn = 'arn:aws:elasticloadbalancing:region-code:account-id:loadbalancer/
app/load-balancer-name/load-balancer-id'

encoded_jwt = headers.dict['x-amzn-oidc-data']
jwt_headers = encoded_jwt.split('.')[0]
decoded_jwt_headers = base64.b64decode(jwt_headers)
decoded_json = json.loads(decoded_jwt_headers)
received_alb_arn = decoded_json['signer']

assert expected_alb_arn == received_alb_arn, "Invalid Signer"

# Step 2: Get the key id from JWT headers (the kid field)
kid = decoded_json['kid']

# Step 3: Get the public key from regional endpoint
url = 'https://public-keys.auth.elb.' + region + '.amazonaws.com/' + kid
req = requests.get(url)
pub_key = req.text

# Step 4: Get the payload
payload = jwt.decode(encoded_jwt, pub_key, algorithms=['ES256'])
```

Überlegungen

- In diesen Beispielen wird nicht behandelt, wie die Signatur des Ausstellers anhand der Signatur im Token überprüft werden kann.
- Standardbibliotheken sind nicht mit dem Padding kompatibel, das im Application-Load-Balancer-Authentifizierungstoken im JWT-Format enthalten ist.

Zeitüberschreitung

Sitzungs-Timeout

Das Aktualisierungstoken und die Sitzungs-Zeitüberschreitung arbeiten wie folgt zusammen:

- Wenn das Sitzungstimeout kürzer als die Ablaufzeit des Zugriffstokens ist, berücksichtigt der Load Balancer das Sitzungstimeout. Wenn der Benutzer eine aktive Sitzung mit dem IdP hat, wird der Benutzer möglicherweise nicht aufgefordert, sich erneut anzumelden. Andernfalls wird der Benutzer zur Anmeldung umgeleitet.
- Wenn das IdP-Sitzungs-Timeout länger als das Application-Load-Balancer-Sitzungs-Timeout ist, muss der Benutzer keine Anmeldeinformationen angeben, um sich erneut anzumelden. Stattdessen leitet der IdP die Anforderung mit einem neuen Code für die Gewährung der Autorisierung zurück zum Application Load Balancer. Autorisierungscode können nur einmal verwendet werden, auch wenn keine erneute Anmeldung erfolgt.
- Wenn das IdP-Sitzungs-Timeout gleich lang oder kürzer als das Application-Load-Balancer-Sitzungs-Timeout ist, wird der Benutzer aufgefordert, Anmeldeinformationen einzugeben, um sich erneut anzumelden. Nachdem sich der Benutzer angemeldet hat, leitet der IdP mit einem neuen Code für die Gewährung der Autorisierung die Anforderung zurück zum Application Load Balancer und der Rest des Authentifizierungsvorgangs wird fortgesetzt, bis die Anfrage das Backend erreicht.
- Wenn das Sitzungs-Timeout länger als die Ablaufzeit des Zugriffstokens ist und der Identitätsanbieter keine Aktualisierungstoken unterstützt, behält der Load Balancer die Authentifizierungssitzung bei, bis sie abgelaufen ist. Anschließend muss der Benutzer sich erneut anmelden.
- Wenn die Sitzungs-Zeitüberschreitung länger als die Ablaufzeit des Zugriffstokens ist und der Identitätsanbieter Aktualisierungstoken unterstützt, aktualisiert der Load Balancer die Benutzersitzung jedes Mal, wenn das Zugriffstoken abläuft. Der Load Balancer fordert den Benutzer erst zum erneuten Anmelden auf, nachdem für die Authentifizierungssitzung die Zeitüberschreitung erreicht wurde oder der Aktualisierungsablauf fehlgeschlagen ist.

Client-Anmelde-Timeout

Ein Client muss den Authentifizierungsprozess innerhalb von 15 Minuten einleiten und abschließen. Wenn ein Client die Authentifizierung nicht innerhalb der 15-minütigen Frist abschließt, erhält er vom Load Balancer einen HTTP-401-Fehler. Dieses Timeout kann nicht geändert oder entfernt werden.

Wenn ein Benutzer beispielsweise die Anmeldeseite über den Application Load Balancer lädt, muss er den Anmeldevorgang innerhalb von 15 Minuten abschließen. Wenn der Benutzer wartet und dann versucht, sich nach Ablauf des 15-Minuten-Timeouts anzumelden, gibt der Load Balancer einen HTTP-401-Fehler zurück. Der Benutzer muss die Seite aktualisieren und erneut versuchen, sich anzumelden.

Authentifizierung und Abmeldung

Wenn eine Anwendung einen authentifizierten Benutzer abmelden muss, sollte die Ablaufzeit des Cookies für die Authentifizierungssitzung auf „-1“ festgelegt und der Client an den Identitätsanbieterendpunkt für die Abmeldung (falls vom Identitätsanbieter unterstützt) weitergeleitet werden. Um zu verhindern, dass Benutzer ein gelöscht Cookie wiederverwenden, empfehlen wir Ihnen, die Ablaufzeit für das Zugriffstoken so kurz wie möglich zu konfigurieren. Wenn ein Client für einen Load Balancer ein Sitzungs-Cookie bereitstellt, das über ein abgelaufenes Zugriffstoken mit einem Nicht-NULL-Aktualisierungstoken verfügt, fragt der Load Balancer beim Identitätsanbieter nach, ob der Benutzer noch angemeldet ist.

Die Zielseite zur Client-Abmeldung ist eine nicht authentifizierte Seite. Das bedeutet, dass sie nicht hinter einer Application-Load-Balancer-Regel stehen kann, die eine Authentifizierung erfordert.

- Wenn eine Anforderung an das Ziel gesendet wird, muss die Anwendung den Ablauf für alle Authentifizierungs-Cookies auf -1 setzen. Application Load Balancer unterstützen Cookies mit einer Größe von bis zu 16 KB und können daher bis zu 4 Shards erstellen, die an den Client gesendet werden.
 - Wenn der IdP einen Abmeldeendpunkt hat, sollte er eine Umleitung zum IdP-Abmeldeendpunkt ausgeben, z. B. zu dem im Amazon-Cognito-Entwicklerhandbuch dokumentierten [LOGOUT-Endpunkt](#).
 - Wenn der IdP keinen Abmeldeendpunkt hat, geht die Anfrage zurück zur Client-Abmelde-Zielseite und der Anmeldevorgang wird neu gestartet.
- Unter der Annahme, dass der IdP über einen Abmeldeendpunkt verfügt, muss der IdP Zugriffs- und Aktualisierungstoken ablaufen lassen und den Benutzer zurück zur Zielseite für die Client-Abmeldung weiterleiten.
- Nachfolgende Anforderungen folgen dem ursprünglichen Authentifizierungsablauf.

HTTP-Header und Application Load Balancer

Die HTTP-Anforderungen und -Antworten verwenden Header-Felder, um Informationen über HTTP-Nachrichten zu senden. HTTP-Header werden automatisch hinzugefügt. Header-Felder sind durch einen Doppelpunkt getrennte Name/Wert-Paare, die durch eine Zeilenumschaltung und einen Zeilenvorschub getrennt sind. Ein Standardsatz von HTTP-Header-Feldern ist in RFC 2616, [Nachrichten-Header](#) definiert. Es sind auch Nicht-Standard-HTTP-Header verfügbar, die automatisch hinzugefügt und weithin von den Anwendungen verwendet werden. Einige der Nicht-Standard-HTTP-Header besitzen ein X-Forwarded-Präfix. Application Load Balancer unterstützen die folgenden X-Forwarded-Header.

Weitere Informationen zu HTTP-Verbindungen finden Sie unter [Weiterleitung von Anforderungen](#) im Benutzerhandbuch zu Elastic Load Balancing.

X-Forwarded-Header

- [X-Forwarded-For](#)
- [X-Forwarded-Proto](#)
- [X-Forwarded-Port](#)

X-Forwarded-For

Der X-Forwarded-For-Anforderungs-Header hilft Ihnen, die IP-Adresse eines Clients zu identifizieren, wenn Sie einen HTTP- oder HTTPS-Load Balancer verwenden. Da Load Balancer Datenverkehr zwischen Clients und Servern abfangen, enthalten Ihre Server-Zugriffsprotokolle nur die IP-Adresse des Load Balancers. Verwenden Sie das `routing.http.xff_header_processing.mode`-Attribut, um die IP-Adresse des Clients anzuzeigen. Dieses Attribut ermöglicht das Ändern, Beibehalten oder Entfernen der X-Forwarded-For-Header in der HTTP-Anforderung, bevor der Application Load Balancer die Anforderung an das Ziel sendet. Die möglichen Werte für dieses Attribut sind `append`, `preserve` und `remove`. Der Standardwert für dieses Attribut ist `append`.

Important

Der X-Forwarded-For Header sollte aufgrund potenzieller Sicherheitsrisiken mit Vorsicht verwendet werden. Die Einträge können nur dann als vertrauenswürdig angesehen werden,

wenn sie von Systemen hinzugefügt werden, die innerhalb des Netzwerks ordnungsgemäß gesichert sind.

Anfügen

Der Application Load Balancer speichert die IP-Adresse des Clients standardmäßig im X-Forwarded-For-Anforderungs-Header und übergibt den Header an Ihren Server. Wenn der X-Forwarded-For-Anforderungsheader nicht in der ursprünglichen Anforderung enthalten ist, erstellt der Load Balancer einen Header mit der Client-IP-Adresse als Anforderungswert. Andernfalls hängt der Load Balancer die Client-IP-Adresse an den vorhandenen Header an und leitet den Header dann an Ihren Server weiter. Der X-Forwarded-For-Anforderungsheader kann mehrere IP-Adressen enthalten, die durch Kommas getrennt sind.

Der X-Forwarded-For-Anforderungs-Header besitzt das folgende Format:

```
X-Forwarded-For: client-ip-address
```

Nachfolgend finden Sie ein Beispiel für einen X-Forwarded-For-Anforderungs-Header für einen Client mit der IP-Adresse 203.0.113.7.

```
X-Forwarded-For: 203.0.113.7
```

Nachfolgend finden Sie ein Beispiel für einen X-Forwarded-For-Anforderungs-Header für einen Client mit der IPv6-Adresse 2001:DB8::21f:5bff:febf:ce22:8a2e.

```
X-Forwarded-For: 2001:DB8::21f:5bff:febf:ce22:8a2e
```

Wenn das Attribut zur Beibehaltung des Client-Ports (`routing.http.xff_client_port.enabled`) im Load Balancer aktiviert ist, enthält der X-Forwarded-For-Anforderungsheader die an die `client-port-number` angehängte `client-ip-address`, durch einen Doppelpunkt getrennt. Der Header nimmt dann das folgende Format an:

```
IPv4 -- X-Forwarded-For: client-ip-address:client-port-number
```

```
IPv6 -- X-Forwarded-For: [client-ip-address]:client-port-number
```

Beachten Sie, dass bei IPv6 der Load Balancer, wenn er die `client-ip-address` an den vorhandenen Header anfügt, die Adresse in eckige Klammern setzt.

Nachfolgend finden Sie ein Beispiel für einen `X-Forwarded-For`-Anforderungs-Header für einen Client mit der IPv4-Adresse von `12.34.56.78` und einer Portnummer von `8080`.

```
X-Forwarded-For: 12.34.56.78:8080
```

Nachfolgend finden Sie ein Beispiel für einen `X-Forwarded-For`-Anforderungs-Header für einen Client mit der IPv6-Adresse von `2001:db8:85a3:8d3:1319:8a2e:370:7348` und einer Portnummer von `8080`.

```
X-Forwarded-For: [2001:db8:85a3:8d3:1319:8a2e:370:7348]:8080
```

Beibehalten

Der `preserve`-Modus im Attribut stellt sicher, dass der `X-Forwarded-For`-Header in der HTTP-Anforderung in keiner Weise geändert wird, bevor er an Ziele gesendet wird.

Remove

Der `remove`-Modus im Attribut entfernt den `X-Forwarded-For`-Header in der HTTP-Anforderung, bevor er an Ziele gesendet wird.

Note

Wenn Sie das Attribut zur Beibehaltung des Client-Ports (`routing.http.xff_client_port.enabled`) aktivieren und auch `preserve` oder `remove` für das `routing.http.xff_header_processing.mode` Attribut auswählen, überschreibt der Application Load Balancer das Attribut zur Erhaltung des Client-Ports. Je nach ausgewähltem Modus bleibt der `X-Forwarded-For`-Header unverändert oder wird entfernt, bevor er an die Ziele gesendet wird.

Die folgende Tabelle zeigt Beispiele für den `X-Forwarded-For`-Header, den das Ziel erhält, wenn Sie entweder den Modus `append`, `preserve` oder den Modus `remove` auswählen. In diesem Beispiel lautet die IP-Adresse des letzten Hops `127.0.0.1`.

Beschreibung der Anforderung	Beispielanforderung	XFF im append- Modus	XFF im preserve- Modus	XFF im remove- Modus
Anforderung wird ohne XFF-Header gesendet	GET / index.ht ml HTTP/1.1 Host: example.com	X-Forwarded-For: 127.0.0.1	Nicht vorhanden	Nicht vorhanden
Anforderung wird mit einem XFF-Header und einer Client-IP-Adresse gesendet.	GET / index.ht ml HTTP/1.1 Host: example.com X-Forwarded-For: 127.0.0.4	X-Forwarded-For: 127.0.0.4, 127.0.0.1	X-Forwarded-For: 127.0.0.4	Nicht vorhanden
Anforderung wird mit einem XFF-Header mit mehreren Client-IP-Adressen gesendet.	GET / index.ht ml HTTP/1.1 Host: example.com X-Forwarded-For: 127.0.0.4, 127.0.0.8	X-Forwarded-For: 127.0.0.4, 127.0.0.8, 127.0.0.1	X-Forwarded-For: 127.0.0.4, 127.0.0.8	Nicht vorhanden

So können Sie den X-Forwarded-For-Header mithilfe der Konsole ändern, beibehalten oder entfernen

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.

5. Wählen Sie im Abschnitt Datenverkehrskonfiguration unter Paketverarbeitung für X-Forwarded-For-Header die Option Anfügen (Standard), Beibehalten oder Entfernen aus.
6. Wählen Sie Änderungen speichern aus.

Um den X-Forwarded-For Header zu ändern, beizubehalten oder zu entfernen, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-load-balancer-attributes](#) mit dem Attribut `routing.http.xff_header_processing.mode`.

X-Forwarded-Proto

Der X-Forwarded-Proto-Anforderungs-Header hilft Ihnen, das Protokoll (HTTP oder HTTPS) zu identifizieren, das ein Client für die Verbindung zu Ihrem Load Balancer verwendet hat. Ihre Server-Zugriffsprotokolle enthalten nur das Protokoll zwischen dem Server und dem Load Balancer. Sie enthalten keine Informationen über das Protokoll zwischen dem Client und dem Load Balancer. Verwenden Sie den X-Forwarded-Proto-Anforderungs-Header, um das Protokoll zwischen dem Client und dem Load Balancer zu überprüfen. Elastic Load Balancing speichert das Protokoll zwischen dem Client und dem Load Balancer im X-Forwarded-Proto-Anforderungs-Header und übergibt den Header an den Server.

Ihre Anwendung oder Website kann das im X-Forwarded-Proto-Anforderungs-Header gespeicherte Protokoll verwenden, um eine Rückmeldung auszugeben, die auf die entsprechende URL umleitet.

Der X-Forwarded-Proto-Anforderungs-Header besitzt das folgende Format:

```
X-Forwarded-Proto: originatingProtocol
```

Das folgende Beispiel enthält einen X-Forwarded-Proto-Anforderungs-Header für eine Anforderung, die vom Client als HTTPS-Anforderung ausgegeben wurde:

```
X-Forwarded-Proto: https
```

X-Forwarded-Port

Mit dem X-Forwarded-Port-Anforderungs-Header können Sie den Zielport identifizieren, den der Client für die Verbindung mit dem Load Balancer verwendet hat.

Tags für Ihre Listener und Regeln

Mit Tags können Sie Ihre Listener und Regeln auf unterschiedliche Weise kategorisieren. Sie können Ressourcen beispielsweise nach Zweck, Inhaber oder Umgebung taggen.

Sie können jedem Listener und jeder Regel mehrere Tags hinzufügen. Tag-Schlüssel müssen für jeden Listener und jede Regel eindeutig sein. Wenn Sie eine Markierung mit einem Schlüssel hinzufügen, der dem Listener und der Regel bereits zugeordnet ist, ändert sich der Wert dieses Tags.

Wenn Sie ein Tag nicht mehr benötigen, können Sie es entfernen.

Einschränkungen

- Maximale Anzahl von Tags pro Ressource: 50
- Maximale Schlüssellänge: 127 Unicode-Zeichen
- Maximale Wertlänge: 255 Unicode-Zeichen
- Bei Tag-Schlüsseln und -Werten wird zwischen Groß- und Kleinschreibung unterschieden. Erlaubte Zeichen sind Buchstaben, Leerzeichen und Zahlen, die in UTF-8 darstellbar sind, sowie die folgenden Sonderzeichen: + - = _ : / @. Verwenden Sie keine führenden oder nachgestellten Leerzeichen.
- Verwenden Sie das `aws :` Präfix nicht in Ihren Tagnamen oder -Werten, da es für die AWS Verwendung reserviert ist. Sie können keine Tag-Namen oder Werte mit diesem Präfix bearbeiten oder löschen. Tags mit diesem Präfix werden nicht als Ihre Tags pro Ressourcenlimit angerechnet.

Aktualisieren der Listener-Tags

So aktualisieren Sie die Tags für einen Listener mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-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 Namen des Load Balancers aus, der den Listener enthält, den Sie aktualisieren möchten, um dessen Detailseite zu öffnen.
4. Führen Sie auf der Registerkarte Listener und Regeln einen der folgenden Schritte durch:
 - a. Wählen Sie den Text in der Spalte Protokoll: Port aus, um die Detailseite für den Listener zu öffnen.

Wählen Sie auf der Registerkarte Tags die Option Manage tags (Tags verwalten).

- b. Wählen Sie den Listener aus, für den Sie Tags aktualisieren möchten.

Wählen Sie Listener verwalten und dann Tags verwalten aus.

- c. Wählen Sie den Text in der Spalte Tags aus, um die Seite mit den Listener-Details auf der Registerkarte „Tags“ zu öffnen.

Wählen Sie Tags verwalten aus.

5. Auf der Seite Tags verwalten haben Sie folgende Möglichkeiten:
 - a. Um ein Tag zu aktualisieren, geben Sie neue Werte für Schlüssel und Wert ein.
 - b. Um ein Tag hinzuzufügen, wählen Sie Neues Tag hinzufügen aus und geben Sie Werte für Schlüssel und Wert ein.
 - c. Um ein Tag zu löschen, wählen Sie Entfernen neben dem Tag.
6. Wenn Sie die Aktualisierung der Tags abgeschlossen haben, wählen Sie Änderungen speichern.

Um die Tags für einen Listener mit dem zu aktualisieren AWS CLI

Verwenden Sie die Befehle [add-tags](#) und [remove-tags](#).

Aktualisieren von Regel-Tags

So aktualisieren Sie die Tags für eine Regel mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-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 Namen des Load Balancers aus, der die Regel enthält, die Sie aktualisieren möchten, um dessen Detailseite zu öffnen.
4. Wählen Sie auf der Registerkarte Listener und Regeln den Text in der Spalte Protokoll: Port des Listeners aus, der die zu aktualisierende Regel enthält, um die Detailseite für den Listener zu öffnen.
5. Führen Sie auf der Detailseite des Listeners einen der folgenden Schritte aus:
 - a. Wählen Sie den Text in der Spalte Namens-Tag aus, um die Detailseite für die Regel zu öffnen.

Wählen Sie auf der Seite mit den Details der Regel die Option Tags verwalten aus.

- b. Wählen Sie den Text in der Spalte Tags für die Regel aus, die Sie aktualisieren möchten.

Wählen Sie im Pop-up-Fenster mit der Zusammenfassung der Tags die Option Tags verwalten aus.

6. Auf der Seite Tags verwalten haben Sie folgende Möglichkeiten:
 - a. Um ein Tag zu aktualisieren, geben Sie neue Werte für Schlüssel und Wert ein.
 - b. Um ein Tag hinzuzufügen, wählen Sie Neues Tag hinzufügen aus und geben Sie Werte für Schlüssel und Wert ein.
 - c. Um ein Tag zu löschen, wählen Sie Entfernen neben dem Tag.
7. Wenn Sie die Aktualisierung der Tags abgeschlossen haben, wählen Sie Änderungen speichern.

Um die Tags für eine Regel mit dem zu aktualisieren AWS CLI

Verwenden Sie die Befehle [add-tags](#) und [remove-tags](#).

Löschen eines Listeners für Ihren Application Load Balancer

Sie können einen Listener jederzeit löschen. Wenn Sie einen Load Balancer löschen, werden all seine Listener gelöscht.

Löschen eines Listener mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Load Balancers aus.
3. Wählen Sie den Load Balancer aus.
4. Aktivieren Sie auf der Registerkarte Listener und Regeln das Kontrollkästchen für den Listener und wählen Sie Listener verwalten, Listener löschen aus.
5. Wenn Sie zur Bestätigung aufgefordert werden, geben Sie **confirm** ein und wählen Sie Löschen aus.

Um einen Listener mit dem zu löschen AWS CLI

Verwenden Sie den Befehl [delete-listener](#).

Zielgruppen für Ihre Application Load Balancer

Zielgruppen leiten Anforderungen an einzelne registrierte Ziele, z. B. EC2-Instances, über das Protokoll und die Port-Nummer, die Sie angeben, weiter. Sie können ein Ziel bei mehreren Zielgruppen registrieren. Sie können Zustandsprüfungen pro Zielgruppe konfigurieren. Zustandsprüfungen werden auf allen Zielen ausgeführt, die bei einer Zielgruppe registriert sind, welche in einer Listener-Regel für Ihren Load Balancer abgegeben ist.

Jede Zielgruppe wird verwendet, um Anfragen an ein oder mehrere registrierte Ziele weiterzuleiten. Beim Erstellen der jeweiligen Listener-Regeln geben Sie eine Zielgruppe und Bedingungen an. Wenn die Bedingung einer Regel erfüllt ist, wird der Datenverkehr an die entsprechende Zielgruppe weitergeleitet. Sie können unterschiedliche Zielgruppen für verschiedene Arten von Anfragen erstellen. Erstellen Sie beispielsweise eine Zielgruppe für allgemeine Anfragen und andere Zielgruppen für Anfragen an die Microservices für Ihre Anwendung. Sie können für jede Zielgruppe nur einen Load Balancer verwenden. Weitere Informationen finden Sie unter [Application-Load-Balancer-Komponenten](#).

Sie definieren Zustandsprüfungseinstellungen für Ihren Load Balancer pro Zielgruppe. Jede Zielgruppe verwendet die standardmäßigen Zustandsprüfungseinstellungen, es sei denn, Sie überschreiben diese, wenn Sie die Zielgruppe erstellen, oder ändern sie später. Nachdem Sie eine Zielgruppe in einer Regel für einen Listener angegeben haben, überwacht der Load Balancer kontinuierlich den Zustand aller mit der Zielgruppe registrierten Ziele, die in einer Availability Zone vorhanden sind, die für den Load Balancer aktiviert ist. Der Load Balancer leitet Anfragen an die registrierten Ziele weiter, die fehlerfrei sind.

Inhalt

- [Weiterleitungskonfiguration](#)
- [Zieltyp](#)
- [IP-Adresstyp](#)
- [Protokollversion](#)
- [Registrierte Ziele](#)
- [Zielgruppenattribute](#)
- [Routing-Algorithmen](#)
- [Automatische Zielgewichte \(ATW\)](#)
- [Verzögerung der Registrierungsaufhebung](#)

- [Modus des langsamen Hochfahrens](#)
- [Erstellen einer Zielgruppe](#)
- [Zustandsprüfungen für Ihre Zielgruppen](#)
- [Zonenübergreifendes Load Balancing für Zielgruppen](#)
- [Zustand der Zielgruppe](#)
- [Registrieren Sie Ziele mit Ihrer Zielgruppe](#)
- [Sticky Sessions für Ihren Application Load Balancer](#)
- [Lambda-Funktionen als Ziele](#)
- [Tags für Ihre Zielgruppe](#)
- [Löschen einer Zielgruppe](#)

Weiterleitungskonfiguration

Ein Load Balancer leitet standardmäßig mithilfe des Protokolls und der Portnummer, die Sie beim Erstellen der Zielgruppe angegeben haben, Anfragen an die Ziele weiter. Alternativ können Sie den für Weiterleitung von Datenverkehr zu einem Ziel verwendeten Port überschreiben, wenn Sie es bei der Zielgruppe registrieren.

Zielgruppen unterstützen die folgenden Protokolle und Ports:

- Protocols (Protokolle): HTTP, HTTPS
- Ports: 1-65535

Wenn eine Zielgruppe mit dem HTTPS-Protokoll konfiguriert ist oder HTTPS-Zustandsprüfungen verwendet, verwenden TLS-Verbindungen zu den Zielen die Sicherheitseinstellungen aus der `ELBSecurityPolicy-2016-08`-Richtlinie. Der Load Balancer stellt TLS-Verbindungen mit den Zielen her, wobei er die auf den Zielen installierten Zertifikate verwendet. Der Load Balancer überprüft diese Zertifikate nicht. Daher können Sie selbstsignierte Zertifikate oder Zertifikate verwenden, die abgelaufen sind. Da sich der Load Balancer und seine Ziele in einer Virtual Private Cloud (VPC) befinden, wird der Verkehr zwischen dem Load Balancer und den Zielen auf Paketebene authentifiziert, sodass kein Risiko von man-in-the-middle Angriffen oder Spoofing besteht, selbst wenn die Zertifikate auf den Zielen nicht gültig sind. Für den ausgehenden Datenverkehr AWS gilt nicht der gleiche Schutz, und es sind möglicherweise zusätzliche Schritte erforderlich, um den Datenverkehr weiter zu sichern.

Zieltyp

Wenn Sie eine Zielgruppe erstellen, legen Sie ihren Zieltyp fest, wodurch festgelegt wird, welchen Zieltyp Sie beim Registrieren von Zielen bei dieser Zielgruppe angeben. Nachdem Sie eine Zielgruppe erstellt haben, können Sie ihren Zieltyp nicht mehr ändern.

Die folgenden Zieltypen sind möglich:

instance

Die Ziele werden nach Instance-ID angegeben.

ip

Die Ziele sind IP-Adressen.

lambda

Das Ziel ist eine Lambda-Funktion.

Wenn der Zieltyp `ip` ist, können Sie IP-Adressen von einem der folgenden CIDR-Blöcke angeben:

- Die Subnetze der VPC für die Zielgruppe
- 10.0.0.0/8 ([RFC 1918](#))
- 100.64.0.0/10 ([RFC 6598](#))
- 172.16.0.0/12 (RFC 1918)
- 192.168.0.0/16 (RFC 1918)

Important

Sie können keine öffentlich weiterleitungsfähigen IP-Adressen angeben.

Mit allen unterstützten CIDR-Blöcken können Sie die folgenden Ziele bei einer Zielgruppe registrieren:

- Instances in einer VPC, die durch Peering mit der Load-Balancer-VPC verbunden sind (dieselbe Region oder eine andere Region).
- AWS Ressourcen, die über IP-Adresse und Port adressierbar sind (z. B. Datenbanken).

- Lokale Ressourcen, die AWS über eine AWS Direct Connect oder eine Site-to-Site-VPN-Verbindung verknüpft sind.

Note

Bei Application Load Balancern, die in einer lokalen Zone bereitgestellt werden, müssen sich die `ip`-Ziele in derselben lokalen Zone befinden, um Datenverkehr empfangen zu können. Weitere Informationen finden Sie unter [Was sind AWS Local Zones?](#)

Wenn Sie Ziele unter Verwendung einer Instance-ID angeben, wird Datenverkehr an Instances unter Verwendung der primären privaten IP-Adresse weitergeleitet, die in der primären Netzwerkschnittstelle für die Instance angegeben ist. Wenn Sie Ziele unter Verwendung von IP-Adressen angeben, können Sie den Datenverkehr über eine beliebige private IP-Adresse aus einer oder mehreren Netzwerkschnittstellen an eine Instance weiterleiten. Auf diese Weise können mehrere Anwendungen auf eine Instance denselben Port verwenden. Jede Netzwerkschnittstelle kann über eine eigene Sicherheitsgruppe verfügen.

Wenn der Zieltyp der Zielgruppe `Lambda` lautet, können Sie eine einzelne Lambda-Funktion registrieren. Wenn der Load Balancer eine Anfrage für eine Lambda-Funktion empfängt, ruft er die Lambda-Funktion auf. Weitere Informationen finden Sie unter [Lambda-Funktionen als Ziele](#).

Sie können Amazon Elastic Container Service (Amazon ECS) als Ziel für Ihren Application Load Balancer konfigurieren. Weitere Informationen finden Sie unter [Creating an Application Load Balancer](#) im Amazon Elastic Container Service-Benutzerhandbuch für AWS Fargate.

IP-Adresstyp

Wenn Sie eine neue Zielgruppe erstellen, können Sie den IP-Adresstyp Ihrer Zielgruppe auswählen. Dadurch wird die IP-Version gesteuert, die für die Kommunikation mit Zielen und die Überprüfung ihres Status verwendet wird.

Application Load Balancer unterstützen sowohl IPv4- als auch IPv6-Zielgruppen. Die Standardauswahl ist IPv4.

Überlegungen

- Alle IP-Adressen innerhalb einer Zielgruppe müssen denselben IP-Adresstyp haben. Sie können beispielsweise kein IPv4-Ziel bei einer IPv6-Zielgruppe registrieren.

- IPv6-Zielgruppen können nur mit dualstack-Load-Balancern verwendet werden.
- IPv6-Zielgruppen unterstützen Ziele vom Typ „IP“ und „Instance“.

Protokollversion

Standardmäßig senden Application Load Balancer Anforderungen über HTTP/1.1 an Ziele. Sie können die Protokollversion verwenden, um Anforderungen mit HTTP/2 oder gRPC an Ziele zu senden.

In der folgenden Tabelle sind die Ergebnisse für die Kombinationen aus Anforderungsprotokoll und Zielgruppen-Protokollversion zusammengefasst.

Anforderungsprotokoll	Protokollversion	Ergebnis
HTTP/1.1	HTTP/1.1	Herzlichen Glückwunsch
HTTP/2	HTTP/1.1	Herzlichen Glückwunsch
gRPC	HTTP/1.1	Fehler
HTTP/1.1	HTTP/2	Fehler
HTTP/2	HTTP/2	Herzlichen Glückwunsch
gRPC	HTTP/2	Erfolg, wenn Ziele gRPC unterstützen
HTTP/1.1	gRPC	Fehler
HTTP/2	gRPC	Erfolg bei einer POST-Anforderung
gRPC	gRPC	Herzlichen Glückwunsch

Überlegungen zur gRPC-Protokollversion

- Das einzige unterstützte Listener-Protokoll ist HTTPS.
- Der einzige unterstützte Aktionstyp für Listener-Regeln ist `forward`.

- Die einzigen unterstützten Zieltypen sind `instance` und `ip`.
- Der Load Balancer analysiert gRPC-Anfragen und leitet gRPC-Aufrufe basierend auf dem Paket, dem Dienst und der Methode an die entsprechenden Zielgruppen weiter.
- Der Load Balancer unterstützt unäres, clientseitiges Streaming, serverseitiges Streaming und bidirektionales Streaming.
- Sie müssen eine benutzerdefinierte Zustandsprüfungsmethode mit dem Format `/package.service/method` bereitstellen.
- Sie müssen die gRPC-Statuscodes angeben, die verwendet werden, um ein Ziel auf eine erfolgreiche Antwort zu überprüfen.
- Sie können keine Lambda-Funktionen als Ziel verwenden.

Überlegungen zur HTTP/2-Protokollversion

- Das einzige unterstützte Listener-Protokoll ist HTTPS.
- Der einzige unterstützte Aktionstyp für Listener-Regeln ist `forward`.
- Die einzigen unterstützten Zieltypen sind `instance` und `ip`.
- Der Load Balancer unterstützt Streaming von Clients. Der Load Balancer unterstützt kein Streaming zu den Zielen.

Registrierte Ziele

Ihr Load Balancer dient als zentraler Kontaktpunkt für Clients und verteilt eingehenden Datenverkehr an die fehlerfreien registrierten Ziele. Sie können jedes Ziel bei einer oder mehreren Zielgruppen registrieren.

Wenn die Nachfrage nach Ihrer Anwendung steigt, können Sie zusätzliche Ziele bei einer oder mehreren Zielgruppen registrieren, um die Nachfrage zu bewältigen. Der Load Balancer leitet den Traffic an ein neu registriertes Ziel weiter, sobald der Registrierungsprozess abgeschlossen ist und das Ziel die erste Zustandsprüfung bestanden hat, unabhängig vom konfigurierten Schwellenwert.

Wenn die Nachfrage nach Ihrer Anwendung sinkt oder Sie Ihre Ziele warten müssen, können Sie die Registrierung von Zielen bei Ihren Zielgruppen aufheben. Bei der Aufhebung der Registrierung eines Ziels wird es aus Ihrer Zielgruppe entfernt. Ansonsten hat dies keine Auswirkungen auf das Ziel. Der Load Balancer stoppt das Weiterleiten von Anfragen an ein Ziel, sobald die Registrierung des Ziels aufgehoben wird. Das Ziel wechselt in den Zustand `draining`, bis laufende Anfragen

abgeschlossen wurden. Sie können das Ziel erneut bei der Zielgruppe registrieren, wenn es bereit ist, wieder Anfragen zu erhalten.

Wenn Sie Ziele nach Instance-ID registrieren, können Sie Ihren Load Balancer mit einer Auto-Scaling-Gruppe verwenden. Nachdem Sie eine Zielgruppe einer Auto-Scaling-Gruppe angefügt haben, registriert Auto Scaling Ihre Ziele bei der Zielgruppe für Sie, wenn es sie startet. Weitere Informationen finden Sie unter [Anhängen eines Load Balancers an Ihre Auto-Scaling-Gruppe](#) im Benutzerhandbuch zu Amazon EC2 Auto Scaling.

Einschränkungen

- Es ist nicht möglich, IP-Adressen eines anderen Application Load Balancers in derselben VPC zu registrieren. Wenn der andere Application Load Balancer sich in einer VPC befindet, die durch Peering mit dem Load Balancer verbunden ist, können Sie die IP-Adressen registrieren.
- Sie können Instances nicht nach Instance-ID registrieren, wenn sie sich in einer VPC befinden, die mit der Load-Balancer-VPC gekoppelt ist (dieselbe Region oder eine andere Region). Sie können diese Instances nach IP-Adresse registrieren.

Zielgruppenattribute

Die folgenden Zielgruppenattribute werden unterstützt, wenn die Zielgruppe vom Typ `instance` oder `ip` ist:

`deregistration_delay.timeout_seconds`

Die Zeit, die Elastic Load Balancing wartet, bevor die Registrierung eines Ziels aufgehoben wird. Der Bereich liegt zwischen 0 und 3 600 Sekunden. Der Standardwert beträgt 300 Sekunden.

`load_balancing.algorithm.type`

Der Lastausgleichsalgorithmus bestimmt, wie der Load Balancer Ziele beim Routing von Anforderungen auswählt. Der Wert ist `round_robinleast_outstanding_requests`, oder `weighted_random`. Der Standardwert ist `round_robin`.

`load_balancing.algorithm.anomaly_mitigation`

Nur verfügbar, wenn `load_balancing.algorithm.type` ist `weighted_random`. Zeigt an, ob die Minimierung von Anomalien aktiviert ist. Der Wert ist entweder `on` oder `off`. Der Standardwert ist `off`.

`load_balancing.cross_zone.enabled`

Gibt an, ob zonenübergreifendes Load Balancing aktiviert ist. Der Wert ist entweder `true`, `false` oder `use_load_balancer_configuration`. Der Standardwert ist `use_load_balancer_configuration`.

`slow_start.duration_seconds`

Der Zeitraum in Sekunden, in dem der Load Balancer für ein neu registriertes Ziel einen linear zunehmenden Anteil des Datenverkehrs an die Zielgruppe sendet. Der Bereich liegt zwischen 30 und 900 Sekunden (15 Minuten). Die Standardwert ist 0 Sekunden (deaktiviert).

`stickiness.enabled`

Gibt an, ob Sticky Sessions aktiviert sind. Der Wert ist entweder `true` oder `false`. Der Standardwert ist `false`.

`stickiness.app_cookie.cookie_name`

Der Name der Anwendungs-Cookies. Der Name des Anwendungs-Cookies darf nicht die folgenden Präfixe haben: `AWSALB`, `AWSALBAPP` oder `AWSALBTG`. Sie sind für die Verwendung durch den Load Balancer reserviert.

`stickiness.app_cookie.duration_seconds`

Die Ablaufzeit anwendungsbasierter Cookies in Sekunden. Nach Ablauf dieses Zeitraums wird das Cookie als veraltet eingestuft. Der Mindestwert ist 1 Sekunde und der Maximalwert ist 7 Tage (604800 Sekunden). Die Standardwert ist 1 Tag (86 400 Sekunden).

`stickiness.lb_cookie.duration_seconds`

Die Ablaufzeit der auf Dauer basierenden Cookies in Sekunden. Nach Ablauf dieses Zeitraums wird das Cookie als veraltet eingestuft. Der Mindestwert ist 1 Sekunde und der Maximalwert ist 7 Tage (604800 Sekunden). Die Standardwert ist 1 Tag (86 400 Sekunden).

`stickiness.type`

Die Art der „Sticky Sessions“. Die möglichen Werte sind `lb_cookie` und `app_cookie`.

`target_group_health.dns_failover.minimum_healthy_targets.count`

Die Mindestanzahl von Zielen, die fehlerfrei sein müssen. Wenn die Anzahl der fehlerfreien Ziele unter diesem Wert liegt, markieren Sie die Zone im DNS als fehlerhaft, sodass der Datenverkehr nur an fehlerfreie Zonen weitergeleitet wird. Die möglichen Werte sind `off` oder eine ganze Zahl von 1 bis zur maximalen Anzahl von Zielen. Falls `off`, wird DNS-Failaway deaktiviert, was bedeutet, dass jede Zielgruppe unabhängig zum DNS-Failover beiträgt. Der Standardwert ist 1.

`target_group_health.dns_failover.minimum_healthy_targets.percentage`

Der Mindestprozentsatz der Ziele, die fehlerfrei sein müssen. Wenn der Prozentsatz fehlerfreier Ziele unter diesem Wert liegt, markieren Sie die Zone im DNS als fehlerhaft, sodass der Datenverkehr nur an fehlerfreie Zonen weitergeleitet wird. Die möglichen Werte sind `off` oder eine ganze Zahl von 1 bis zur maximalen Anzahl von Zielen. Falls `off`, wird DNS-Failaway deaktiviert, was bedeutet, dass jede Zielgruppe unabhängig zum DNS-Failover beiträgt. Der Standardwert ist 1.

`target_group_health.unhealthy_state_routing.minimum_healthy_targets.count`

Die Mindestanzahl von Zielen, die fehlerfrei sein müssen. Wenn die Anzahl fehlerfreier Ziele unter diesem Wert liegt, senden Sie Datenverkehr an alle Ziele, einschließlich nicht fehlerfreier Ziele. Der Bereich reicht von 1 bis zur maximalen Anzahl von Zielen. Der Standardwert ist 1.

`target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage`

Der Mindestprozentsatz der Ziele, die fehlerfrei sein müssen. Wenn der Prozentsatz fehlerfreier Ziele unter diesem Wert liegt, senden Sie Datenverkehr an alle Ziele, einschließlich nicht fehlerfreier Ziele. Die möglichen Werte sind `off` oder eine Ganzzahl von 1 bis 100. Der Standardwert ist `off`.

Das folgende Zielgruppenattribut wird unterstützt, wenn der Zielgruppentyp `lambda` ist:

`lambda.multi_value_headers.enabled`

Gibt an, ob die Anfrage- und Antwort-Header, die zwischen dem Load Balancer und der Lambda-Funktion ausgetauscht werden, Arrays von Werten oder Zeichenfolgen enthalten. Die möglichen Werte sind `true` oder `false`. Der Standardwert ist `false`. Weitere Informationen finden Sie unter [Header mit mehreren Werten](#).

Routing-Algorithmen

Ein Routing-Algorithmus ist die Methode, mit der der Load Balancer bestimmt, welche Ziele Anfragen erhalten. Der Round-Robin-Routing-Algorithmus wird standardmäßig verwendet, um Anfragen auf Zielgruppenebene weiterzuleiten. Je nach den Anforderungen Ihrer Anwendung sind auch die am wenigsten ausstehenden Anfragen und gewichtete zufällige Routing-Algorithmen verfügbar. Eine Zielgruppe kann jeweils nur über einen aktiven Routing-Algorithmus verfügen, der Routing-Algorithmus kann jedoch bei Bedarf aktualisiert werden.

Wenn Sie Sticky Sessions aktivieren, wird der ausgewählte Routing-Algorithmus für die anfängliche Zielauswahl verwendet. Künftige Anfragen von demselben Client werden an dasselbe Ziel weitergeleitet, wobei der ausgewählte Routing-Algorithmus umgangen wird.

Rundenturnier

- Der Round-Robin-Routing-Algorithmus leitet Anfragen gleichmäßig in einer sequentiellen Reihenfolge an gesunde Ziele in der Zielgruppe weiter.
- Dieser Algorithmus wird häufig verwendet, wenn die eingehenden Anfragen eine ähnliche Komplexität aufweisen, die registrierten Ziele eine ähnliche Verarbeitungsfähigkeit aufweisen oder wenn Sie Anfragen gleichmäßig auf die Ziele verteilen müssen.

Am wenigsten ausstehende Anfragen

- Der Routing-Algorithmus für die wenigsten ausstehenden Anfragen leitet Anfragen an die Ziele mit der geringsten Anzahl an laufenden Anfragen weiter.
- Dieser Algorithmus wird häufig verwendet, wenn die eingehenden Anfragen unterschiedlich komplex sind und die Verarbeitungskapazität der registrierten Ziele unterschiedlich ist.
- Wenn ein Load Balancer, der HTTP/2 unterstützt, Ziele verwendet, die nur HTTP/1.1 unterstützen, konvertiert er die Anfrage in mehrere HTTP/1.1-Anfragen. In dieser Konfiguration behandelt der Algorithmus für die wenigsten ausstehenden Anfragen jede HTTP/2-Anfrage als mehrere Anfragen.
- Bei Verwendung wird WebSockets das Ziel anhand des Algorithmus für die wenigsten ausstehenden Anfragen ausgewählt. Nach der Auswahl stellt der Load Balancer eine Verbindung zum Ziel her und sendet alle Nachrichten über diese Verbindung.
- Der Routing-Algorithmus für die wenigsten ausstehenden Anfragen kann nicht im langsamen Startmodus verwendet werden.

Zufällig gewichtet

- Der gewichtete Zufalls-Routing-Algorithmus leitet Anfragen gleichmäßig und in zufälliger Reihenfolge an gesunde Ziele in der Zielgruppe weiter.
- Dieser Algorithmus unterstützt die Minimierung von ATW-Anomalien (Automatic Target Weights).
- Der Algorithmus für gewichtetes zufälliges Routing kann nicht im langsamen Startmodus verwendet werden.

Ändern Sie den Routing-Algorithmus einer Zielgruppe

Sie können den Routing-Algorithmus für Ihre Zielgruppe jederzeit ändern.

Um den Routing-Algorithmus mit der neuen Konsole zu ändern

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie auf der Detailseite der Zielgruppen auf der Registerkarte Attribute die Option Bearbeiten aus.
5. Wählen Sie auf der Seite Zielgruppenattribute bearbeiten im Abschnitt Verkehrskonfiguration unter Load Balancing Algorithm die Optionen Round Robin, Least Outstanding Requests oder Weighted Random aus.
6. Wählen Sie Änderungen speichern aus.

Um den Routing-Algorithmus mit dem zu ändern AWS CLI

Verwenden Sie den Befehl [modify-target-group-attributes](#) zusammen mit dem Attribut `load_balancing.algorithm.type`.

Automatische Zielgewichte (ATW)

Automatic Target Weights (ATW) überwacht ständig die Ziele, auf denen Ihre Anwendungen ausgeführt werden, und erkennt signifikante Leistungsabweichungen, sogenannte Anomalien. ATW bietet die Möglichkeit, die Menge des an Ziele weitergeleiteten Datenverkehrs dynamisch anzupassen, indem Datenanomalien in Echtzeit erkannt werden.

Automatic Target Weights (ATW) führt automatisch eine Anomalieerkennung für jeden Application Load Balancer in Ihrem Konto durch. Wenn anomale Ziele identifiziert werden, kann ATW automatisch versuchen, sie zu stabilisieren, indem es den Umfang des Datenverkehrs, den sie weiterleiten, reduziert. Dies wird als Abwehr von Anomalien bezeichnet. ATW optimiert kontinuierlich die Verteilung des Datenverkehrs, um die Erfolgsquoten pro Ziel zu maximieren und gleichzeitig die Ausfallraten der Zielgruppen zu minimieren.

Überlegungen:

- Die Anomalieerkennung überwacht derzeit HTTP 5xx-Antwortcodes, die von Ihren Zielen kommen, und Verbindungsfehler zu diesen. Die Anomalieerkennung ist immer aktiviert und kann nicht ausgeschaltet werden.
- ATW wird nicht unterstützt, wenn Lambda als Ziel verwendet wird.

Anomalie-Erkennung

Die ATW-Anomalieerkennung überwacht alle Ziele, die eine signifikante Abweichung im Verhalten von anderen Zielen in ihrer Zielgruppe aufweisen. Diese Abweichungen, die als Anomalien bezeichnet werden, werden ermittelt, indem die prozentualen Fehler eines Ziels mit den prozentualen Fehlern anderer Ziele in der Zielgruppe verglichen werden. Bei diesen Fehlern kann es sich sowohl um Verbindungsfehler als auch um HTTP-Fehlercodes handeln. Ziele, die deutlich höhere Werte als ihre Mitbewerber melden, werden dann als anomal eingestuft.

Für die Erkennung von Anomalien sind mindestens drei gesunde Ziele in der Zielgruppe erforderlich. Wenn ein Ziel für eine Zielgruppe registriert ist, muss es zuerst die Gesundheitschecks bestehen, um Traffic empfangen zu können. Sobald das Ziel das Ziel empfängt, beginnt ATW mit der Überwachung des Ziels und veröffentlicht kontinuierlich das Ergebnis der Anomalie. Für Ziele ohne Anomalien lautet das Anomalieergebnis. `normal` Für Ziele mit Anomalien lautet das Anomalieergebnis. `anomalous`

Die ATW-Anomalieerkennung funktioniert unabhängig von den Gesundheitschecks der Zielgruppe. Ein Ziel kann alle Zustandsprüfungen für die Zielgruppe bestehen, aber aufgrund einer erhöhten Fehlerquote trotzdem als anomal eingestuft werden. Wenn Ziele anomal werden, hat das keinen Einfluss auf den Status ihrer Zielgruppen-Gesundheitschecks.

Status der Erkennung von Anomalien

ATW veröffentlicht kontinuierlich den Status der Anomalieerkennungen, die es an Zielen durchführt. Sie können den aktuellen Status jederzeit mit der Taste oder einsehen. AWS Management Console
AWS CLI

Um den Status der Anomalieerkennung mithilfe der Konsole anzuzeigen

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.

3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie auf der Detailseite der Zielgruppen die Registerkarte Ziele aus.
5. In der Tabelle Registrierte Ziele können Sie den Anomaliestatus der einzelnen Ziele in der Spalte mit den Ergebnissen der Anomalieerkennung einsehen.

Wenn keine Anomalien festgestellt wurden, lautet das Ergebnis. `normal`

Wenn Anomalien festgestellt wurden, lautet das Ergebnis. `anomalous`

Um die Ergebnisse der Anomalieerkennung anzuzeigen, verwenden Sie AWS CLI

Verwenden Sie den Befehl [describe-target-health](#), wobei der `Include.member.N` Attributwert auf `AnomalyDetection` gesetzt ist.

Minimierung von Anomalien

Important

Die Funktion zur Minimierung von Anomalien von ATW ist nur verfügbar, wenn der Algorithmus Weighted Random Routing verwendet wird.

Die ATW-Abschwächung leitet den Verkehr automatisch von anomalen Zielen weg und gibt ihnen so die Möglichkeit, sich zu erholen.

Während der Schadensbegrenzung:

- ATW passt in regelmäßigen Abständen die Menge des Datenverkehrs an, der zu anomalen Zielen geleitet wird. Derzeit beträgt der Zeitraum alle fünf Sekunden.
- ATW reduziert die Menge des Datenverkehrs, der zu anomalen Zielen geleitet wird, auf das Minimum, das zur Behebung von Anomalien erforderlich ist.
- Bei Zielen, die nicht mehr als `anomalous` erkannt werden, wird nach und nach mehr Traffic an sie weitergeleitet, bis sie die Parität mit anderen normalen Zielen in der Zielgruppe erreichen.

Schalten Sie die ATW-Abschwächung von Anomalien ein

Sie können die Minimierung von Anomalien jederzeit aktivieren.

Um die Minimierung von Anomalien mithilfe der Konsole zu aktivieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie auf der Detailseite der Zielgruppen auf der Registerkarte Attribute die Option Bearbeiten aus.
5. Stellen Sie sicher, dass auf der Seite Zielgruppenattribute bearbeiten im Abschnitt Verkehrskonfiguration unter Load Balancing-Algorithmus die Option Weighted Random ausgewählt ist.

Hinweis: Wenn der gewichtete Zufallsalgorithmus anfänglich ausgewählt ist, ist die Anomalieerkennung standardmäßig aktiviert.

6. Stellen Sie sicher, dass unter Minimierung von Anomalien die Option Minimierung von Anomalien ausgewählt ist.
7. Wählen Sie Änderungen speichern aus.

Um die Minimierung von Anomalien zu aktivieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-target-group-attributes](#) zusammen mit dem Attribut `load_balancing.algorithm.anomaly_mitigation`.

Status der Minimierung von Anomalien

Immer wenn ATW eine Risikominderung an einem Ziel durchführt, können Sie den aktuellen Status jederzeit mit dem oder anzeigen. AWS Management Console AWS CLI

Um den Status zur Behebung von Anomalien mithilfe der Konsole einzusehen

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie auf der Detailseite der Zielgruppen die Registerkarte Ziele aus.
5. In der Tabelle Registrierte Ziele können Sie den Status der einzelnen Ziele zur Minimierung von Anomalien in der Spalte Minderung wirksam einsehen.

Wenn die Schadensbegrenzung nicht im Gange ist, lautet der Status. `yes`

Wenn die Schadensbegrenzung im Gange ist, lautet der Status. `no`

Um den Status der Minimierung von Anomalien anzuzeigen, verwenden Sie den AWS CLI

Verwenden Sie den Befehl [describe-target-health](#), wobei der Attributwert auf `set` ist.

```
Include.member.N AnomalyDetection
```

Verzögerung der Registrierungsaufhebung

Elastic Load Balancing stoppt das Senden von Anforderungen an Ziele, deren Registrierung aufgehoben wird. Standardmäßig wartet Elastic Load Balancing 300 Sekunden, bevor die Registrierungsaufhebung abgeschlossen wird. So können laufende Anforderungen an das Ziel abgeschlossen werden. Um die Zeitdauer zu ändern, die Elastic Load Balancing wartet, müssen Sie den Wert für die Verzögerung der Registrierungsaufhebung anpassen.

Der ursprüngliche Zustand eines Ziels, dessen Registrierung aufgehoben wird, lautet `draining`. Nach Ablauf der Verzögerung der Registrierungsaufhebung wird die Registrierungsaufhebung abgeschlossen und der Zustand des Ziels lautet `unused`. Wenn das Ziel Teil einer Auto-Scaling-Gruppe ist, kann es beendet und ersetzt werden.

Falls ein Ziel, dessen Registrierung aufgehoben wird, keine aktiven Anforderungen und keine aktiven Verbindungen aufweist, wird der Aufhebungsprozess von Elastic Load Balancing sofort abgeschlossen, ohne auf das Verstreichen der Verzögerung für die Registrierungsaufhebung zu warten. Auch wenn die Zielregistrierung aufgehoben wurde, wird bis zum Ablauf des Timeouts der Verzögerung der Registrierungsaufhebung `draining` als Status des Ziels angezeigt. Nach Ablauf des Timeouts wechselt das Ziel in einen `unused`-Zustand.

Wenn ein Ziel, dessen Registrierung aufgehoben wird, die Verbindung beendet, bevor die Verzögerung der Registrierungsaufhebung abgelaufen ist, erhält der Client eine 500-Level-Fehlerantwort.

So aktualisieren Sie den Wert für die Verzögerung der Registrierungsaufhebung mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.

3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie auf der Registerkarte Gruppendetails im Abschnitt Attribute die Option Bearbeiten aus.
5. Ändern Sie auf der Seite Attribute bearbeiten den Wert für Abmeldeverzögerung nach Bedarf.
6. Wählen Sie Änderungen speichern.

Um den Wert für die Verzögerung bei der Abmeldung zu aktualisieren, verwenden Sie den AWS CLI

Verwenden Sie den Befehl [modify-target-group-attributes](#) zusammen mit dem Attribut `deregistration_delay.timeout_seconds`.

Modus des langsamen Hochfahrens

Standardmäßig beginnt ein Ziel mit dem Empfang seines vollständigen Anteils an Anfragen, sobald es bei einer Zielgruppe registriert ist und die anfängliche Zustandsprüfung bestanden hat. Der Modus des langsamen Hochfahrens bietet Zielen mehr Zeit zum Warmlaufen, bevor der Load Balancer ihnen den vollständigen Anteil an Anfragen sendet.

Nachdem Sie das langsame Hochfahren für eine Zielgruppe aktiviert haben, wechseln die Ziele in den Modus des langsamen Hochfahrens, wenn sie von der Zielgruppe als fehlerfrei eingestuft werden. Ein Ziel im Modus des langsamen Hochfahrens beendet diesen, wenn die konfigurierte Dauer des langsamen Hochfahrens abgelaufen ist oder das Ziel fehlerhaft ist. Der Load Balancer erhöht linear die Anzahl von Anfragen, die er im Modus des langsamen Hochfahrens an ein Ziel senden kann. Nachdem ein fehlerfreies Ziel den Modus des langsamen Hochfahrens beendet hat, kann der Load Balancer diesem den vollständigen Anteil von Anfragen senden.

Überlegungen

- Wenn Sie das langsame Hochfahren für eine Zielgruppe aktivieren, gehen die bei der Zielgruppe registrierten fehlerfreien Ziele nicht in den Modus des langsamen Hochfahrens über.
- Wenn Sie das langsame Hochfahren für eine leere Zielgruppe aktivieren und dann mit einer einzigen Registrierungsoperation auf ein Ziel anwenden, dann gehen diese Ziele nicht in den Modus des langsamen Hochfahrens über. Neu registrierte Ziele gehen nur dann in den Modus des langsamen Hochfahrens über, wenn mindestens ein fehlerfreies Ziel vorhanden ist, das sich nicht im Modus des langsamen Hochfahrens befindet.
- Wenn Sie die Registrierung eines Ziels im Modus des langsamen Hochfahrens aufheben, beendet das Ziel den Modus des langsamen Hochfahrens. Wenn Sie dasselbe Ziel erneut registrieren,

wechselt es in den Modus des langsamen Hochfahrens, wenn es von der Zielgruppe als fehlerfrei eingestuft wird.

- Wenn ein Ziel im Modus des langsamen Hochfahrens fehlerhaft ist, beendet das Ziel den Modus des langsamen Hochfahrens. Wenn das Ziel dann fehlerfrei ist, wechselt es wieder in den Modus des langsamen Hochfahrens.
- Sie können den langsamen Startmodus nicht aktivieren, wenn Sie die wenigsten ausstehenden Anfragen oder gewichtete zufällige Routing-Algorithmen verwenden.

So aktualisieren Sie den Wert für die Dauer des langsamen Hochfahrens mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie auf der Registerkarte Gruppendetails im Abschnitt Attribute die Option Bearbeiten aus.
5. Ändern Sie auf der Seite Attribute bearbeiten den Wert von Dauer des langsamen Hochfahrens nach Bedarf. Um den Modus des langsamen Hochfahrens zu deaktivieren, legen Sie als Dauer 0 fest.
6. Wählen Sie Änderungen speichern aus.

Um den Wert für die Dauer des langsamen Starts zu aktualisieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-target-group-attributes](#) zusammen mit dem Attribut `slow_start.duration_seconds`.

Erstellen einer Zielgruppe

Sie registrieren Ihre Ziele bei einer Zielgruppe. Der Load Balancer sendet standardmäßig Anfragen an registrierte Ziele mithilfe des Ports und des Protokolls, den bzw. das Sie für die Zielgruppe angegeben haben. Sie können diesen Port überschreiben, wenn Sie jedes Ziel bei der Zielgruppe registrieren.

Nachdem Sie eine Zielgruppe erstellt haben, können Sie Tags hinzufügen.

Um Datenverkehr an die Ziele in einer Zielgruppe weiterzuleiten, geben Sie die Zielgruppe in einer Aktion an, wenn Sie einen Listener erstellen oder eine Regel für den Listener erstellen. Weitere Informationen finden Sie unter [Listener-Regeln](#). Sie können dieselbe Zielgruppe in mehreren Listenern angeben, aber diese Listener müssen demselben Application Load Balancer angehören. Um eine Zielgruppe mit einem Load Balancer zu verwenden, müssen Sie sicherstellen, dass die Zielgruppe nicht von einem Listener verwendet wird, der einem anderen Load Balancer angehört.

Sie können jederzeit Ziele zu Ihrer Zielgruppe hinzufügen oder aus dieser entfernen. Weitere Informationen finden Sie unter [Registrieren Sie Ziele mit Ihrer Zielgruppe](#). Sie können auch die Zustandsprüfungseinstellungen für Ihre Zielgruppe ändern. Weitere Informationen finden Sie unter [Ändern der Zustandsprüfungseinstellungen für eine Zielgruppe](#).

Erstellen einer Zielgruppe mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie Zielgruppe erstellen aus.
4. Wählen Sie für Zieltyp auswählen die Option Instances aus, um Ziele nach Instance-ID zu registrieren, IP-Adressen, um Ziele nach IP-Adresse zu registrieren, oder Lambda-Funktion, um eine Lambda-Funktion als Ziel zu registrieren.
5. Geben Sie im Feld Target group name (Zielgruppenname) einen Namen für die neue Zielgruppe ein. Dieser Name muss für jede Region und jedes Konto eindeutig sein, darf maximal 32 Zeichen lang sein, darf nur alphanumerische Zeichen oder Bindestriche enthalten und darf nicht mit einem Bindestrich beginnen oder enden.
6. (Optional) Ändern Sie ggf. die Standardwerte im Feld Protocol (Protokoll) und Port nach Bedarf.
7. Wenn der Zieltyp Instances oder IP-Adressen ist, wählen Sie IPv4 oder IPv6 als IP-Adresstyp aus. Fahren Sie andernfalls mit dem nächsten Schritt fort.

Beachten Sie, dass nur Ziele, die den ausgewählten IP-Adresstyp haben, in diese Zielgruppe eingefügt werden können. Der IP-Adresstyp kann nicht geändert werden, nachdem die Zielgruppe erstellt wurde.

8. Wählen Sie im Feld VPC eine Virtual Private Cloud (VPC) aus. Beachten Sie, dass für die Zieltypen von IP addresses (IP-Adressen) diejenigen VPCs zur Auswahl stehen, die den IP address type (IP-Adresstyp) unterstützen, den Sie im vorherigen Schritt ausgewählt haben.
9. (Optional) Ändern Sie den Standardwert im Feld Protokollversion nach Bedarf.

10. (Optional) Behalten Sie im Abschnitt Zustandsprüfungen die Standardeinstellungen nach Bedarf bei.
11. Wenn der Zieltyp Lambda-Funktion ist, können Sie Zustandsprüfungen aktivieren, indem Sie im Abschnitt Zustandsprüfungen die Option Aktivieren auswählen.
12. (Optional) Fügen Sie einen oder mehrere Tags wie folgt hinzu:
 - a. Erweitern Sie den Abschnitt Tags.
 - b. Wählen Sie Add tag.
 - c. Geben Sie den Tag-Schlüssel und den Tag-Wert ein.
13. Wählen Sie Weiter aus.
14. (Optional) Fügen Sie ein oder mehrere Ziele wie folgt hinzu:
 - Wenn der Zieltyp Instances ist, wählen Sie eine oder mehrere Instances aus, geben Sie einen oder mehrere Ports ein und wählen Sie dann Schließen Sie die unten angeführten als ausstehend ein aus.

Hinweis: Den Instances muss eine primäre IPv6-Adresse zugewiesen sein, um bei einer IPv6-Zielgruppe registriert zu werden.
 - Wenn der Zieltyp IP-Adressen lautet, gehen Sie wie folgt vor:
 - a. Wählen Sie eine Netzwerk-VPC in der Liste aus oder wählen Sie Andere private IP-Adresse aus.
 - b. Geben Sie die IP-Adresse manuell ein oder suchen Sie die IP-Adresse anhand der Instance-Details. Sie können bis zu fünf IP-Adressen gleichzeitig eingeben.
 - c. Geben Sie die Ports für die Weiterleitung des Datenverkehrs an die angegebenen IP-Adressen ein.
 - d. Wählen Sie Schließen Sie die unten angeführten als ausstehend ein aus.
 - Wenn der Zieltyp eine Lambda-Funktion ist, geben Sie eine einzelne Lambda-Funktion an oder lassen Sie diesen Schritt aus und geben Sie später eine Lambda-Funktion an.
15. Wählen Sie Zielgruppe erstellen aus.
16. (Optional) Sie können die Zielgruppe in einer Listener-Regel angeben. Weitere Informationen finden Sie unter [Listener-Regeln](#).

Um eine Zielgruppe mit dem zu erstellen AWS CLI

Verwenden Sie den Befehl [create-target-group](#), um die Zielgruppe zu erstellen, den Befehl [add-tags](#), um die Zielgruppe zu kennzeichnen, und den Befehl [register-targets](#), um Ziele hinzuzufügen.

Zustandsprüfungen für Ihre Zielgruppen

Ihr Application Load Balancer sendet regelmäßig Anforderungen an die registrierten Ziele, um deren Status zu überprüfen. Diese Tests werden als Zustandsprüfungen bezeichnet.

Jeder Load Balancer-Knoten leitet Anfragen nur an die betriebsbereiten Ziele in den aktivierten Availability Zones des Balancer. Jeder Load Balancer-Knoten überprüft den Zustand jedes Ziels mit den Zustandsprüfungseinstellungen für die Zielgruppen, in denen das Ziel registriert ist. Nachdem Ihr Ziel registriert wurde, muss es die Zustandsprüfung fehlerfrei bestehen, um als stabil eingestuft zu werden. Nachdem die einzelnen Zustandsprüfungen abgeschlossen wurden, schließt der Load Balancer-Knoten die Verbindung, die für die Zustandsprüfung eingerichtet wurde.

Wenn eine Zielgruppe nur fehlerhafte registrierte Ziele enthält, leitet der Load Balancer Anforderungen an all diese Ziele weiter, unabhängig von ihrem Zustandstatus. Das bedeutet, dass sich der Load Balancer nicht öffnen lässt, wenn alle Ziele gleichzeitig die Zustandsprüfungen in allen aktivierten Availability Zones nicht bestehen. Das Fail-Open hat zur Folge, dass der Datenverkehr an alle Ziele in allen aktivierten Availability Zones unabhängig von deren Zustandsstatus auf der Grundlage des Load-Balancing-Algorithmus aktiviert wird.

Gesundheitschecks werden nicht unterstützt WebSockets.

Zustandsprüfungseinstellungen

Sie können Zustandsprüfungen für die Ziele in einer Zielgruppe konfigurieren, wie in der folgenden Tabelle beschrieben. Die in der Tabelle verwendeten Einstellungsnamen sind die in der API verwendeten Namen. Der Load Balancer sendet alle HealthCheckIntervalSecondsSekunden eine Integritätsprüfanforderung an jedes registrierte Ziel. Dabei werden der angegebene Port, das Protokoll und der Pfad zur Integritätsprüfung verwendet. Jede Anfrage nach einer Zustandsprüfung ist unabhängig und das Ergebnis hält über das gesamte Intervall an. Die Zeit, die das Ziel für die Antwort benötigt, hat keinen Einfluss auf das Intervall für die nächste Anfrage zur Zustandsprüfung. Wenn die Zustandsprüfungen mehrere UnhealthyThresholdCountaufeinanderfolgende Fehler überschreiten, nimmt der Load Balancer das Ziel außer Betrieb. Wenn die Zustandsprüfungen HealthyThresholdCountaufeinanderfolgende Erfolge überschreiten, nimmt der Load Balancer das Ziel wieder in Betrieb.

Einstellung	Beschreibung
HealthCheckProtocol	<p>Das Protokoll, das der Load Balancer für die Zustandsprüfungen der Ziele verwendet. Möglichen Protokolle sind HTTP und HTTPS. Das Standardprotokoll ist HTTP.</p> <p>Diese Protokolle verwenden die HTTP-GET-Methode, um Gibt den Pfad an, um Zustandsprüfungsanforderungen zu senden.</p>
HealthCheckPort	<p>Der Port, den der Load Balancer für die Zustandsprüfungen der Ziele verwendet . Standardmäßig wird der Port verwendet, auf dem jedes Ziel Datenverkehr vom Load Balancer empfängt.</p>
HealthCheckPath	<p>Das Ziel für Zustandsprüfungen der Ziele.</p> <p>Wenn die Protokollversion HTTP/1.1 oder HTTP/2 ist, geben Sie einen gültigen URI an (/Pfad?Abfrage). Der Standardwert ist /.</p> <p>Wenn die Protokollversion gRPC ist, geben Sie den Pfad einer benutzerdefinierten Zustandsprüfungsmethode mit dem Format <code>/package.service/method</code> an. Der Standardwert ist <code>/AWS.ALB/healthcheck</code> .</p>
HealthCheckTimeoutSeconds	<p>Die Anzahl der Sekunden, in denen keine Antwort von einem Ziel bedeutet, dass die Zustandsprüfung fehlgeschlagen ist. Der Bereich liegt zwischen 2 und 120 Sekunden. Standardmäßig ist bei dem Zieltyp <code>instance</code> oder <code>ip</code> ein Zeitraum von 5 Sekunden und bei dem Zieltyp <code>lambda</code> ein Zeitraum von 30 Sekunden festgelegt.</p>

Einstellung	Beschreibung
HealthCheckIntervalSeconds	Der etwaige Zeitraum in Sekunden zwischen den Zustandsprüfungen der einzelnen Ziele. Der Bereich liegt zwischen 5 und 300 Sekunden. Standardmäßig ist bei dem Zieltyp <code>instance</code> oder <code>ip</code> ein Zeitraum von 30 Sekunden und bei dem Zieltyp <code>lambda</code> ein Zeitraum von 35 Sekunden festgelegt.
HealthyThresholdCount	Die Anzahl der aufeinanderfolgenden erfolgreichen Zustandsprüfungen, die erforderlich ist, damit ein fehlerhaftes Ziel als stabil eingestuft wird. Der Bereich liegt zwischen 2 und 10. Der Standardwert ist 5.
UnhealthyThresholdCount	Die Anzahl fortlaufender fehlgeschlagener Zustandsprüfungen, die erforderlich ist, damit ein Ziel als nicht betriebsbereit eingestuft wird. Der Bereich liegt zwischen 2 und 10. Der Standardwert ist 2.

Einstellung	Beschreibung
Matcher	<p>Die Codes, die verwendet werden, um ein Ziel auf eine erfolgreiche Antwort zu überprüfen. Diese werden in der Konsole als Erfolgscodes bezeichnet.</p> <p>Wenn die Protokollversion HTTP/1.1 oder HTTP/2 ist, liegen die möglichen Werte zwischen 200 und 499. Sie können mehrere Werte angeben (z. B. "200,202") oder einen Wertebereich (z. B. "200-299"). Der Standardwert ist 200.</p> <p>Wenn die Protokollversion gRPC ist, liegen die möglichen Werte zwischen 0 und 99. Sie können mehrere Werte angeben (z. B. "0,1") oder einen Wertebereich (z. B. "0-5"). Der Standardwert lautet 12.</p>

Zustandsstatus des Ziels

Bevor der Load Balancer eine Zustandsprüfungsanforderung an ein Ziel sendet, müssen Sie dieses Ziel in einer Zielgruppe registrieren, die Zielgruppe in einer Listener-Regel spezifizieren und sicherstellen, dass die Availability Zone des Ziels für den Load Balancer aktiviert ist. Damit ein Ziel Anforderungen vom Load Balancer erhalten kann, muss es die anfänglichen Zustandsprüfungen bestehen. Nachdem ein Ziel die anfänglichen Zustandsprüfungen bestanden hat, ist sein Status `Healthy`.

Die folgende Tabelle beschreibt die möglichen Werte für den Zustandsstatus eines registrierten Ziels.

Wert	Beschreibung
<code>initial</code>	Der Load Balancer befindet sich im Prozess der Registrierung eines Ziels oder der Durchführung der anfänglichen Zustandsprüfungen für das Ziel.

Wert	Beschreibung
	Zugehörige Ursachencodes: <code>Elb.RegistrationInProgress</code> <code>Elb.InitialHealthChecking</code>
healthy	Das Ziel ist fehlerfrei. Zugehörige Ursachencodes: Keine
unhealthy	Das Ziel hat nicht auf eine Zustandsprüfung geantwortet oder die Zustandsprüfung ist fehlgeschlagen. Zugehörige Ursachencodes: <code>Target.ResponseCodeMismatch</code> <code>Target.Timeout</code> <code>Target.FailedHealthChecks</code> <code>Elb.InternalError</code>
unused	Das Ziel wurde nicht für eine Zielgruppe registriert, die Zielgruppe wird nicht in einer Listener-Regel verwendet oder das Ziel befindet sich in einer Availability Zone, die nicht aktiviert ist, oder das Ziel sich im Status „Angehalten“ oder „Beendet“. Zugehörige Ursachencodes: <code>Target.NotRegistered</code> <code>Target.NotInUse</code> <code>Target.InvalidState</code> <code>Target.IpUnusable</code>
draining	Die Registrierung für das Ziel wird aufgehoben, und Connection Draining wird durchgeführt. Zugehöriger Ursachencode: <code>Target.DeregistrationInProgress</code>
unavailable	Die Zustandsprüfungen sind für die Zielgruppe deaktiviert. Zugehöriger Ursachencode: <code>Target.HealthCheckDisabled</code>

Ursachencodes für Zustandsprüfungen

Ist der Status eines Ziels ein anderer Wert als `Healthy`, gibt die API einen Ursachencode und eine Beschreibung des Problems zurück und die Konsole zeigt die gleiche Beschreibung an. Ursachencodes, die mit `Elb` beginnen, haben ihren Ursprung auf dem Load Balancer, und Ursachencodes, die mit `Target` beginnen, haben ihren Ursprung auf der Seite des Ziels. Weitere Informationen zu möglichen Ursachen für Fehler bei der Zustandsprüfung finden Sie unter [Fehlerbehebung](#).

Ursachencode	Beschreibung
<code>Elb.InitialHealthChecking</code>	Anfängliche Zustandsprüfungen in Bearbeitung
<code>Elb.InternalError</code>	Zustandsprüfungen aufgrund eines internen Fehlers fehlgeschlagen
<code>Elb.RegistrationInProgress</code>	Zielregistrierung wird durchgeführt
<code>Target.DeregistrationInProgress</code>	Zielregistrierung wird aufgehoben
<code>Target.FailedHealthChecks</code>	Zustandsprüfungen fehlgeschlagen
<code>Target.HealthCheckDisabled</code>	Zustandsprüfungen sind deaktiviert
<code>Target.InvalidState</code>	Ziel hat den Status „Angehalten“ Ziel hat den Status „Beendet“ Ziel hat den Status „Beendet oder Angehalten“ Ziel hat den Status „Ungültig“
<code>Target.IpUnusable</code>	Die IP-Adresse kann nicht als Ziel verwendet werden, da sie von einem Load Balancer verwendet wird.
<code>Target.NotInUse</code>	Zielgruppe ist nicht konfiguriert, um Verkehr vom Load Balancer zu erhalten

Ursachencode	Beschreibung
	Ziel ist in einer Availability Zone, die nicht für den Load Balancer aktiviert ist
Target.NotRegistered	Ziel ist nicht in der Zielgruppe registriert
Target.ResponseCodeMismatch	Zustandsprüfungen sind mit diesen Codes fehlgeschlagen: [Code]
Target.Timeout	Zeitlimit für Anforderung überschritten

Zustand der Ziele prüfen

Sie können den Zustand der Ziele, die in Ihren Zielgruppen registriert sind, überprüfen.

Überprüfen des Zustands Ihrer Ziele mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. In der Registerkarte Targets (Ziele) gibt die Spalte Status den Status der einzelnen Ziele wieder.
5. Wenn der Status einen anderen Wert als Healthy hat, enthält die Spalte Statusdetails weitere Informationen. Hilfe bei fehlgeschlagenen Zustandsprüfungen finden Sie unter [Fehlerbehebung](#).

Um den Zustand Ihrer Ziele zu überprüfen, verwenden Sie AWS CLI

Verwenden Sie den [describe-target-health](#)-Befehl. Die Ausgabe dieses Befehls enthält den Zustand des Ziels. Wenn der Status einen anderen Wert als Healthy aufweist, enthält die Ausgabe auch einen Ursachencode.

So erhalten Sie E-Mail-Benachrichtigungen über fehlerhafte Ziele

Verwenden Sie CloudWatch Alarme, um eine Lambda-Funktion auszulösen, um Details über fehlerhafte Ziele zu senden. step-by-step Anweisungen finden Sie im folgenden Blogbeitrag: [Identifizieren fehlerhafter Ziele Ihres Load Balancers](#).

Ändern der Zustandsprüfungseinstellungen für eine Zielgruppe

Sie können die Zustandsprüfungseinstellungen für Ihre Zielgruppe jederzeit ändern.

Ändern der Zustandsprüfungseinstellungen für eine Zielgruppe mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie auf der Registerkarte Gruppendetails im Abschnitt Einstellungen für die Zustandsprüfung die Option Bearbeiten aus.
5. Ändern Sie auf der Seite Einstellungen für die Zustandsprüfung bearbeiten die Einstellungen nach Bedarf und wählen Sie dann Änderungen speichern.

Um die Einstellungen für die Gesundheitsprüfung einer Zielgruppe zu ändern, verwenden Sie AWS CLI

Verwenden Sie den [modify-target-group](#)-Befehl.

Zonenübergreifendes Load Balancing für Zielgruppen

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 registrierten 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. Dies könnte verwendet werden, wenn zonale Ausfalldomänen regionalen vorzuziehen sind, um sicherzustellen, dass eine fehlerfreie Zone nicht von einer fehlerhaften Zone beeinträchtigt wird, oder um die allgemeine Latenz zu verbessern.

Bei Application Load Balancern ist der zonenübergreifende Load Balancing immer auf Load Balancer-Ebene aktiviert und kann nicht ausgeschaltet werden. Für Zielgruppen wird standardmäßig die Load-Balancer-Einstellung verwendet. Sie können die Standardeinstellung jedoch überschreiben, indem Sie den zonenübergreifenden Load Balancing auf Zielgruppenebene explizit ausschalten.

Überlegungen

- Die Zielgruppenbindung wird nicht unterstützt, wenn zonenübergreifendes Load Balancing deaktiviert ist.
- Lambda-Funktionen als Ziele werden nicht unterstützt, wenn zonenübergreifendes Load Balancing deaktiviert ist.
- Beim Versuch, das zonenübergreifende Load Balancing über die `ModifyTargetGroupAttributes`-API zu deaktivieren, wenn bei Zielen der Parameter `AvailabilityZone` auf `all` gesetzt ist, tritt ein Fehler auf.
- Bei der Registrierung von Zielen ist der `AvailabilityZone`-Parameter erforderlich. Spezifische `Availability Zone`-Werte sind nur zulässig, wenn zonenübergreifendes Load Balancing deaktiviert ist. Andernfalls wird der Parameter ignoriert und als `all` behandelt.

Bewährte Methoden

- Planen Sie für jede Zielgruppe genügend Zielkapazität für alle `Availability Zones` ein, die Sie voraussichtlich nutzen werden. Wenn Sie nicht genügend Kapazität für alle teilnehmenden `Availability Zones` einplanen können, empfehlen wir Ihnen, das zonenübergreifende Load Balancing aktiviert zu lassen.
- Wenn Sie Ihren Application Load Balancer mit mehreren Zielgruppen konfigurieren, stellen Sie sicher, dass alle Zielgruppen innerhalb der konfigurierten Region an denselben `Availability Zones` teilnehmen. Dadurch soll verhindert werden, dass eine `Availability Zone` leer ist, während das zonenübergreifende Load Balancing deaktiviert ist, da dies für alle HTTP-Anfragen, die in die leere `Availability Zone` gelangen, einen 503-Fehler auslöst.
- Vermeiden Sie das Erstellen leerer Subnetze. Application Load Balancer stellen zonale IP-Adressen über DNS für die leeren Subnetze zur Verfügung, was 503-Fehler bei HTTP-Anfragen auslöst.
- Es kann vorkommen, dass eine Zielgruppe mit deaktiviertem zonenübergreifendem Load Balancing über genügend geplante Zielkapazität pro `Availability Zone` verfügt, aber alle Ziele in einer `Availability Zone` fehlerhaft werden. Wenn es mindestens eine Zielgruppe mit ausschließlich fehlerhaften Zielen gibt, werden die IP-Adressen der Load Balancer-Knoten aus dem DNS entfernt. Sobald die Zielgruppe mindestens ein fehlerfreies Ziel hat, werden die IP-Adressen im DNS wiederhergestellt.

Wenn zonenübergreifendes Load Balancing deaktivieren

Sie können das zonenübergreifende Load Balancing für Ihre Application Load Balancer-Zielgruppen jederzeit deaktivieren.

Deaktivieren des zonenübergreifenden Load Balancing mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Zielgruppen aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Wählen Sie auf der Seite Zielgruppenattribute bearbeiten für zonenübergreifendes Load Balancing die Option Aus aus.
6. Wählen Sie Änderungen speichern.

Deaktivieren des zonenübergreifenden Load Balancing mithilfe der AWS CLI

Verwenden Sie den Befehl [modify-target-group-attributes](#) und legen Sie für das Attribut `load_balancing.cross_zone.enabled` den Wert `false` fest.

```
aws elbv2 modify-target-group-attributes --target-group-arn my-targetgroup-arn --attributes Key=load_balancing.cross_zone.enabled,Value=false
```

Nachfolgend finden Sie eine Beispielantwort:

```
{
  "Attributes": [
    {
      "Key": "load_balancing.cross_zone.enabled",
      "Value": "false"
    },
  ],
}
```

Zonenübergreifendes Load Balancing aktivieren

Sie können das zonenübergreifende Load Balancing für Ihre Application Load Balancer-Zielgruppen jederzeit aktivieren. Die Einstellung für das zonenübergreifende Load Balancing auf Zielgruppenebene hat Vorrang vor der Einstellung auf Load-Balancer-Ebene.

Aktivieren des zonenübergreifenden Load Balancing mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Zielgruppen aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Wählen Sie auf der Seite Zielgruppenattribute bearbeiten für zonenübergreifendes Load Balancing die Option An aus.
6. Wählen Sie Änderungen speichern.

Aktivieren des zonenübergreifenden Load Balancing mithilfe der AWS CLI

Verwenden Sie den Befehl [modify-target-group-attributes](#) und legen Sie für das Attribut `load_balancing.cross_zone.enabled` den Wert `true` fest.

```
aws elbv2 modify-target-group-attributes --target-group-arn my-targetgroup-arn --attributes Key=load_balancing.cross_zone.enabled,Value=true
```

Nachfolgend finden Sie eine Beispielantwort:

```
{
  "Attributes": [
    {
      "Key": "load_balancing.cross_zone.enabled",
      "Value": "true"
    },
  ]
}
```

Zustand der Zielgruppe

Standardmäßig gilt eine Zielgruppe als fehlerfrei, solange sie mindestens ein fehlerfreies Ziel hat. Wenn Sie eine große Flotte haben, reicht es nicht aus, nur ein fehlerfreies Ziel zu haben, das den Datenverkehr bereitstellt. Stattdessen können Sie eine Mindestanzahl oder einen Prozentsatz von Zielen angeben, die fehlerfrei sein müssen, und angeben, welche Aktionen der Load Balancer ergreift, wenn die fehlerfreien Ziele unter den angegebenen Schwellenwert fallen. Dies verbessert die Verfügbarkeit.

Maßnahmen bei fehlerhaftem Zustand

Sie können fehlerhafte Schwellenwerte für die folgenden Aktionen konfigurieren:

- **DNS-Failover** – Wenn die fehlerfreien Ziele in einer Zone unter den Schwellenwert fallen, markieren wir die IP-Adressen des Load-Balancer-Knotens für die Zone im DNS als fehlerhaft. Wenn Clients den DNS-Namen des Load Balancers auflösen, wird der Datenverkehr daher nur an fehlerfreie Zonen weitergeleitet.
- **Routing-Failover** – Wenn die fehlerfreien Ziele in einer Zone unter den Schwellenwert fallen, sendet der Load Balancer den Datenverkehr an alle Ziele, die für den Load-Balancer-Knoten verfügbar sind, einschließlich fehlerhafter Ziele. Dies erhöht die Wahrscheinlichkeit, dass eine Client-Verbindung erfolgreich ist, insbesondere wenn Ziele vorübergehend die Integritätsprüfungen nicht bestehen, und verringert das Risiko, dass die fehlerfreien Ziele überlastet werden.

Anforderungen und Überlegungen

- Dieses Feature kann nicht für Zielgruppen verwendet werden, bei denen das Ziel eine Lambda-Funktion ist. Wenn der Application Load Balancer das Ziel für einen Network Load Balancer oder Global Accelerator ist, konfigurieren Sie keinen Schwellenwert für ein DNS-Failover.
- Wenn Sie beide Arten von Schwellenwerten für eine Aktion angeben (Anzahl und Prozentsatz), ergreift der Load Balancer die Aktion, wenn einer der Schwellenwerte überschritten wird.
- Wenn Sie Schwellenwerte für beide Aktionen angeben, muss der Schwellenwert für DNS-Failover größer oder gleich dem Schwellenwert für Routing-Failover sein, sodass der DNS-Failover entweder mit oder vor dem Routing-Failover erfolgt.
- Wenn Sie den Schwellenwert als Prozentsatz angeben, berechnen wir den Wert dynamisch auf der Grundlage der Gesamtzahl der Ziele, die bei den Zielgruppen registriert sind.

- Die Gesamtzahl der Ziele hängt davon ab, ob zonenübergreifendes Load Balancing deaktiviert oder aktiviert ist. Wenn zonenübergreifendes Load Balancing deaktiviert ist, sendet jeder Knoten Datenverkehr nur an die Ziele in seiner eigenen Zone, was bedeutet, dass die Schwellenwerte für die Anzahl der Ziele in jeder aktivierten Zone separat gelten. Wenn zonenübergreifende Load Balancing aktiviert ist, sendet jeder Knoten Datenverkehr an alle aktivierten Ziele, was bedeutet, dass die angegebenen Schwellenwerte für die Gesamtanzahl der Ziele in allen aktivierten Zonen gelten.
- Beim DNS-Failover entfernen wir die IP-Adressen für die fehlerhaften Zonen aus dem DNS-Hostnamen für den Load Balancer. Der DNS-Cache des lokalen Clients kann diese IP-Adressen jedoch enthalten, bis die time-to-live (TTL) im DNS-Eintrag abläuft (60 Sekunden).
- Wenn ein DNS-Failover auftritt, wirkt sich dies auf alle Zielgruppen aus, die dem Load Balancer zugeordnet sind. Stellen Sie sicher, dass Sie in Ihren verbleibenden Zonen über genügend Kapazität verfügen, um diesen zusätzlichen Datenverkehr zu bewältigen, insbesondere wenn das zonenübergreifende Load Balancing deaktiviert ist.
- Wenn beim DNS-Failover alle Load-Balancer-Zonen als fehlerhaft eingestuft werden, sendet der Load Balancer Datenverkehr an alle Zonen, einschließlich der fehlerhaften Zonen.
- Neben der Frage, ob genügend fehlerfreie Ziele vorhanden sind, die zu einem DNS-Failover führen könnten, gibt es noch andere Faktoren, z. B. den Zustand der Zone.

Überwachen

Informationen zur Überwachung des Zustands Ihrer Zielgruppen finden Sie unter [CloudWatch Kennzahlen zur Zielgruppengesundheit](#).

Beispiel

Im folgenden Beispiel wird veranschaulicht, wie Zustandseinstellungen für Zielgruppen angewendet werden.

Szenario

- Ein Load Balancer, der zwei Availability Zones, A und B, unterstützt
- Jede Availability Zone enthält 10 registrierte Ziele
- Die Zielgruppe hat die folgenden Zustandseinstellungen für Zielgruppen:
 - DNS-Failover – 50 %
 - Routing-Failover – 50 %

- Sechs Ziele fallen in der Availability Zone B aus

Wenn zonenübergreifendes Load Balancing deaktiviert ist

- Der Load-Balancer-Knoten in jeder Availability Zone kann Datenverkehr nur an die 10 Ziele in seiner Availability Zone senden.
- In der Availability Zone A gibt es 10 fehlerfreie Ziele, sodass der erforderliche Prozentsatz fehlerfreier Ziele erreicht wird. Der Load Balancer verteilt weiterhin den Datenverkehr zwischen den 10 fehlerfreien Zielen.
- In der Availability Zone B gibt es nur 4 fehlerfreie Ziele, was 40 % der Ziele für den Load-Balancer-Knoten in Availability Zone B entspricht. Da dies weniger ist als der erforderliche Prozentsatz fehlerfreier Ziele, ergreift der Load Balancer die folgenden Aktionen:
 - DNS-Failover – Die Availability Zone B ist im DNS als fehlerhaft markiert. Da Clients den Load-Balancer-Namen nicht in den Load-Balancer-Knoten in Availability Zone B auflösen können und Availability Zone A fehlerfrei ist, senden Clients neue Verbindungen zur Availability Zone A.
 - Routing-Failover – Wenn neue Verbindungen explizit an Availability Zone B gesendet werden, verteilt der Load Balancer den Datenverkehr an alle Ziele in Availability Zone B, einschließlich der fehlerhaften Ziele. Dadurch werden Ausfälle bei den verbleibenden fehlerlosen Zielen verhindert.

Wenn zonenübergreifendes Load Balancing aktiviert ist

- Jeder Load-Balancer-Knoten kann Datenverkehr an alle 20 registrierten Ziele in beiden Availability Zones senden.
- Es gibt 10 fehlerfreie Ziele in Availability Zone A und 4 fehlerfreie Ziele in Availability Zone B, also insgesamt 14 fehlerfreie Ziele. Das sind 70 % der Ziele für die Load-Balancer-Knoten in beiden Availability Zones, wodurch der erforderliche Prozentsatz fehlerfreier Ziele erreicht wird.
- Der Load Balancer verteilt den Datenverkehr zwischen den 14 fehlerfreien Zielen in beiden Availability Zones.

Ändern der Zustandseinstellungen der Zielgruppe

Sie können die Zustandseinstellungen für Ihre Zielgruppe wie folgt ändern.

So ändern Sie die Zustandseinstellungen für eine Zielgruppe mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Load Balancer aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Überprüfen Sie, ob zonenübergreifendes Load Balancing aktiviert oder deaktiviert ist. Aktualisieren Sie diese Einstellung nach Bedarf, um sicherzustellen, dass Sie über genügend Kapazität verfügen, um den zusätzlichen Datenverkehr zu bewältigen, falls eine Zone ausfällt.
6. Erweitern Sie die Anforderungen an den Zustand der Zielgruppe.
7. Wir empfehlen, dass Sie als Konfigurationstyp die Option Einheitliche Konfiguration wählen, wodurch für beide Aktionen derselbe Schwellenwert festgelegt wird.
8. Führen Sie für Anforderungen für fehlerfreie Zustände einen der folgenden Schritte aus:
 - Wählen Sie Mindestanzahl fehlerfreier Ziele aus und geben Sie dann eine Zahl zwischen 1 und der maximalen Anzahl von Zielen für Ihre Zielgruppe ein.
 - Wählen Sie Mindestprozentsatz fehlerfreier Ziele und geben Sie dann eine Zahl zwischen 1 und 100 ein.
9. Wählen Sie Änderungen speichern aus.

Um die Gesundheitseinstellungen für Zielgruppen zu ändern, verwenden Sie den AWS CLI

Verwenden Sie den Befehl [modify-target-group-attributes](#). Im folgenden Beispiel wird der Schwellenwert für den fehlerfreien Zustand für beide Aktionen mit einem fehlerhaften Zustand auf 50 % festgelegt.

```
aws elbv2 modify-target-group-attributes \  
--target-group-arn arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-  
targets/73e2d6bc24d8a067 \  
--attributes  
Key=target_group_health.dns_failover.minimum_healthy_targets.percentage,Value=50 \  
  
Key=target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage,Value=50
```

Verwenden des Route-53-DNS-Failovers für Ihren Load Balancer

Wenn Sie mithilfe von Route 53 DNS-Abfragen an Ihren Load Balancer leiten, können Sie mithilfe von Route 53 auch DNS-Failover für Ihren Load Balancer konfigurieren. In einer Failover-Konfiguration prüft Route 53 die Integrität der Zielgruppen für den Load Balancer, um zu ermitteln, ob diese verfügbar sind. Wenn keine funktionsfähigen Ziele für den Load Balancer registriert sind oder der Load Balancer selbst fehlerhaft ist, leitet Route 53 den Datenverkehr an eine andere verfügbare Ressource, z. B. einen fehlerfreien Load Balancer oder eine statische Website in Amazon S3, weiter.

Beispiel: Sie haben eine Webanwendung `www.example.com` und möchten, dass redundante Instances hinter zwei Load Balancern in verschiedenen Regionen laufen. Sie möchten, dass der Datenverkehr in erster Linie auf den Load Balancer in einer Region weitergeleitet wird, und der Load Balancer in der anderen Region soll bei Ausfällen als Sicherung dienen. Wenn Sie DNS Failover konfigurieren, können Sie einen primären und einen sekundären (Sicherung) Load Balancer festlegen. Route 53 leitet den Datenverkehr direkt zum primären Load Balancer, und wenn dieser nicht verfügbar ist, wird der Datenverkehr zum sekundären Load Balancer geleitet.

Verwenden von „Zielintegrität auswerten“

- Wenn die Option „Zielintegrität auswerten“ für einen Aliaseintrag für einen Application Load Balancer auf Yes festgelegt ist, wertet Route 53 die Integrität der durch den `alias target`-Wert angegebenen Ressource aus. Für einen Application Load Balancer verwendet Route 53 die mit dem Load Balancer verknüpften Zustandsprüfungen für Zielgruppen.
- Wenn alle Zielgruppen in einem Application Load Balancer fehlerfrei sind, markiert Route 53 den Aliaseintrag als fehlerfrei. Wenn eine Zielgruppe mindestens ein gesundes Ziel enthält, ist die Zustandsprüfung für die Zielgruppe erfolgreich. Route 53 gibt dann Datensätze gemäß Ihrer Routing-Richtlinie zurück. Wenn die Failover-Routing-Richtlinie verwendet wird, gibt Route 53 den primären Datensatz zurück.
- Wenn eine der Zielgruppen in einem Application Load Balancer fehlerhaft ist, besteht der Aliaseintrag die Route-53-Zustandsprüfung (Fail-Open) nicht. Bei Verwendung von „Zielintegrität auswerten“ würde dies die Failover-Routing-Richtlinie nicht erfüllen.
- Wenn alle Zielgruppen in einem Application Load Balancer leer sind (keine Ziele), betrachtet Route 53 den Datensatz als fehlerhaft (Fail-Open). Bei Verwendung von „Zielintegrität auswerten“ würde dies die Failover-Routing-Richtlinie nicht erfüllen.

Weitere Informationen finden Sie unter [Konfigurieren von DNS-Failover](#) im Entwicklerhandbuch für Amazon Route 53.

Registrieren Sie Ziele mit Ihrer Zielgruppe

Sie registrieren Ihre Ziele bei einer Zielgruppe. Wenn Sie eine Zielgruppe erstellen, können Sie ihren Zieltyp angeben, durch den festgelegt wird, wie Sie ihre Ziele registrieren. Beispielsweise können Sie Instance-IDs, IP-Adressen oder Lambda-Funktionen registrieren. Weitere Informationen finden Sie unter [Zielgruppen für Ihre Application Load Balancer](#).

Wenn die Nachfrage nach Ihren aktuell registrierten Zielen steigt, können Sie zusätzliche Ziele registrieren, um die Nachfrage zu bewältigen. Wenn Ihr Ziel für die Verarbeitung von Anfragen bereit ist, registrieren Sie es bei Ihrer Zielgruppe. Der Load Balancer beginnt, Anfragen an das Ziel weiterzuleiten, sobald der Registrierungsvorgang abgeschlossen ist und das Ziel die ersten Zustandsprüfungen bestanden hat.

Wenn die Nachfrage nach Ihren registrierten Zielen sinkt oder Sie ein Ziel warten müssen, können Sie dessen Registrierung bei Ihrer Zielgruppe aufheben. Der Load Balancer stoppt das Weiterleiten von Anfragen an ein Ziel, sobald Sie die Registrierung des Ziels aufheben. Wenn das Ziel für den Empfang von Anfragen bereit ist, können Sie es wieder bei der Zielgruppe registrieren.

Wenn Sie die Registrierung eines Ziels aufheben, wartet der Load Balancer, bis laufende Anfragen abgeschlossen wurden. Dies wird als Connection Draining bezeichnet. Der Status eines Ziels ist `draining`, während Connection Draining erfolgt.

Wenn Sie die Registrierung eines Ziels aufheben, das durch IP-Adresse registriert war, müssen Sie die Verzögerung der Registrierungsaufhebung abwarten, bevor Sie dieselbe IP-Adresse erneut registrieren können.

Wenn Sie Ziele nach Instance-ID registrieren, können Sie Ihren Load Balancer mit einer Auto-Scaling-Gruppe verwenden. Nachdem Sie eine Zielgruppe einer Auto-Scaling-Gruppe zugeordnet haben und die Gruppe hochskaliert wird, werden die von der Auto-Scaling-Gruppe gestarteten Instances automatisch bei der Zielgruppe registriert. Wenn Sie die Zielgruppe von der Auto-Scaling-Gruppe trennen, wird die Registrierung der Instances bei der Zielgruppe automatisch aufgehoben. Weitere Informationen finden Sie unter [Anhängen eines Load Balancers an Ihre Auto-Scaling-Gruppe](#) im Benutzerhandbuch zu Amazon EC2 Auto Scaling.

Zielsicherheitsgruppen

Wenn Sie EC2-Instances als Ziele registrieren, müssen Sie sicherstellen, dass die Sicherheitsgruppen für Ihre Instances zulassen, dass der Load Balancer auf dem Listener-Port und dem Zustandsprüfungs-Port mit Ihren Instances kommuniziert.

Empfohlene Regeln

Inbound

Source	Port Range	Comment
<i>Load Balancer-Sicherheitsgruppe</i>	<i>Instance-Listener</i>	Datenverkehr vom Load Balancer auf dem Listener-Port der Instance zulassen
<i>Load Balancer-Sicherheitsgruppe</i>	<i>Zustandsprüfung</i>	Datenverkehr vom Load Balancer auf dem Zustandsprüfungs-Port zulassen

Außerdem sollten Sie eingehenden ICMP-Datenverkehr zur Unterstützung von Path MTU Discovery erlauben. Weitere Informationen finden Sie unter [Path MTU Discovery](#) im Amazon EC2 EC2-Benutzerhandbuch.

Gemeinsam genutzte Subnetze

Teilnehmer können einen Application Load Balancer in einer gemeinsam genutzten VPC erstellen. Teilnehmer können kein Ziel registrieren, das in einem Subnetz ausgeführt wird, das nicht für sie freigegeben ist.

Registrieren oder Aufheben der Registrierung von Zielen

Der Zieltyp der Zielgruppe legt fest, wie Sie Ziele bei dieser Zielgruppe registrieren. Weitere Informationen finden Sie unter [Zieltyp](#).

Inhalt

- [Ziele nach Instance-ID registrieren oder die Registrierung aufheben](#)
- [Ziele nach IP-Adresse registrieren oder die Registrierung aufheben](#)
- [Registrieren und Aufheben der Registrierung einer Lambda-Funktion](#)
- [Registrieren oder Aufheben der Registrierung von Zielen mithilfe der AWS CLI](#)

Ziele nach Instance-ID registrieren oder die Registrierung aufheben

Note

Bei der Registrierung von Zielen anhand der Instance-ID für eine IPv6-Zielgruppe müssen die Ziele über eine zugewiesene primäre IPv6-Adresse verfügen. Weitere Informationen finden Sie unter [IPv6-Adressen](#) im Amazon EC2 EC2-Benutzerhandbuch

Die Instance muss sich in der Virtual Private Cloud (VPC) befinden, die Sie für die Zielgruppe angegeben haben. Die Instance muss sich auch im Status `running` befinden, wenn Sie sie registrieren.

So verfahren Sie zum Registrieren oder Aufheben der Registrierung von Zielen nach Instance-ID mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie die Registerkarte Ziele.
5. Um Instances zu registrieren, wählen Sie Ziele registrieren. Wählen Sie eine oder mehrere Instances aus, geben Sie bei Bedarf den Instance-Standardport ein und wählen Sie dann Schließen Sie die unten angeführten als ausstehend ein aus. Wenn Sie mit dem Hinzufügen der Instances fertig sind, wählen Sie Ausstehende Ziele registrieren aus.

Hinweis:

- Den Instances muss eine primäre IPv6-Adresse zugewiesen sein, um bei einer IPv6-Zielgruppe registriert zu werden.
 - AWS GovCloud (US) Region en unterstützen die Zuweisung einer primären IPv6-Adresse über die Konsole nicht. Sie müssen die API verwenden, um primäre IPv6-Adressen in s zuzuweisen. AWS GovCloud (US) Region
6. Um die Registrierung von Instances aufzuheben, wählen Sie die Instances aus und klicken Sie dann auf Abmelden.

Ziele nach IP-Adresse registrieren oder die Registrierung aufheben

IPv4-Ziele

Die IP-Adressen, die Sie registrieren, müssen aus einem der folgenden CIDR-Blöcke stammen:

- Die Subnetze der VPC für die Zielgruppe
- 10.0.0.0/8 (RFC 1918)
- 100.64.0.0/10 (RFC 6598)
- 172.16.0.0/12 (RFC 1918)
- 192.168.0.0/16 (RFC 1918)

Es ist nicht möglich, IP-Adressen eines anderen Application Load Balancers in derselben VPC zu registrieren. Wenn der andere Application Load Balancer sich in einer VPC befindet, die durch Peering mit dem Load Balancer verbunden ist, können Sie die IP-Adressen registrieren.

IPv6-Ziele

- Die IP-Adressen, die Sie registrieren, müssen sich im VPC-CIDR-Block oder in einem Peer-VPC-CIDR-Block befinden.

So verfahren Sie zum Registrieren oder Aufheben der Registrierung von Zielen nach IP-Adresse mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie die Registerkarte Ziele.
5. Um IP-Adressen zu registrieren, wählen Sie Ziele registrieren. Wählen Sie für jede IP-Adresse das Netzwerk, die IP-Adresse und den Port aus und wählen Sie dann Schließen Sie die unten angeführten als ausstehend ein aus. Wenn Sie die Eingabe der Adressen abgeschlossen haben, wählen Sie Ausstehende Ziele registrieren.
6. Um die Registrierung von IP-Adressen aufzuheben, wählen Sie die IP-Adressen aus und klicken Sie dann auf Abmelden. Wenn Sie viele registrierte IP-Adressen haben, können Sie einen Filter hinzufügen oder die Sortierreihenfolge ändern.

Registrieren und Aufheben der Registrierung einer Lambda-Funktion

Sie können für jede Zielgruppe eine einzelne Lambda-Funktion registrieren. Elastic Load Balancing benötigt Berechtigungen zum Aufrufen der Lambda-Funktion. Wenn Sie zu Ihrer Lambda-Funktion keinen Datenverkehr mehr senden müssen, können Sie ihre Registrierung aufheben. Nachdem Sie die Registrierung einer Lambda-Funktion aufgehoben haben, schlagen laufende Anfragen mit HTTP-5XX-Fehlermeldungen fehl. Zum Ersetzen einer Lambda-Funktion ist es daher besser, stattdessen eine neue Zielgruppe zu erstellen. Weitere Informationen finden Sie unter [Lambda-Funktionen als Ziele](#).

So registrieren oder deregistrieren Sie eine Lambda-Funktion mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie die Registerkarte Ziele.
5. Wenn keine Lambda-Funktion registriert ist, wählen Sie Register (Registrieren). Wählen Sie die Lambda-Funktion und danach Register (Registrieren) aus.
6. Um die Registrierung einer Lambda-Funktion aufzuheben, wählen Sie Deregister (Registrierung aufheben). Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Deregister aus.

Registrieren oder Aufheben der Registrierung von Zielen mithilfe der AWS CLI

Verwenden Sie den Befehl [register-targets](#) zum Hinzufügen von Zielen und den Befehl [deregister-targets](#) zum Entfernen von Zielen.

Sticky Sessions für Ihren Application Load Balancer

Standardmäßig leitet ein Application Load Balancer jede Anforderung getrennt an ein registriertes Ziel weiter, basierend auf dem ausgewählten Load-Balancing-Algorithmus. Sie können jedoch das Feature „Sticky Session“ (auch als gebundene Sitzungen bezeichnet) verwenden, damit der Load Balancer die Sitzung eines Benutzers an ein bestimmtes Ziel binden kann. So wird sichergestellt, dass alle Anforderungen, die während der Sitzung vom Benutzer gesendet werden, an dasselbe Ziel weitergeleitet werden. Dieses Feature ist nützlich für Server, die Zustandsinformationen verwalten, um Clients eine kontinuierliche Erfahrung zu bieten. Um Sticky Sessions zu verwenden, muss der Client Cookies unterstützen.

Application Load Balancer unterstützen sowohl dauerbasierte Cookies als auch anwendungsbasierte Cookies. Sticky Sessions werden auf Ebene der Zielgruppe aktiviert. Sie können für Ihre Zielgruppen eine Kombination aus auf Dauer basierenden Sticky Sessions, anwendungsbasierten Sticky Sessions und keine Sticky Sessions verwenden.

Bei der Verwaltung von Sticky Sessions ist es besonders wichtig festzulegen, wie lange der Load Balancer die Anforderung des Benutzers an das gleiche Ziel leiten soll. Wenn Ihre Anwendung über ein eigenes Sitzungscookie verfügt, können Sie anwendungsbasierte Sticky Session verwenden. Das Sitzungscookie der Load-Balancer-Sitzung hält dann die durch das Sitzungscookie der Anwendung festgelegte Dauer ein. Wenn Ihre Anwendung kein eigenes Sitzungscookie hat, können Sie auf Dauer basierende Sticky Sessions verwenden, um ein Load-Balancer-Sitzungscookie mit einer von Ihnen angegebenen Dauer zu generieren.

Der Inhalt der von Load Balancern generierten Cookies wird mit einem rotierenden Schlüssel verschlüsselt. Sie können von Load Balancern generierte Cookies nicht entschlüsseln oder ändern.

Bei beiden Stickiness-Typen setzt der Application Load Balancer den Ablauf der von ihm generierten Cookies nach jeder Anforderung zurück. Wenn ein Cookie abläuft, ist die Sitzung keine Sticky Session mehr und der Client sollte das Cookie aus seinem Cookiespeicher entfernen.

Voraussetzungen

- Ein HTTP/HTTPS-Load Balancer.
- Mindestens eine funktionierende Instance in jeder Availability Zone.

Überlegungen

- Sticky Sessions werden nicht unterstützt, wenn [zonenübergreifendes Load Balancing](#) deaktiviert ist. Der Versuch, Sticky Sessions zu aktivieren, während zonenübergreifendes Load Balancing deaktiviert ist, scheitert.
- Bei anwendungsbasierten Cookies müssen die Namen der Cookies für jede Zielgruppe einzeln angegeben werden. Bei dauerbasierten Cookies ist AWSALB jedoch der einzige Name, der für alle Zielgruppen verwendet wird.
- Wenn Sie mehrere Ebenen von Application Load Balancern nutzen, können Sie mit anwendungsbasierten Cookies Sticky Sessions auf allen Ebenen aktivieren. Mit dauerbasierten Cookies können Sie Sticky Sessions jedoch nur auf einer Ebene aktivieren, weil AWSALB der einzige verfügbare Name ist.
- Anwendungs-basierte Sticky Sessions funktionieren nicht bei gewichteten Zielgruppen.

- Wenn Sie über eine [Weiterleitungsregel](#) mit mehreren Zielgruppen verfügen und für mindestens eine Sticky Sessions aktiviert sind, müssen Sie die Stickiness der Zielgruppe aktivieren.
- WebSocket Verbindungen sind von Natur aus klebrig. Wenn der Client ein Verbindungs-Upgrade für anfordert WebSockets, ist das in der Verbindung verwendete Ziel das Ziel, das einen HTTP-101-Statuscode zurückgibt, um das WebSockets Verbindungs-Upgrade zu akzeptieren. Nach Abschluss des WebSockets Upgrades wird die auf Cookies basierende Stickiness nicht mehr verwendet.
- Application Load Balancer verwenden das Expires-Attribut im Cookie-Header anstelle des Max-Age-Attributs.
- Application Load Balancer unterstützen keine Cookie-Werte, die URL-codiert sind.

Sticky Sessions auf Basis der Dauer

Bei auf Dauer basierenden Sticky Sessions werden Anforderungen mithilfe eines vom Load Balancer generierten Cookies (AWSALB) an dasselbe Ziel in einer Zielgruppe weitergeleitet. Das Cookie wird verwendet, um die Sitzung dem Ziel zuzuordnen. Wenn Ihre Anwendung kein eigenes Sitzungscookie hat, können Sie Ihre eigene Dauer der Sticky Sessions angeben und festlegen, wie lange Ihr Load Balancer die Anforderung des Benutzers konsistent an dasselbe Ziel weiterleiten soll.

Wenn ein Load Balancer eine Anforderung von einem Client erhält, leitet er die Anforderung (basierend auf der Grundlage des ausgewählten Algorithmus) an ein Ziel weiter und generiert ein Cookie namens AWSALB. Er codiert Informationen über das ausgewählte Ziel, verschlüsselt das Cookie und schließt das Cookie in die Antwort an den Client ein. Das vom Load Balancer generierte Cookie hat eine eigene Dauer der Gültigkeit von 7 Tagen. Dieses Ablaufdatum ist nicht konfigurierbar.

Bei nachfolgenden Anforderungen sollte der Client das AWSALB-Cookie enthalten. Wenn der Load Balancer eine Anforderung von einem Client erhält, die das Cookie enthält, erkennt er es und leitet die Anforderung an dasselbe Ziel weiter. Wenn das Cookie vorhanden ist, aber nicht entschlüsselt werden kann, oder wenn es sich auf ein Ziel bezieht, das abgemeldet wurde oder fehlerhaft ist, wählt der Load Balancer ein neues Ziel aus und aktualisiert das Cookie mit Informationen zum neuen Ziel.

Für CORS-Anfragen (Cross-Origin Resource Sharing) müssen einige Browser Stickiness aktivieren. SameSite=None; Secure Um diese Browser zu unterstützen, generiert der Load Balancer immer ein zweites Stickiness-Cookie AWSALBCORS, das dieselben Informationen wie das ursprüngliche Stickiness-Cookie sowie das Attribut enthält. SameSite Kunden erhalten beide Cookies, auch Anfragen, die nicht von CORS stammen.

So aktivieren Sie auf Dauer basierende Sticky Sessions mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie auf der Registerkarte Gruppendetails im Abschnitt Attribute die Option Bearbeiten aus.
5. Führen Sie auf der Seite Edit attributes die folgenden Schritte aus:
 - a. Wählen Sie Stickiness aus.
 - b. Wählen Sie für den Stickiness-Typ die Option Vom Load Balancer generiertes Cookie aus.
 - c. Geben Sie im Feld Stickiness duration(Erhaltungsdauer) einen Wert zwischen 1 Sekunde und 7 Tagen aus.
 - d. Wählen Sie Änderungen speichern aus.

Um die dauerabhängige Klebrigkeit zu aktivieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-target-group-attributes](#) zusammen mit den Attributen `stickiness.enabled` und `stickiness.lb_cookie.duration_seconds`.

Verwenden Sie den folgenden Befehl, um auf Dauer basierende Sticky Sessions zu aktivieren.

```
aws elbv2 modify-target-group-attributes --target-group-arn ARN --attributes
Key=stickiness.enabled,Value=true
Key=stickiness.lb_cookie.duration_seconds,Value=time-in-seconds
```

Die Ausgabe sollte in etwa wie folgt aussehen.

```
{
  "Attributes": [
    ...
    {
      "Key": "stickiness.enabled",
      "Value": "true"
    },
    {
      "Key": "stickiness.lb_cookie.duration_seconds",
      "Value": "86500"
    }
  ]
}
```

```
    },  
    ...  
  ]  
}
```

Anwendungsbasierte Sticky Sessions

Anwendungsbasierte Sticky Sessions geben Ihnen die Flexibilität, Ihre eigenen Kriterien für die Client-Ziel-Stickiness festzulegen. Wenn Sie die anwendungsbasierte Stickiness aktivieren, leitet der Load Balancer die erste Anforderung auf der Grundlage des ausgewählten Algorithmus an ein Ziel innerhalb der Zielgruppe weiter. Es wird erwartet, dass das Ziel ein benutzerdefiniertes Anwendungscookie setzt, das dem im Load Balancer konfigurierten Cookie entspricht, um Stickiness zu aktivieren. Dieses benutzerdefinierte Cookie kann jedes Cookie-Attribut enthalten, das von der Anwendung benötigt wird.

Wenn der Application Load Balancer das benutzerdefinierte Anwendungscookie vom Ziel empfängt, generiert er automatisch ein neues verschlüsseltes Anwendungscookie zum Erfassen von Stickiness-Informationen. Dieses vom Load Balancer generierte Anwendungscookie erfasst Stickiness-Informationen für jede Zielgruppe, für die anwendungsbasierte Sticky Sessions aktiviert sind.

Das vom Load Balancer generierte Anwendungscookie kopiert nicht die Attribute des vom Ziel gesetzten benutzerdefinierten Cookies. Es hat eine eigene Dauer der Gültigkeit von 7 Tagen. Dieses Ablaufdatum ist nicht konfigurierbar. In der Antwort an den Client validiert der Application Load Balancer nur den Namen, mit dem das benutzerdefinierte Cookie auf Zielgruppenebene konfiguriert wurde, und nicht den Wert oder das Ablaufattribut des benutzerdefinierten Cookies. Solange der Name übereinstimmt, sendet der Load Balancer in der Antwort beide Cookies an den Client: das vom Ziel gesetzte benutzerdefinierte Cookie und das vom Load Balancer generierte Anwendungscookie.

Bei nachfolgenden Anforderungen müssen die Clients beide Cookies zurücksenden, um die Stickiness aufrechtzuerhalten. Der Load Balancer entschlüsselt das Anwendungscookie und prüft, ob die konfigurierte Dauer der Stickiness noch gültig ist. Anschließend werden die im Cookie enthaltenen Informationen verwendet, um die Anforderung an dasselbe Ziel innerhalb der Zielgruppe zu senden, um die Stickiness aufrechtzuerhalten. Der Load Balancer leitet das benutzerdefinierte Anwendungscookie auch über einen Proxy an das Ziel weiter, ohne es zu überprüfen oder zu ändern. In nachfolgenden Antworten werden der Ablauf des vom Load Balancer generierten Anwendungs-Cookies und die im Load Balancer konfigurierte Dauer der Stickiness zurückgesetzt. Um die Stickiness zwischen Client und Target aufrechtzuerhalten, sollten das Cookie und die Dauer der Stickiness nicht ablaufen.

Wenn ein Ziel ausfällt oder fehlerhaft ist, leitet der Load Balancer keine Anforderungen mehr an dieses Ziel weiter und wählt basierend auf dem ausgewählten Load Balancing-Algorithmus ein neues fehlerfreies Ziel aus. Der Load Balancer behandelt die Sitzung jetzt als dem neuen fehlerfreien Ziel „angeheftet“ und leitet Anforderungen auch dann an dieses neue fehlerfreie Ziel, wenn das fehlerhafte Ziel wieder funktionsfähig ist.

Bei CORS-Anforderungen (Cross-Origin Resource Sharing) fügt der Load Balancer die `SameSite=None; Secure`-Attribute nur dann dem vom Load Balancer generierten Anwendungs-Cookie hinzu, um Stickiness zu aktivieren, wenn die Benutzeragent-Version Chromium80 oder höher ist.

Da für die meisten Browser in Bezug auf Cookies eine Größenbeschränkung von 4 KB gilt, unterteilt der Load Balancer Anwendungs-Cookies, die größer als 4 KB sind, in mehrere Cookies. Application Load Balancer unterstützen Cookies mit einer Größe von bis zu 16 KB und können daher bis zu 4 Shards erstellen, die an den Client gesendet werden. Der Name des Anwendungscookies, den der Client sieht, beginnt mit „AWSALBAPP-“ und enthält eine Fragmentnummer. Wenn die Größe des Cookies beispielsweise 0-4K ist, sieht der Client AWSALBAPP -0. Wenn die Größe des Cookies 4—8 KB beträgt, sieht der Client AWSALBAPP -0 und AWSALBAPP -1 und so weiter.

So aktivieren Sie anwendungsbasierte Sticky Sessions mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie auf der Registerkarte Gruppendetails im Abschnitt Attribute die Option Bearbeiten aus.
5. Führen Sie auf der Seite Edit attributes die folgenden Schritte aus:
 - a. Wählen Sie Stickiness aus.
 - b. Wählen Sie für den Stickiness-Typ die Option Anwendungsbasiertes Cookie aus.
 - c. Geben Sie im Feld Stickiness duration(Erhaltungsdauer) einen Wert zwischen 1 Sekunde und 7 Tagen aus.
 - d. Geben Sie unter App-Cookie-Name einen Namen für Ihr anwendungsbasiertes Cookie ein.

Verwenden Sie nicht AWSALB, AWSALBAPP oder AWSALBTG für den Cookie-Namen. Sie sind für die Verwendung durch den Load Balancer reserviert.

- e. Wählen Sie Änderungen speichern aus.

Um anwendungsbasierte Klebrigkeit zu aktivieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-target-group-attributes](#) mit den folgenden Attributen:

- `stickiness.enabled`
- `stickiness.type`
- `stickiness.app_cookie.cookie_name`
- `stickiness.app_cookie.duration_seconds`

Verwenden Sie den folgenden Befehl, um anwendungsbasierte Sticky Sessions zu aktivieren.

```
aws elbv2 modify-target-group-attributes --target-group-arn ARN --attributes
Key=stickiness.enabled,Value=true Key=stickiness.type,Value=app_cookie
Key=stickiness.app_cookie.cookie_name,Value=my-cookie-name
Key=stickiness.app_cookie.duration_seconds,Value=time-in-seconds
```

Die Ausgabe sollte in etwa wie folgt aussehen.

```
{
  "Attributes": [
    ...
    {
      "Key": "stickiness.enabled",
      "Value": "true"
    },
    {
      "Key": "stickiness.app_cookie.cookie_name",
      "Value": "MyCookie"
    },
    {
      "Key": "stickiness.type",
      "Value": "app_cookie"
    },
    {
      "Key": "stickiness.app_cookie.duration_seconds",
      "Value": "86500"
    }
  ]
}
```

```
    },  
    ...  
  ]  
}
```

Manuelle Neuverteilung

Wenn beim Hochskalieren die Anzahl der Ziele erheblich zunimmt, besteht die Gefahr einer ungleichmäßigen Lastverteilung aufgrund von Stickiness. In diesem Szenario können Sie die Last mithilfe der folgenden zwei Optionen neu auf Ihre Ziele verteilen:

- Legen Sie für das von der Anwendung generierte Cookie ein Ablaufdatum fest, das vor dem aktuellen Datum und der aktuellen Uhrzeit liegt. Dadurch wird verhindert, dass Clients das Cookie an den Application Load Balancer senden, der den Prozess der Herstellung von Stickiness neu startet.
- Legen Sie für die Konfiguration der anwendungsbasierten Stickiness des Load Balancers eine sehr kurze Dauer fest, z. B. 1 Sekunde. Dadurch wird der Application Load Balancer gezwungen, die Stickiness wiederherzustellen, auch wenn das vom Ziel gesetzte Cookie nicht abgelaufen ist.

Lambda-Funktionen als Ziele

Sie können Ihre Lambda-Funktionen als Ziele registrieren und eine Listener-Regel für das Weiterleiten von Anfragen an die Zielgruppe für Ihre Lambda-Funktion konfigurieren. Wenn der Load Balancer die Anfrage an eine Zielgruppe mit einer Lambda-Funktion als Ziel weiterleitet, ruft er Ihre Lambda-Funktion auf und übergibt den Inhalt der Anfrage im JSON-Format an die Lambda-Funktion.

Einschränkungen

- Die Lambda-Funktion und die Zielgruppe müssen sich im gleichen Konto und in der gleichen Region befinden.
- Der Anfragetext, den Sie an eine Lambda-Funktion senden können, darf maximal 1 MB betragen. Entsprechende Größenbeschränkungen finden Sie unter [HTTP-Header-Limits](#).
- Die Lambda-Funktion darf als Antwort-JSON maximal 1 MB senden.
- WebSockets werden nicht unterstützt. Upgrade-Anfragen werden mit dem HTTP 400-Code abgelehnt.

- Local Zones werden nicht unterstützt.
- Automatic Target Weights (ATW) wird nicht unterstützt.

Inhalt

- [Vorbereiten der Lambda-Funktion](#)
- [Erstellen Sie einer Zielgruppe für die Lambda-Funktion](#)
- [Empfangen von Ereignissen vom Load Balancer](#)
- [Antwort an den Load Balancer](#)
- [Header mit mehreren Werten](#)
- [Aktivieren von Zustandsprüfungen](#)
- [Aufheben der Registrierung der Lambda-Funktion](#)

Eine Demo finden Sie unter [Lambda-Ziel im Application Load Balancer](#).

Vorbereiten der Lambda-Funktion

Die folgenden Empfehlungen gelten, wenn Sie Ihre Lambda-Funktion mit einem Application Load Balancer verwenden.

Berechtigungen zum Aufrufen der Lambda-Funktion

Wenn Sie die Zielgruppe erstellen und die Lambda-Funktion mithilfe der AWS Management Console registrieren, fügt die Konsole in Ihrem Namen die erforderlichen Berechtigungen zu Ihrer Lambda-Funktionsrichtlinie hinzu. Andernfalls müssen Sie, nachdem Sie die Zielgruppe erstellt und die Funktion mit dem registriert haben AWS CLI, den Befehl [add-permission](#) verwenden, um Elastic Load Balancing die Berechtigung zum Aufrufen Ihrer Lambda-Funktion zu erteilen. Es wird empfohlen, die `aws:SourceAccount-` und `aws:SourceArn-`Bedingungsschlüssel zum Einschränken des Funktionsaufrufs an die angegebene Zielgruppe zu verwenden. Weitere Informationen finden Sie unter [Das Problem des verwirrten Stellvertreters](#) im IAM-Benutzerhandbuch.

```
aws lambda add-permission \  
--function-name lambda-function-arn-with-alias-name \  
--statement-id elb1 \  
--principal elasticloadbalancing.amazonaws.com \  
--action lambda:InvokeFunction \  
--source-arn target-group-arn \  

```

```
--source-account target-group-account-id
```

Versionsverwaltung der Lambda-Funktion

Sie können eine Lambda-Funktion pro Zielgruppe registrieren. Um sicherzustellen, dass Sie Ihre Lambda-Funktion ändern können und dass der Load Balancer immer die aktuelle Version der Lambda-Funktion aufruft, erstellen Sie einen Funktionsalias und schließen Sie den Alias in den Funktions-ARN ein, wenn Sie die Lambda-Funktion bei dem Load Balancer registrieren. Weitere Informationen finden Sie unter [Versionsverwaltung und Aliasse der AWS Lambda -Funktion](#) und [Verschiebung von Datenverkehr mithilfe von Aliassen](#) im AWS Lambda -Entwicklerhandbuch.

Funktions-Timeout

Der Load Balancer wartet, bis die Lambda-Funktion reagiert oder eine Zeitüberschreitung auftritt. Es wird empfohlen, die Zeitüberschreitung der Lambda-Funktion auf der Grundlage Ihrer erwarteten Laufzeit zu konfigurieren. Weitere Informationen zum Standard-Timeout-Wert und wie er geändert wird, finden Sie unter [Grundlegende AWS Lambda -Funktionskonfiguration](#). Weitere Informationen über den maximalen Timeout-Wert, den Sie konfigurieren können, finden Sie unter [Limits für AWS Lambda](#).

Erstellen Sie einer Zielgruppe für die Lambda-Funktion

Erstellen Sie eine Zielgruppe, die bei der Weiterleitung von Anforderungen verwendet wird. Wenn der Inhalt der Anfrage mit einer Listener-Regel mit einer Aktion für ihre Weiterleitung an diese Zielgruppe übereinstimmt, ruft der Load Balancer die registrierte Lambda-Funktion auf.

Um eine Zielgruppe zu erstellen und die Lambda-Funktion über die Konsole zu registrieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie Zielgruppe erstellen aus.
4. Wählen Sie unter Zieltyp auswählen die Option Lambda-Funktion aus.
5. Geben Sie im Feld Target group name (Zielgruppenname) einen Namen für die neue Zielgruppe ein.
6. (Optional) Um Zustandsprüfungen zu aktivieren, wählen Sie Aktivieren im Abschnitt Zustandsprüfungen aus.
7. (Optional) Fügen Sie einen oder mehrere Tags wie folgt hinzu:

- a. Erweitern Sie den Abschnitt Tags.
 - b. Wählen Sie Add tag.
 - c. Geben Sie den Tag-Schlüssel und den Tag-Wert ein.
8. Wählen Sie Weiter aus.
 9. Geben Sie eine einzelne Lambda-Funktion an oder lassen Sie alternativ diesen Schritt aus und geben Sie später eine Lambda-Funktion an.
 10. Wählen Sie Zielgruppe erstellen aus.

So erstellen und registrieren Sie eine Zielgruppe und registrieren die Lambda-Funktion mithilfe der AWS CLI

Verwenden Sie die Befehle [create-target-group](#) und [register-targets](#).

Empfangen von Ereignissen vom Load Balancer

Der Load Balancer unterstützt den Lambda-Aufruf für Anfragen über HTTP und HTTPS. Der Load Balancer sendet ein Ereignis im JSON-Format. Der Load Balancer fügt zu jeder Anfrage die folgenden Header hinzu: `X-Amzn-Trace-Id`, `X-Forwarded-For`, `X-Forwarded-Port` und `X-Forwarded-Proto`.

Wenn der `content-encoding`-Header vorhanden ist, codiert der Load Balancer mit Base64 den Text und setzt `isBase64Encoded` auf `true`.

Wenn der `content-encoding`-Header nicht vorhanden ist, hängt die Base64-Codierung vom Inhaltstyp ab. Für die folgenden Typen sendet der Load Balancer den Text unverändert und stellt `isBase64Encoded` auf `false` ein: `text/*`, `application/json`, `application/javascript` und `application/xml`. Andernfalls codiert der Load Balancer den Text mit Base64 und stellt `isBase64Encoded` auf `true` ein.

Es folgt ein Beispiereignis.

```
{
  "requestContext": {
    "elb": {
      "targetGroupArn":
        "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-target-
        group/6d0ecf831eec9f09"
```

```

    }
  },
  "httpMethod": "GET",
  "path": "/",
  "queryStringParameters": {parameters},
  "headers": {
    "accept": "text/html,application/xhtml+xml",
    "accept-language": "en-US,en;q=0.8",
    "content-type": "text/plain",
    "cookie": "cookies",
    "host": "lambda-846800462-us-east-2.elb.amazonaws.com",
    "user-agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6)",
    "x-amzn-trace-id": "Root=1-5bdb40ca-556d8b0c50dc66f0511bf520",
    "x-forwarded-for": "72.21.198.66",
    "x-forwarded-port": "443",
    "x-forwarded-proto": "https"
  },
  "isBase64Encoded": false,
  "body": "request_body"
}

```

Antwort an den Load Balancer

Die Antwort von Ihrer Lambda-Funktion muss den Base64-codierten Status, Statuscode und die Header beinhalten. Sie können den Text weglassen.

Um den binären Inhalt in den Text der Antwort einzuschließen, müssen Sie den Inhalt mit Base64 codieren und `isBase64Encoded` auf `true` einstellen. Der Load Balancer decodiert die Inhalte, um den binären Inhalt abzurufen, und sendet ihn im Text der HTTP-Antwort zum Client.

Der Load Balancer berücksichtigt keine hop-by-hop Header wie `oder. Connection Transfer-Encoding`. Sie können den `Content-Length-Header` weglassen, da der Load Balancer ihn berechnet, bevor er Antworten an Clients sendet.

Nachfolgend finden Sie eine Beispielantwort von einer auf `nodejs` basierenden Lambda-Funktion.

```

{
  "isBase64Encoded": false,
  "statusCode": 200,
  "statusDescription": "200 OK",
  "headers": {
    "Set-cookie": "cookies",

```

```
    "Content-Type": "application/json"
  },
  "body": "Hello from Lambda (optional)"
}
```

Informationen zu Lambda-Funktionsvorlagen, die mit Application Load Balancern funktionieren, finden Sie unter [application-load-balancer-serverless-app](#) auf github. Öffnen Sie alternativ die [Lambda-Konsole](#), wählen Sie Anwendungen, Anwendung erstellen und dann eine der folgenden Optionen aus: AWS Serverless Application Repository

- ALB-Lambda-Ziel-S3 UploadFileto
- ALB-Lambda-Ziel- BinaryResponse
- ALB-Lambda-Ziel-IP WhatisMy

Header mit mehreren Werten

Wenn Anfragen von einem Client oder Antworten von einer Lambda-Funktion Header mit mehreren Werten, denselben Header mehrmals oder Abfrageparameter mit mehreren Werten für den gleichen Schlüssel enthalten, können Sie Unterstützung für die Syntax von Headern mit mehreren Werten aktivieren. Nach dem Aktivieren von Headern mit mehreren Werten werden in den Headern und Abfrageparametern, die zwischen dem Load Balancer und der Lambda-Funktion ausgetauscht werden, Arrays anstelle von Zeichenfolgen verwendet. Wenn Sie die Syntax für Header mit mehreren Werten nicht aktivieren und ein Header oder Abfrageparameter mehrere Werte aufweist, verwendet der Load Balancer den zuletzt empfangenen Wert.

Inhalt

- [Anfragen mit Headern mit mehreren Werten](#)
- [Antworten mit Headern mit mehreren Werten](#)
- [Aktivieren von Headern mit mehreren Werten](#)

Anfragen mit Headern mit mehreren Werten

Die Namen der Felder, die für Header und Abfragezeichenfolgeparameter verwendet werden, unterscheiden sich abhängig davon, ob Sie Header mit mehreren Werten für die Zielgruppe aktivieren.

Die folgende Beispielanfrage enthält zwei Abfrageparameter mit demselben Schlüssel:


```
http://www.example.com?&myKey=val1&myKey=val2
```

Bei dem Standardformat verwendet der Load Balancer den letzten vom Client gesendeten Wert und sendet Ihnen mit `queryStringParameters` ein Ereignis, das Abfragezeichenfolgeparameter umfasst. Beispielsweise:

```
"queryStringParameters": { "myKey": "val2"},
```

Wenn Sie Header mit mehreren Werten aktivieren, verwendet der Load Balancer beide vom Client gesendeten Schlüsselwerte und sendet Ihnen einen Zeichenfolgeparameter für eine Ereignisabfrage, der `multiValueQueryStringParameters` verwendet. Beispielsweise:

```
"multiValueQueryStringParameters": { "myKey": ["val1", "val2"] },
```

Dementsprechend gilt Folgendes, wenn der Client eine Anfrage mit zwei Cookies im Header sendet:

```
"cookie": "name1=value1",  
"cookie": "name2=value2",
```

Bei dem Standardformat verwendet der Load Balancer den letzten vom Client gesendeten Cookie und sendet Ihnen mit `headers` ein Ereignis, das Header umfasst. Beispielsweise:

```
"headers": {  
  "cookie": "name2=value2",  
  ...  
},
```

Bei Headern mit mehreren Werten verwendet der Load Balancer beide vom Client gesendeten Cookies und sendet Ihnen mit `multiValueHeaders` ein Ereignis, das Header umfasst. Beispielsweise:

```
"multiValueHeaders": {  
  "cookie": ["name1=value1", "name2=value2"],  
  ...  
},
```

Wenn die Abfrageparameter URL-kodiert sind, dekodiert der Load Balancer sie nicht. Sie müssen sie in Ihrer Lambda-Funktion dekodieren.

Antworten mit Headern mit mehreren Werten

Der Name der für Header verwendeten Felder unterscheidet sich abhängig davon, ob Sie Header mit mehreren Werten für die Zielgruppe verwenden. Sie müssen `multiValueHeaders` verwenden, wenn Sie Header mit mehreren Werten aktivieren. Andernfalls verwenden Sie `headers`.

Bei dem Standardformat können Sie ein einziges Cookie angeben:

```
{
  "headers": {
    "Set-cookie": "cookie-name=cookie-value;Domain=myweb.com;Secure;HttpOnly",
    "Content-Type": "application/json"
  },
}
```

Bei Headern mit mehreren Werten müssen Sie wie folgt mehrere Cookies angeben:

```
{
  "multiValueHeaders": {
    "Set-cookie": ["cookie-name=cookie-
value;Domain=myweb.com;Secure;HttpOnly", "cookie-name=cookie-value;Expires=May 8,
2019"],
    "Content-Type": ["application/json"]
  },
}
```

Der Load Balancer sendet die Header ggf. in einer anderen Reihenfolge als der in der Lambda-Antwortnutzlast angegebenen Reihenfolge an den Client. Verlassen Sie sich daher nicht darauf, dass Header in einer bestimmten Reihenfolge zurückgegeben werden.

Aktivieren von Headern mit mehreren Werten

Sie können Header mit mehreren Werten für eine Zielgruppe mit dem Zieltyp `lambda` aktivieren oder deaktivieren.

So aktivieren Sie die Header mit mehreren Werten mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.

4. Wählen Sie auf der Registerkarte Gruppendetails im Abschnitt Attribute die Option Bearbeiten aus.
5. Aktivieren oder deaktivieren Sie Header mit mehreren Werten.
6. Wählen Sie Änderungen speichern aus.

Um Header mit mehreren Werten zu aktivieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-target-group-attributes](#) zusammen mit dem Attribut `lambda.multi_value_headers.enabled`.

Aktivieren von Zustandsprüfungen

Zustandsprüfungen sind für Zielgruppen des Typs Lambda standardmäßig deaktiviert. Sie können Zustandsprüfungen aktivieren, um ein DNS Failover mit Amazon Route 53 zu implementieren. Die Lambda-Funktion kann den Zustand eines Downstream-Service prüfen, bevor sie auf die Anfrage einer Zustandsprüfung antwortet. Wenn die Antwort von der Lambda-Funktion auf eine nicht bestandene Zustandsprüfung hinweist, wird die nicht bestandene Zustandsprüfung an Amazon Route 53 übergeben. Sie können Amazon Route 53 für das Failover auf einen Sicherungsanwendungs-Stack konfigurieren.

Ihnen werden für Zustandsprüfungen genauso wie für jeden anderen Lambda-Funktionsaufruf Gebühren erhoben.

Im Folgenden finden Sie das Format des an Ihre Lambda-Funktion gesendeten Zustandsprüfungsereignisses. Um zu prüfen, ob ein Ereignis eine Zustandsprüfungsereignis ist, überprüfen Sie den Wert des Feldes "user-agent". Der Benutzeragent für Zustandsprüfungen ist `ELB-HealthChecker/2.0`.

```
{
  "requestContext": {
    "elb": {
      "targetGroupArn":
      "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-target-
      group/6d0ecf831eec9f09"
    }
  },
  "httpMethod": "GET",
  "path": "/",
  "queryStringParameters": {},
```

```
"headers": {
  "user-agent": "ELB-HealthChecker/2.0"
},
"body": "",
"isBase64Encoded": false
}
```

Um Integritätsprüfungen für eine Zielgruppe mithilfe der Konsole zu aktivieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie auf der Registerkarte Gruppendetails im Abschnitt Einstellungen für die Zustandsprüfung die Option Bearbeiten aus.
5. Wählen Sie unter Zustandsprüfungen die Option Aktivieren aus.
6. Wählen Sie Änderungen speichern aus.

Um Integritätsprüfungen für eine Zielgruppe zu aktivieren, verwenden Sie den AWS CLI

Verwenden Sie den Befehl [modify-target-group](#) zusammen mit der Option `--health-check-enabled`.

Aufheben der Registrierung der Lambda-Funktion

Wenn Sie zu Ihrer Lambda-Funktion keinen Datenverkehr mehr senden müssen, können Sie ihre Registrierung aufheben. Nachdem Sie die Registrierung einer Lambda-Funktion aufgehoben haben, schlagen laufende Anfragen mit HTTP-5XX-Fehlermeldungen fehl.

Zum Ersetzen einer Lambda-Funktion wird empfohlen, eine neue Zielgruppe zu erstellen, die neue Funktion bei der neuen Zielgruppe zu registrieren und die Listener-Regeln so zu aktualisieren, dass anstatt der vorhandenen die neue Zielgruppe verwendet wird.

So deregistrieren Sie die Lambda-Funktion mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.

3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Klicken Sie auf der Registerkarte Targets (Ziele) auf Deregister (Registrierung aufheben).
5. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Deregister aus.

Um die Registrierung der Lambda-Funktion mit dem AWS CLI

Verwenden Sie den Befehl [deregister-targets](#).

Tags für Ihre Zielgruppe

Tags helfen Ihnen, Ihre Zielgruppen auf unterschiedliche Weise zu kategorisieren, z.B. nach Zweck, Eigentümer oder Umgebung.

Sie können mehrere Tags für jede Zielgruppe hinzufügen. Tag-Schlüssel müssen für jede Zielgruppe eindeutig sein. Wenn Sie ein Tag mit einem Schlüssel hinzufügen, der der Zielgruppe bereits zugeordnet ist, ändert sich der Wert dieses Tags.

Wenn Sie ein Tag nicht mehr benötigen, können Sie es entfernen.

Einschränkungen

- Maximale Anzahl von Tags pro Ressource: 50
- Maximale Schlüssellänge: 127 Unicode-Zeichen
- Maximale Wertlänge: 255 Unicode-Zeichen
- Bei Tag-Schlüsseln und -Werten wird zwischen Groß- und Kleinschreibung unterschieden. Erlaubte Zeichen sind Buchstaben, Leerzeichen und Zahlen, die in UTF-8 darstellbar sind, sowie die folgenden Sonderzeichen: + - = _ : / @. Verwenden Sie keine führenden oder nachgestellten Leerzeichen.
- Verwenden Sie das `aws :` Präfix nicht in Ihren Tagnamen oder -Werten, da es für die Verwendung reserviert ist. AWS Sie können keine Tag-Namen oder Werte mit diesem Präfix bearbeiten oder löschen. Tags mit diesem Präfix werden nicht als Ihre Tags pro Ressourcenlimit angerechnet.

So aktualisieren Sie die Tags für eine Zielgruppe mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.

3. Wählen Sie den Namen der Zielgruppe aus, um deren Detailseite zu öffnen.
4. Wählen Sie auf der Registerkarte Tags die Option Tags verwalten und führen Sie einen oder mehrere der folgenden Schritte aus:
 - a. Um ein Tag zu aktualisieren, geben Sie neue Werte für Schlüssel und Wert ein.
 - b. Um ein Tag hinzuzufügen, wählen Sie Tag hinzufügen und geben Sie Werte für Schlüssel und Wert ein.
 - c. Um ein Tag zu löschen, wählen Sie Entfernen neben dem Tag.
5. Wenn Sie die Aktualisierung der Tags abgeschlossen haben, wählen Sie Änderungen speichern.

Um die Tags für eine Zielgruppe mit dem zu aktualisieren AWS CLI

Verwenden Sie die Befehle [add-tags](#) und [remove-tags](#).

Löschen einer Zielgruppe

Sie können eine Zielgruppe löschen, wenn sie nicht von den Weiterleitungsaktionen der Listener-Regeln referenziert wird. Das Löschen einer Zielgruppe hat keine Auswirkungen auf die Ziele, die bei der Zielgruppe registriert sind. Wenn Sie die registrierte EC2-Instance nicht mehr benötigen, können Sie sie anhalten oder beenden.

Löschen einer Zielgruppe mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich unter Load Balancing die Option Target Groups (Zielgruppen) aus.
3. Markieren Sie die Zielgruppe und wählen Sie Aktionen, Löschen.
4. Wenn Sie zur Bestätigung aufgefordert werden, wählen Sie Ja, löschen.

Um eine Zielgruppe mit dem zu löschen AWS CLI

Verwenden Sie den Befehl [delete-target-group](#).

Überwachen Ihrer Application Load Balancer

Sie können die folgenden Features verwenden, um Ihre Load Balancer zu überwachen, Datenverkehrsmuster zu analysieren und Probleme mit Ihrem Load Balancer und Zielen zu beheben.

CloudWatch Metriken

Sie können Amazon verwenden CloudWatch , um Statistiken über Datenpunkte für Ihre Load Balancer und Ziele in Form eines geordneten Satzes von Zeitreihendaten, den so genannten Metriken, abzurufen. Mit diesen Metriken können Sie überprüfen, ob Ihr System die erwartete Leistung zeigt. Weitere Informationen finden Sie unter [CloudWatch Metriken für Ihren Application Load Balancer](#).

Zugriffsprotokolle

Sie können Zugriffsprotokolle verwenden, um detaillierte Informationen zu den Anfragen, die an Ihren Load Balancer gestellt werden, zu erfassen und sie als Protokolldateien in Amazon S3 speichern. Sie können anhand dieser Zugriffsprotokolle Datenverkehrsmuster analysieren und Probleme mit Ihren Zielen beheben. Weitere Informationen finden Sie unter [Zugriffsprotokolle für Ihre Application Load Balancer](#).

Verbindungsprotokolle.

Sie können Verbindungsprotokolle verwenden, um Attribute der an Ihren Load Balancer gesendeten Anfragen zu erfassen und sie als Protokolldateien in Amazon S3 zu speichern. Sie können diese Verbindungsprotokolle verwenden, um die Client-IP-Adresse und den Port, die Client-Zertifikatsinformationen, die Verbindungsergebnisse und die verwendeten TLS-Chiffren zu ermitteln. Diese Verbindungsprotokolle können dann verwendet werden, um Anforderungsmuster und andere Trends zu überprüfen. Weitere Informationen finden Sie unter [Verbindungsprotokolle für Ihren Application Load Balancer](#).

Anfragenachverfolgung

Sie können Anfragenachverfolgung verwenden, um HTTP-Anfragen nachzuverfolgen. Der Load Balancer fügt zu jede Anfrage, die er erhält, einen Header mit eine Ablaufverfolgungskennung hinzu. Weitere Informationen finden Sie unter [Anfragenachverfolgung für Ihren Application Load Balancer](#).

CloudTrail Protokolle

Sie können AWS CloudTrail damit detaillierte Informationen zu den Aufrufen der Elastic Load Balancing API erfassen und sie als Protokolldateien in Amazon S3 speichern. Sie können diese

CloudTrail Protokolle verwenden, um festzustellen, welche Aufrufe getätigt wurden, von welcher Quell-IP-Adresse der Anruf kam, wer den Anruf getätigt hat, wann der Anruf getätigt wurde usw. Weitere Informationen finden Sie unter [Protokollierung von API-Aufrufen für Ihren Application Load Balancer mit AWS CloudTrail](#).

CloudWatch Metriken für Ihren Application Load Balancer

Elastic Load Balancing veröffentlicht Datenpunkte CloudWatch für Ihre Load Balancer und Ihre Ziele auf Amazon. CloudWatchermöglicht es Ihnen, Statistiken über diese Datenpunkte in Form eines geordneten Satzes von Zeitreihendaten, sogenannten Metriken, abzurufen. Sie können sich eine Metrik als eine zu überwachende Variable und die Datenpunkte als die Werte dieser Variable im Laufe der Zeit vorstellen. Sie können z. B. die Gesamtanzahl der funktionierenden Ziele für einen Load Balancer für einen angegebenen Zeitraum überwachen. Jeder Datenpunkt verfügt über einen zugewiesenen Zeitstempel und eine optionale Maßeinheit.

Mit den Metriken können Sie überprüfen, ob Ihr System die erwartete Leistung zeigt. Sie können beispielsweise einen CloudWatch Alarm erstellen, um eine bestimmte Metrik zu überwachen und eine Aktion einzuleiten (z. B. das Senden einer Benachrichtigung an eine E-Mail-Adresse), wenn die Metrik außerhalb des für Sie akzeptablen Bereichs liegt.

Elastic Load Balancing meldet Metriken CloudWatch nur dann, wenn Anfragen durch den Load Balancer fließen. Wenn Anforderungen über den Load Balancer erfolgen, misst Elastic Load Balancing diese und sendet seine Metriken in 60-Sekunden-Intervallen. Wenn es keine Anfragen über den Load Balancer gibt oder keine Daten für eine Metrik vorliegen, wird die Metrik nicht gemeldet.

Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

Inhalt

- [Application-Load-Balancer-Metriken](#)
- [Metrik-Dimensionen für Application Load Balancer](#)
- [Statistiken für Application-Load-Balancer-Metriken](#)
- [CloudWatch Metriken für Ihren Load Balancer anzeigen](#)

Application-Load-Balancer-Metriken

- [Load Balancers](#)

- [Targets \(Ziele\)](#)
- [Zustand der Zielgruppe](#)
- [Lambda-Funktionen](#)
- [Benutzerauthentifizierung](#)

Der AWS/ApplicationELB-Namespace enthält die folgenden Metriken für Load Balancer.

Metrik	Beschreibung
ActiveConnectionCount	<p>Gesamtanzahl gleichzeitiger aktiver TCP-Verbindungen zwischen Clients und Load Balancer sowie zwischen Load Balancer und Zielen.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer
AnomalousHostCount	<p>Die Anzahl der Hosts, bei denen Anomalien festgestellt wurden.</p> <p>Berichtskriterien: Always reported</p> <p>Statistiken: Die nützlichsten Statistiken sind Average, Minimum und Maximum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • TargetGroup , LoadBalancer • TargetGroup , AvailabilityZone , LoadBalancer
ClientTLSNegotiationErrorCount	<p>Die Anzahl der TLS-Verbindungen, die vom Client initiiert wurden und aufgrund eines TLS-Fehlers keine Sitzung mit dem Load Balancer hergestellt haben. Mögliche Ursachen sind eine Nichtübereinstimmung</p>

Metrik	Beschreibung
	<p>ng von Verschlüsselungen oder Protokollen oder der Client, der das Serverzertifikat nicht überprüft und die Verbindung schließt.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer
ConsumedLCUs	<p>Anzahl von Load Balancer-Kapazitätseinheiten (LCU), die von Ihrem Load Balancer verwendet werden. Sie zahlen nur für die pro Stunde genutzte Anzahl der LCU. Weitere Informationen finden Sie unter Elastic Load Balancing Pricing.</p> <p>Berichtskriterien: Always reported</p> <p>Statistiken: Alle</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer
DesyncMitigationMode_NonCompliant_Request_Count	<p>Die Anzahl der Anforderungen, die nicht RFC 7230 entsprechen.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer

Metrik	Beschreibung
DroppedInvalidHeaderRequestCount	<p>Die Anzahl der Anforderungen, bei denen der Load Balancer HTTP-Header mit Header-Feldern entfernt hat, die nicht gültig sind, bevor die Anforderung nicht weitergeleitet wird. Der Load Balancer entfernt diese Header nur, wenn das <code>routing.http.drop_invalid_header_fields.enabled</code> -Attribut auf „true“ festgelegt ist.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Alle</p> <p>Dimensionen</p> <ul style="list-style-type: none">• AvailabilityZone , LoadBalancer
MitigatedHostCount	<p>Die Anzahl der Ziele, für die eine Minderung vorgesehen ist.</p> <p>Berichtskriterien: Always reported</p> <p>Statistiken: Die nützlichsten Statistiken sind Average, Minimum und Maximum.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• TargetGroup , LoadBalancer• TargetGroup , AvailabilityZone , LoadBalancer

Metrik	Beschreibung
ForwardedInvalidHeaderRequestCount	<p>Die Anzahl der vom Load Balancer weitergeleiteten Anfragen mit HTTP-Headern mit ungültigen Header-Feldern. Der Load Balancer leitet Anfragen mit diesen Headern nur dann weiter, wenn das <code>routing.http.drop_invalid_header_fields.enabled</code>-Attribut auf „false“ festgelegt ist.</p> <p>Berichtskriterien: Always reported</p> <p>Statistiken: Alle</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • AvailabilityZone , LoadBalancer
GrpcRequestCount	<p>Die Anzahl von gRPC-Anforderungen, die per IPv4 und IPv6 verarbeitet wurden.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum. Minimum, Maximum und Average geben alle 1 zurück.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer
HTTP_Fixed_Response_Count	<p>Die Anzahl der Aktionen mit feststehender Antwort, die erfolgreich waren.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer

Metrik	Beschreibung
HTTP_Redirect_Count	<p>Die Anzahl der Redirect-Aktionen, die erfolgreich waren.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• LoadBalancer• AvailabilityZone , LoadBalancer
HTTP_Redirect_Url_Limit_Exceeded_Count	<p>Die Anzahl der Redirect-Aktionen, die nicht abgeschlossen werden konnten, weil die URL im Header der Antwortadresse größer als 8K ist.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• LoadBalancer• AvailabilityZone , LoadBalancer
HTTPCode_ELB_3XX_Count	<p>Die Anzahl der HTTP-3XX-Redirect-Codes, die vom Load Balancer stammen. Diese Anzahl enthält keine von Zielen erzeugte Antwortcodes.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• LoadBalancer• AvailabilityZone , LoadBalancer

Metrik	Beschreibung
HTTPCode_ELB_4XX_Count	<p>Anzahl von HTTP-4XX-Client-Fehlercodes, die vom Load Balancer verursacht werden. Diese Anzahl enthält keine von Zielen erzeugte Antwortcodes.</p> <p>Client-Fehler werden bei Anforderungen mit falschem Format oder unvollständigen Anforderungen generiert. Diese Anforderungen wurden vom Ziel nicht empfangen, anders als in dem Fall, in dem der Load Balancer einen HTTP 460-Fehlercode zurückgibt. Diese Anzahl enthält keine von Zielen erzeugte Antwortcodes.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum. Minimum, Maximum und Average geben alle 1 zurück.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• LoadBalancer• AvailabilityZone , LoadBalancer
HTTPCode_ELB_5XX_Count	<p>Anzahl von HTTP-5XX-Server-Fehlercodes, die vom Load Balancer verursacht werden. Diese Anzahl enthält keine von Zielen erzeugte Antwortcodes.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum. Minimum, Maximum und Average geben alle 1 zurück.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• LoadBalancer• AvailabilityZone , LoadBalancer

Metrik	Beschreibung
HTTPCode_ELB_500_Count	<p>Anzahl von HTTP-500-Fehlercodes, die vom Load Balancer verursacht werden.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• LoadBalancer• AvailabilityZone , LoadBalancer
HTTPCode_ELB_502_Count	<p>Anzahl von HTTP-502-Fehlercodes, die vom Load Balancer verursacht werden.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• LoadBalancer• AvailabilityZone , LoadBalancer
HTTPCode_ELB_503_Count	<p>Anzahl von HTTP-503-Fehlercodes, die vom Load Balancer verursacht werden.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• LoadBalancer• AvailabilityZone , LoadBalancer

Metrik	Beschreibung
HTTPCode_ELB_504_Count	<p>Anzahl von HTTP-504-Fehlercodes, die vom Load Balancer verursacht werden.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• LoadBalancer• AvailabilityZone , LoadBalancer
IPv6ProcessedBytes	<p>Gesamtanzahl der von einem Load Balancer über IPv6 verarbeiteten Byte. Diese Anzahl ist in ProcessedBytes enthalten.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• LoadBalancer• AvailabilityZone , LoadBalancer
IPv6RequestCount	<p>Anzahl der vom Load Balancer empfangenen IPv6-Anforderungen.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum. Minimum, Maximum und Average geben alle 1 zurück.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• LoadBalancer• AvailabilityZone , LoadBalancer

Metrik	Beschreibung
NewConnectionCount	<p data-bbox="521 226 1471 352">Gesamtanzahl neuer TCP-Verbindungen, die zwischen Clients und Load Balancer und zwischen Load Balancer und Zielen hergestellt wurden.</p> <p data-bbox="521 401 1068 436">Berichtskriterien: Ein Wert ungleich Null</p> <p data-bbox="521 478 1130 514">Statistiken: Die nützlichste Statistik ist Sum.</p> <p data-bbox="521 556 711 592">Dimensionen</p> <ul data-bbox="521 640 1130 737" style="list-style-type: none"><li data-bbox="521 640 784 676">• LoadBalancer<li data-bbox="521 695 1130 737">• AvailabilityZone , LoadBalancer
NonStickyRequestCount	<p data-bbox="521 779 1498 1150">Die Anzahl von Anfragen, bei denen der Load Balancer ein neues Ziel wählte, da eine vorhandene Sticky Session nicht verwendet werden konnte. Beispiel: Die Anfrage war die erste Anfrage von einem neuen Client und es wurde kein Sticky-Cookie vorgelegt oder es wurde ein Sticky-Cookie vorgelegt, aber ohne Angabe eines Ziels, das bei dieser Zielgruppe registriert war, oder das Sticky-Cookie war falsch formatiert oder abgelaufen oder der Load Balancer konnte das Sticky-Cookie aufgrund eines internen Fehlers nicht lesen.</p> <p data-bbox="521 1192 1430 1228">Berichtskriterien: Sticky Sessions sind für die Zielgruppe aktiviert.</p> <p data-bbox="521 1270 1317 1306">Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p data-bbox="521 1348 711 1383">Dimensionen</p> <ul data-bbox="521 1432 1130 1528" style="list-style-type: none"><li data-bbox="521 1432 784 1467">• LoadBalancer<li data-bbox="521 1488 1130 1528">• AvailabilityZone , LoadBalancer

Metrik	Beschreibung
ProcessedBytes	<p data-bbox="521 226 1474 453">Gesamtanzahl der von einem Load Balancer über IPv4 und IPv6 verarbeiteten Bytes (HTTP-Header und HTTP-Nutzlast). Diese Anzahl umfasst den Datenverkehr zu und von Clients und Lambda-Funktionen sowie den Datenverkehr von einem Identity Provider (IdP), wenn die Benutzerauthentifizierung aktiviert ist.</p> <p data-bbox="521 495 1068 531">Berichtskriterien: Ein Wert ungleich Null</p> <p data-bbox="521 573 1130 609">Statistiken: Die nützlichste Statistik ist Sum.</p> <p data-bbox="521 651 711 686">Dimensionen</p> <ul data-bbox="521 728 1130 827" style="list-style-type: none"><li data-bbox="521 728 784 764">• LoadBalancer<li data-bbox="521 785 1130 827">• AvailabilityZone , LoadBalancer
RejectedConnectionCount	<p data-bbox="521 873 1463 957">Anzahl der abgelehnten Verbindungen, weil der Load Balancer die maximale Anzahl an Verbindungen erreicht hat.</p> <p data-bbox="521 999 1068 1035">Berichtskriterien: Ein Wert ungleich Null</p> <p data-bbox="521 1077 1130 1113">Statistiken: Die nützlichste Statistik ist Sum.</p> <p data-bbox="521 1155 711 1190">Dimensionen</p> <ul data-bbox="521 1232 1130 1331" style="list-style-type: none"><li data-bbox="521 1232 784 1268">• LoadBalancer<li data-bbox="521 1289 1130 1331">• AvailabilityZone , LoadBalancer

Metrik	Beschreibung
RequestCount	<p>Die Anzahl von Anforderungen, die per IPv4 und IPv6 verarbeitet wurden. Diese Metrik wird nur für Anforderungen erhöht, bei denen der Load-Balancer-Knoten ein Ziel auswählen konnte. Anforderungen, die abgelehnt werden, bevor ein Ziel ausgewählt wurde, werden in dieser Metrik nicht berücksichtigt.</p> <p>Berichtskriterien: Always reported</p> <p>Statistiken: Die nützlichste Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • LoadBalancer , AvailabilityZone • LoadBalancer , TargetGroup • LoadBalancer , AvailabilityZone , TargetGroup
RuleEvaluations	<p>Anzahl der von einem Load Balancer verarbeiteten Regeln auf Grundlage einer über eine Stunde gemittelten Anforderungsrate.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer

Der AWS/ApplicationELB-Namespace enthält die folgenden Metriken für Ziele.

Metrik	Beschreibung
HealthyHostCount	<p>Anzahl der als stabil betrachteten Ziele.</p> <p>Berichtskriterien: Wird gemeldet, wenn Zustandsprüfungen aktiviert sind</p>

Metrik	Beschreibung
	<p>Statistiken: Die nützlichsten Statistiken sind Average, Minimum und Maximum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer , TargetGroup • LoadBalancer , AvailabilityZone , TargetGroup
<p>HTTPCode_Target_2XX_Count , HTTPCode_Target_3XX_Count , HTTPCode_Target_4XX_Count , HTTPCode_Target_5XX_Count</p>	<p>Anzahl der HTTP-Antwortcodes, die von den Zielen generiert wurden. Hierin sind keine vom Load Balancer generierten Antwortcodes enthalten.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum. Minimum, Maximum und Average geben alle 1 zurück.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup , LoadBalancer • TargetGroup , AvailabilityZone , LoadBalancer

Metrik	Beschreibung
RequestCountPerTarget	<p>Die durchschnittliche Anzahl der Anfragen pro Ziel in einer Zielgruppe. Sie müssen die Zielgruppe mithilfe der Dimension TargetGroup angeben. Diese Metrik gilt nicht, wenn das Ziel eine Lambda-Funktion ist.</p> <p>Diese Anzahl basiert auf der Gesamtzahl der Anfragen, die von der Zielgruppe eingegangen sind, geteilt durch die Anzahl der gesunden Ziele in der Zielgruppe. Wenn es in der Zielgruppe keine gesunden Ziele gibt, wird die Gesamtzahl der Ziele gemeldet.</p> <p>Berichtskriterien: Always reported</p> <p>Statistiken: Die einzige zulässige Statistik ist Sum. Hier wird der Durchschnitt und nicht die Summe angegeben.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • TargetGroup • TargetGroup , AvailabilityZone • LoadBalancer , TargetGroup • LoadBalancer , AvailabilityZone , TargetGroup
TargetConnectionErrorCount	<p>Anzahl der Verbindungen, die zwischen dem Load Balancer und dem Ziel nicht erfolgreich hergestellt wurden. Diese Metrik gilt nicht, wenn das Ziel eine Lambda-Funktion ist.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup , LoadBalancer • TargetGroup , AvailabilityZone , LoadBalancer

Metrik	Beschreibung
TargetResponseTime	<p>Die verstrichene Zeit in Sekunden, nachdem die Anfrage den Load Balancer verlassen hat, bis das Ziel beginnt, die Antwortheader zu senden. Dies entspricht dem Feld <code>target_processing_time</code> in den Zugriffsprotokollen.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichsten Statistiken sind Average und pNN.NN (Perzentile).</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup , LoadBalancer • TargetGroup , AvailabilityZone , LoadBalancer
TargetTLSEnvironmentErrorCount	<p>Anzahl der TLS-Verbindungen, die vom Load Balancer initiiert wurden und keine Sitzung mit dem Ziel hergestellt haben. Als mögliche Ursachen kommen unter anderem fehlende Übereinstimmung bei Verschlüsselungsverfahren oder Protokollen infrage. Diese Metrik gilt nicht, wenn das Ziel eine Lambda-Funktion ist.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer • TargetGroup , LoadBalancer • TargetGroup , AvailabilityZone , LoadBalancer

Metrik	Beschreibung
UnHealthyHostCount	<p>Die Anzahl der als instabil betrachteten Ziele.</p> <p>Berichtskriterien: Wird gemeldet, wenn Zustandsprüfungen aktiviert sind</p> <p>Statistiken: Die nützlichsten Statistiken sind Average, Minimum und Maximum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer , TargetGroup • LoadBalancer , AvailabilityZone , TargetGroup

Der AWS/ApplicationELB-Namespace enthält die folgenden Metriken für den Zustand der Zielgruppe. Weitere Informationen finden Sie unter [the section called “Zustand der Zielgruppe”](#).

Metrik	Beschreibung
HealthyStateDNS	<p>Die Anzahl der Zonen, die die Anforderungen an einen fehlerfreien DNS-Zustand erfüllen.</p> <p>Statistiken: Die nützlichste Statistik ist Min.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer , TargetGroup • AvailabilityZone , LoadBalancer , TargetGroup
HealthyStateRouting	<p>Die Anzahl der Zonen, die die Anforderungen an einen fehlerfreien Zustand für das Routing erfüllen.</p> <p>Statistiken: Die nützlichste Statistik ist Min.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer , TargetGroup

Metrik	Beschreibung
	<ul style="list-style-type: none"> • AvailabilityZone , LoadBalancer , TargetGroup
UnhealthyRoutingRequestCount	<p>Die Anzahl der Anforderungen, die mithilfe der Routing-Failover-Aktion (Fail-Open) weitergeleitet werden.</p> <p>Statistiken: Die nützlichste Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer , TargetGroup • AvailabilityZone , LoadBalancer , TargetGroup
UnhealthyStateDNS	<p>Die Anzahl der Zonen, die die Anforderungen an einen fehlerfreien DNS-Zustand nicht erfüllen und daher in DNS als fehlerhaft markiert wurden.</p> <p>Statistiken: Die nützlichste Statistik ist Min.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer , TargetGroup • AvailabilityZone , LoadBalancer , TargetGroup
UnhealthyStateRouting	<p>Die Anzahl der Zonen, die die Anforderungen an einen fehlerfreien Zustand für das Routing nicht erfüllen. Daher verteilt der Load Balancer den Datenverkehr an alle Ziele in der Zone, einschließlich der fehlerhaften Ziele.</p> <p>Statistiken: Die nützlichste Statistik ist Min.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer , TargetGroup • AvailabilityZone , LoadBalancer , TargetGroup

Der AWS/ApplicationELB-Namespace enthält die folgenden Metriken für Lambda-Funktionen, die als Ziele registriert sind.

Metrik	Beschreibung
<code>LambdaInternalError</code>	<p>Die Anzahl von Anfragen an eine Lambda-Funktion, die aufgrund eines internen Problems des Load Balancer oder von AWS Lambda fehlgeschlagen sind. Um die Codes für die Fehlerursache zu erhalten, überprüfen Sie das Feld "error_reason" des Zugriffsprotokolls.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• <code>TargetGroup</code>• <code>TargetGroup</code> , <code>LoadBalancer</code>
<code>LambdaTargetProcessedBytes</code>	<p>Die Gesamtzahl von Bytes, die vom Load Balancer für Anfragen an und Antworten von einer Lambda-Funktion verarbeitet wurden.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none">• <code>LoadBalancer</code>
<code>LambdaUserError</code>	<p>Die Anzahl von Anfragen an eine Lambda-Funktion, die aufgrund eines Problems mit der Lambda-Funktion fehlgeschlagen sind. Beispiel: Der Load Balancer war nicht zum Aufrufen der Funktion berechtigt, der Load Balancer empfing eine JSON-Datei von der Funktion, die falsch formatiert war oder nicht alle erforderlichen Felder enthielt, oder die Größe des Anfragetextes oder der Antwort überschritt die maximale Größe von 1 MB. Um die Codes für die Fehlerursache zu erhalten, überprüfen Sie das Feld "error_reason" des Zugriffsprotokolls.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p>

Metrik	Beschreibung
	<p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • TargetGroup • TargetGroup , LoadBalancer

Der AWS/ApplicationELB-Namespace enthält die folgenden Metriken für die Benutzerauthentifizierung.

Metrik	Beschreibung
ELBAuthError	<p>Die Anzahl der Benutzerauthentifizierungen, die nicht abgeschlossen werden konnten, weil eine Authentifizierungsaktion falsch konfiguriert war, der Load Balancer keine Verbindung mit dem Identitätsanbieter herstellen konnte oder der Load Balancer den Authentifizierungsfluss aufgrund eines internen Fehlers nicht abschließen konnte. Um die Codes für die Fehlerursache zu erhalten, überprüfen Sie das Feld "error_reason" des Zugriffsprotokolls.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer
ELBAuthFailure	<p>Die Anzahl der Benutzerauthentifizierungen, die nicht abgeschlossen werden konnten, weil der Identitätsanbieter den Zugriff auf den Benutzer abgelehnt hat oder ein Autorisierungscode mehr als einmal verwendet wurde. Um die Codes für die Fehlerursache zu erhalten, überprüfen Sie das Feld "error_reason" des Zugriffsprotokolls.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p>

Metrik	Beschreibung
	<p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer
ELBAuthLatency	<p>Die verstrichene Zeit in Millisekunden zur Abfrage des Identitätsanbieters nach dem ID-Token und den Benutzerinformationen. Falls eine oder mehrere dieser Operationen ausfällt, wird ein Fehler ausgegeben.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Alle Statistiken sind aussagekräftig.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer
ELBAuthRefreshTokenSuccess	<p>Gibt an, wie oft der Load Balancer Benutzeransprüche mit einem vom Identitätsanbieter bereitgestellten Aktualisierungstoken erfolgreich aktualisiert hat.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer

Metrik	Beschreibung
ELBAuthSuccess	<p>Die Anzahl der Authentifizierungsaktionen, die erfolgreich waren. Diese Metrik wird am Ende des Authentifizierungsworkflows erhöht, wenn der Load Balancer die Benutzeransprüche vom Identitätsanbieter abgerufen hat.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die nützlichste Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer
ELBAuthUserClaimsSizeExceeded	<p>Gibt an, wie oft ein konfigurierter Identitätsanbieter Benutzeransprüche zurückgegeben hat, die eine Größe von 11 KB überschritten haben.</p> <p>Berichtskriterien: Ein Wert ungleich Null</p> <p>Statistiken: Die einzige aussagekräftige Statistik ist Sum.</p> <p>Dimensionen</p> <ul style="list-style-type: none"> • LoadBalancer • AvailabilityZone , LoadBalancer

Metrik-Dimensionen für Application Load Balancer

Verwenden Sie die nachstehenden Dimensionen, um die Metriken für Ihren Application Load Balancer zu filtern.

Dimension	Beschreibung
AvailabilityZone	Filtert die Metrikdaten nach Availability Zone.

Dimension	Beschreibung
LoadBalancer	Filtert die Metrikdaten nach Load Balancer. Geben Sie den Load Balancer wie folgt an: app/Name des Load Balancers/1234567890123456 (der letzte Teil des Load Balancer-ARNs).
TargetGroup	Filtert die Metrikdaten nach der Zielgruppe. Geben Sie die Zielgruppe wie folgt an: app/Name der Zielgruppe/1234567890123456 (der letzte Teil des Zielgruppen-ARNs).

Statistiken für Application-Load-Balancer-Metriken

CloudWatch stellt Statistiken bereit, die auf den von Elastic Load Balancing veröffentlichten metrischen Datenpunkten basieren. Statistiken sind Metrikdaten-Aggregationen über einen bestimmten Zeitraum. Wenn Sie Statistiken anfordern, wird der zurückgegebene Datenstrom durch den Metriknamen und die Dimension identifiziert. Eine Dimension ist ein Name-Wert-Paar, durch das eine Metrik eindeutig identifiziert wird. Beispielsweise können Sie Statistiken für alle fehlerfreien EC2-Instances hinter einem Load Balancer, die in einer bestimmten Availability Zone gestartet wurden, anfordern.

Die Minimum- und Maximum-Statistiken geben die Mindest- und Maximalwerte der Datenpunkte an, die von den einzelnen Load Balancer-Knoten in jedem Sampling-Fenster gemeldet werden. Nehmen wir beispielsweise an, es gibt 2 Load-Balancer-Knoten, die den Application Load Balancer bilden. Ein Knoten hat `HealthyHostCount` mit dem Minimum-Wert 2, dem Maximum-Wert 10 und dem Average-Wert 6, während der andere Knoten `HealthyHostCount` mit dem Minimum-Wert 1, dem Maximum-Wert 5 und dem Average-Wert 3 aufweist. Somit weist der Load Balancer den Minimum-Wert 1, den Maximum-Wert 10 und den Average-Wert von etwa 4 auf.

Wir empfehlen Ihnen, in der Minimum-Statistik auf `UnHealthyHostCount`-Werte ungleich Null zu achten und bei mehr als einem Datenpunkt einen Alarm zu senden, wenn ein Wert ungleich Null ist. Mithilfe von `Minimum` wird erkannt, wann Ziele von jedem Knoten und jeder Availability Zone Ihres Load Balancers als fehlerhaft eingestuft werden. Das Senden von Alarmen in Bezug auf `Average` oder `Maximum` ist nützlich, wenn Sie vor potenziellen Problemen gewarnt werden möchten. Wir empfehlen unseren Kunden, diese Metrik zu überprüfen und Vorkommnisse zu untersuchen, die ungleich Null sind. Das automatische Minimieren von Ausfällen kann gemäß den bewährten Methoden zur Verwendung der Load-Balancer-Zustandsprüfung in Amazon EC2 Auto Scaling oder Amazon Elastic Container Service (Amazon ECS) erfolgen.

Die Sum-Statistik stellt den Gesamtwert aller Load Balancer-Knoten dar. Da Metriken mehrere Berichte pro Zeitraum umfassen, gilt Sum nur für Metriken, die über alle Load Balancer-Knoten aggregiert werden.

Die SampleCount-Statistik ist die Zahl der gemessenen Stichproben. Da Metriken basierend auf Erfassungsintervallen und Ereignissen erfasst werden, ist diese Statistik in der Regel nicht nützlich. Bei HealthyHostCount basiert SampleCount z. B. auf der Anzahl der Stichproben, die jeder Load Balancer-Knoten meldet, nicht auf der Anzahl fehlerfreier Hosts.

Ein Perzentil gibt die relative Stelle eines Wertes in einem Datensatz an. Sie können ein beliebiges Perzentil mit bis zu zwei Dezimalstellen (z. B. p95,45) angeben. Ein 95. Perzentil bedeutet, dass 95 Prozent der Daten unter diesem Wert und 5 Prozent darüber liegen. Perzentile werden häufig genutzt, um Anomalien zu isolieren. Angenommen, eine Anwendung bedient die meisten Anforderungen aus einem Cache in 1-2 ms, aber benötigt 100 bis 200 ms, wenn der Cache leer ist. Das Maximum spiegelt den langsamsten Fall wider, etwa 200 ms. Der Durchschnitt gibt nicht die Verteilung der Daten an. Perzentile bieten eine aussagekräftigere Darstellung der Anwendungs-Performance. Indem Sie das 99. Perzentil als Auto Scaling-Trigger oder CloudWatch Alarm verwenden, können Sie festlegen, dass die Verarbeitung von nicht mehr als 1 Prozent der Anfragen länger als 2 ms dauert.

CloudWatch Metriken für Ihren Load Balancer anzeigen

Sie können die CloudWatch Metriken für Ihre Load Balancer mithilfe der Amazon EC2 EC2-Konsole anzeigen. Diese Metriken werden in Überwachungsdiagrammen dargestellt. Die Überwachungsdiagramme zeigen Datenpunkte, wenn der Load Balancer aktiv ist und Anforderungen erhält.

Alternativ können Sie die Metriken für Ihren Load Balancer über die Konsole anzeigen. CloudWatch

So zeigen Sie Metriken mithilfe der -Konsole an

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Um nach Zielgruppe gefilterte Metriken anzuzeigen, führen Sie die folgenden Schritte aus:
 - a. Wählen Sie im Navigationsbereich Target Groups aus.
 - b. Wählen Sie Ihre Zielgruppe und wählen Sie dann die Registerkarte Monitoring (Überwachung) aus.
 - c. (Optional) Wählen Sie in Showing data for einen Zeitbereich aus., um die Ergebnisse nach Zeit zu filtern.

- d. Wenn Sie eine größere Ansicht einer Metrik aufrufen möchten, wählen Sie ihr Diagramm aus.
3. Um nach Load Balancer gefilterte Metriken anzuzeigen, gehen Sie wie folgt vor:
 - a. Klicken Sie im Navigationsbereich auf Load Balancers.
 - b. Wählen Sie Ihren Load Balancer aus und wählen Sie dann die Registerkarte Monitoring (Überwachung) aus.
 - c. (Optional) Wählen Sie in Showing data for einen Zeitbereich aus., um die Ergebnisse nach Zeit zu filtern.
 - d. Wenn Sie eine größere Ansicht einer Metrik aufrufen möchten, wählen Sie ihr Diagramm aus.

Um Metriken mit der CloudWatch Konsole anzuzeigen

1. Öffnen Sie die CloudWatch Konsole unter <https://console.aws.amazon.com/cloudwatch/>.
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie den Namespace ApplicationELB aus.
4. (Optional) Um eine Metrik in allen Dimensionen anzuzeigen, geben Sie den Namen in das Suchfeld ein.
5. (Optional) Um nach Maß zu filtern, wählen Sie einen der folgenden Schritte aus:
 - Wenn Sie nur die für Ihre Load Balancer gemeldeten Metriken anzeigen möchten, wählen Sie Per AppELB Metrics (Pro AppELB-Metrik). aus . Um die Metriken für einen einzelnen Load Balancer anzuzeigen, geben Sie den Namen in das Suchfeld ein.
 - Wenn Sie ausschließlich die für Ihre Load Balancer gemeldeten Zielgruppen anzeigen möchten, wählen Sie Per AppELB, per TG Metrics (Metriken pro AppELB, pro Zielgruppe) aus. Um die Metriken für eine einzelne Zielgruppe anzuzeigen, geben Sie den Namen in das Suchfeld ein.
 - Wenn Sie ausschließlich die für Ihre Load Balancer gemeldeten Metriken nach Availability Zone anzeigen möchten, wählen Sie Per AppELB, per AZ Metrics (Metriken pro AZ, pro AppELB) aus. Um die Metriken für einen einzelnen Load Balancer anzuzeigen, geben Sie den Namen in das Suchfeld ein. Um die Metriken für eine einzelne Availability Zone anzuzeigen, geben Sie den Namen in das Suchfeld ein.
 - Wenn Sie ausschließlich die für Ihre Load Balancer gemeldeten Metriken nach Availability Zone und Zielgruppe anzeigen möchten, wählen Sie Per AppELB, per AZ, per TG Metrics

(Metriken pro AZ, pro AppELB, pro Zielgruppe) aus. Um die Metriken für einen einzelnen Load Balancer anzuzeigen, geben Sie den Namen in das Suchfeld ein. Um die Metriken für eine einzelne Zielgruppe anzuzeigen, geben Sie den Namen in das Suchfeld ein. Um die Metriken für eine einzelne Availability Zone anzuzeigen, geben Sie den Namen in das Suchfeld ein.

Um Metriken mit dem anzuzeigen AWS CLI

Verwenden Sie den folgenden [list-metrics](#)-Befehl, um die verfügbaren Metriken aufzuführen:

```
aws cloudwatch list-metrics --namespace AWS/ApplicationELB
```

Um die Statistiken für eine Metrik abzurufen, verwenden Sie AWS CLI

Verwenden Sie den folgenden Befehl [get-metric-statistics, um Statistiken](#) für die angegebene Metrik und Dimension abzurufen. CloudWatch behandelt jede eindeutige Kombination von Dimensionen als separate Metrik. Sie können keine Statistiken abrufen, die Kombinationen von Dimensionen verwenden, die nicht speziell veröffentlicht wurden. Sie müssen die gleichen Dimensionen angeben, die bei der Erstellung der Metriken verwendet wurden.

```
aws cloudwatch get-metric-statistics --namespace AWS/ApplicationELB \  
--metric-name UnHealthyHostCount --statistics Average --period 3600 \  
--dimensions Name=LoadBalancer,Value=app/my-load-balancer/50dc6c495c0c9188 \  
Name=TargetGroup,Value=targetgroup/my-targets/73e2d6bc24d8a067 \  
--start-time 2016-04-18T00:00:00Z --end-time 2016-04-21T00:00:00Z
```

Das Folgende ist Ausgabebeispiel:

```
{  
  "Datapoints": [  
    {  
      "Timestamp": "2016-04-18T22:00:00Z",  
      "Average": 0.0,  
      "Unit": "Count"  
    },  
    {  
      "Timestamp": "2016-04-18T04:00:00Z",  
      "Average": 0.0,  
      "Unit": "Count"  
    },  
    ...  
  ]  
}
```



```
  ],  
  "Label": "UnHealthyHostCount"  
}
```

Zugriffsprotokolle für Ihre Application Load Balancer

Elastic Load Balancing bietet Zugriffsprotokolle, die detaillierte Informationen zu Anforderungen erfassen, die an Ihren Load Balancer gesendet werden. Jedes Protokoll enthält Informationen wie die Zeit, zu der die Anforderung einging, die Client-IP-Adresse, Latenzen, Anforderungspfade und Serverantworten. Sie können diese Zugriffsprotokolle für die Analyse von Datenverkehrsmustern und zur Problembehebung verwenden.

Zugriffsprotokolle sind ein optionales Feature von Elastic Load Balancing, das standardmäßig deaktiviert ist. Nachdem Sie die Zugriffsprotokolle für Ihren Load Balancer aktiviert haben, erfasst Elastic Load Balancing die Protokolle und speichert sie in dem von Ihnen angegebenen Amazon-S3-Bucket als komprimierte Dateien. Sie können die Zugriffsprotokolle jederzeit deaktivieren.

Sie zahlen Speicherkosten für Amazon S3, aber Sie zahlen nicht für die Bandbreite, die von Elastic Load Balancing zum Senden von Protokolldateien an Amazon S3 verwendet wird. Weitere Information zu Speicherkosten finden Sie unter [Amazon S3 – Preise](#).

Inhalt

- [Zugriffsprotokolldateien](#)
- [Zugriffsprotokolleinträge](#)
- [Beispiel-Protokolleinträge](#)
- [Verarbeiten von Zugriffsprotokolldateien](#)
- [Aktivieren der Zugriffsprotokolle für Ihren Application Load Balancer](#)
- [Deaktivieren der Zugriffsprotokolle für Ihren Application Load Balancer](#)

Zugriffsprotokolldateien

Elastic Load Balancing veröffentlicht alle 5 Minuten eine Protokolldatei für jeden Load-Balancer-Knoten. Die Protokollbereitstellung ist letztendlich konsistent. Der Load Balancer kann mehrere Protokolle für denselben Zeitraum bereitstellen. Dies passiert in der Regel, wenn die Website hohen Datenverkehr aufweist.

Die Dateinamen der Zugriffsprotokolle verwenden das folgende Format:

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/aws-account-id_elasticloadbalancing_region_app.load-balancer-id_end-time_ip-address_random-string.log.gz
```

bucket

Der Name des S3-Buckets.

prefix

(Optional) Das Präfix (logische Hierarchie) für den Bucket. Das von Ihnen angegebene Präfix darf die Zeichenfolge AWSLogs nicht enthalten. Weitere Informationen finden Sie unter [Organisieren von Objekten mit Präfixen](#).

AWSLogs

Wir fügen den Teil des Dateinamens hinzu, der mit AWSLogs nach dem von Ihnen angegebenen Bucket-Namen und dem optionalen Präfix beginnt.

aws-account-id

Die AWS Konto-ID des Besitzers.

Region

Die Region für Ihren Load Balancer und den S3-Bucket.

JJJJ/MM/TT

Das Datum, an dem das Protokoll übermittelt wurde.

load-balancer-id

Die Ressourcen-ID des Load Balancer. Wenn die Ressourcen-ID Schrägstriche (/) enthält, werden sie durch Punkte (.) ersetzt.

end-time

Das Datum und die Uhrzeit, an dem das Protokollierungsintervall endete. Beispiel: Die Endzeit 20140215T2340Z enthält Einträge für Anforderungen, die zwischen 23:35 und 23:40 in UTC- oder Zulu-Zeit durchgeführt wurden.

ip-address

Die IP-Adresse des Load Balancer-Knotens, der die Anforderung verarbeitet hat. Für einen internen Load Balancer handelt es sich hierbei um eine private IP-Adresse.

random-string

Eine vom System generierte zufällige Zeichenfolge.

Im Folgenden finden Sie ein Beispiel für einen Protokolldateinamen mit Präfix:

```
s3://my-bucket/my-prefix/AWSLogs/123456789012/elasticloadbalancing/us-east-2/2022/05/01/123456789012_elasticloadbalancing_us-east-2_app.my-loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Im Folgenden finden Sie ein Beispiel für einen Protokolldateinamen ohne Präfix:

```
s3://my-bucket/AWSLogs/123456789012/elasticloadbalancing/us-east-2/2022/05/01/123456789012_elasticloadbalancing_us-east-2_app.my-loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Sie können Ihre Protokolldateien beliebig lange im Bucket speichern. Sie können aber auch Amazon S3-Lebenszyklusregeln aufstellen, anhand derer die Protokolldateien automatisch archiviert oder gelöscht werden. Weitere Informationen finden Sie unter [Objekt-Lebenszyklusmanagement](#) im Benutzerhandbuch zu Amazon Simple Storage Service.

Zugriffsprotokolleinträge

Elastic Load Balancing protokolliert Anforderungen, die an den Load Balancer gesendet wurden, einschließlich Anforderungen, die nicht bei den Zielen ankamen. Wenn ein Client beispielsweise eine falsch formatierte Anfrage sendet oder keine fehlerfreien Ziele vorhanden sind, die auf die Anfrage reagieren können, wird die Anfrage dennoch protokolliert. Elastic Load Balancing protokolliert keine Zustandsprüfungsanforderungen.

Jeder Protokolleintrag enthält die Details einer einzelnen Anfrage (oder Verbindung im Fall von WebSockets), die an den Load Balancer gestellt wurde. Denn ein Eintrag wird erst geschrieben WebSockets, nachdem die Verbindung geschlossen wurde. Wenn die erweiterte Verbindung nicht hergestellt werden kann, entspricht der Eintrag dem für eine HTTP- oder HTTPS-Anfrage.

Important

Elastic Load Balancing protokolliert Anforderungen nach bestmöglichem Bemühen. Wir empfehlen, dass Sie die Zugriffsprotokolle verwenden, um die Art der Anforderungen zu verstehen, nicht als eine vollständige Buchführung aller Anforderungen.

Inhalt

- [Syntax](#)
- [Ergriffene Maßnahmen](#)
- [Gründe für die Klassifizierung](#)
- [Codes für die Fehlerursache](#)

Syntax

Die folgende Tabelle beschreibt die Felder eines Zugriffsprotokolleintrags der Reihe nach. Alle Felder werden durch Leerzeichen voneinander getrennt. Wenn neue Felder eingeführt werden, werden sie am Ende des Protokolleintrags hinzugefügt. Sie sollten alle Felder am Ende des Protokolleintrags ignorieren, die Sie nicht erwartet haben.

Feld	Beschreibung
Typ	<p>Der Typ der Anfrage oder Verbindung. Die möglichen Werte sind wie folgt (alle anderen Werte ignorieren):</p> <ul style="list-style-type: none"> • <code>http</code> – HTTP • <code>https</code> – HTTP über TLS • <code>h2</code> – HTTP/2 über TLS • <code>grpc</code> – gRPC über TLS • <code>ws</code> – WebSockets • <code>wss</code> – WebSockets über TLS
time	Die Zeit, zu der der Load Balancer eine Antwort an den Client generiert hat, im Format ISO 8601. Wenn dies die Zeit WebSockets, in der die Verbindung geschlossen wird.
elb	Die Ressourcen-ID des Load Balancer. Wenn Sie Zugriffsprotokolleinträge analysieren, beachten Sie, dass Ressourcen-IDs Schrägstriche (/) enthalten können.
client:port	Die IP-Adresse und den Port des anfordernden Clients. Wenn sich vor dem Load Balancer ein Proxy befindet, enthält dieses Feld die IP-Adresse des Proxys.

Feld	Beschreibung
target:port	<p>Die IP-Adresse und der Port des Ziels, das diese Anfrage verarbeitet hat.</p> <p>Wenn der Client keine vollständige Anfrage gesendet hat, kann der Load Balancer die Anfrage keinem Ziel zuteilen, und dieser Wert wird auf - festgelegt.</p> <p>Wenn das Ziel eine Lambda-Funktion ist, ist dieser Wert auf - eingestellt.</p> <p>Wenn die Anfrage blockiert wird von AWS WAF, wird dieser Wert auf - und der Wert von <code>elb_status_code</code> auf 403 gesetzt.</p>
request_processing_time	<p>Die insgesamt verstrichene Zeit (in Sekunden, mit einer Genauigkeit in Millisekunden) ab dem Zeitpunkt, zu dem der Load Balancer die Anforderung erhalten hat, bis zu dem Zeitpunkt, zu dem er die Anforderung an ein Ziel gesendet hat.</p> <p>Dieser Wert wird auf -1 festgelegt, wenn der Load Balancer die Anfrage keinem Ziel zuteilen kann. Dies kann der Fall sein, wenn das Ziel die Verbindung vor dem Leerlaufzeitlimit schließt, oder wenn der Client eine falsch formatierte Anfrage sendet.</p> <p>Dieser Wert kann auch auf -1 gesetzt werden, wenn das registrierte Ziel nicht vor dem Timeoutwert für die Leerlaufzeit antwortet.</p> <p>Wenn für Ihren Application Load Balancer aktiviert AWS WAF ist oder der Zieltyp eine Lambda-Funktion ist, wird die Zeit, die der Client benötigt, um die erforderlichen Daten für POST-Anfragen zu senden, angerechnet. <code>request_processing_time</code></p>

Feld	Beschreibung
target_processing_time	<p>Die insgesamt verstrichene Zeit (in Sekunden, mit einer Genauigkeit in Millisekunden) ab dem Zeitpunkt, zu dem der Load Balancer die Anfrage an ein Ziel gesendet hat, bis zu dem Zeitpunkt, zu dem das Ziel begonnen hat, die Antwort-Header zu senden.</p> <p>Dieser Wert wird auf -1 festgelegt, wenn der Load Balancer die Anfrage keinem Ziel zuteilen kann. Dies kann der Fall sein, wenn das Ziel die Verbindung vor dem Leerlaufzeitlimit schließt, oder wenn der Client eine falsch formatierte Anfrage sendet.</p> <p>Dieser Wert kann auch auf -1 gesetzt werden, wenn das registrierte Ziel nicht vor dem Timeoutwert für die Leerlaufzeit antwortet.</p> <p>Wenn es für Ihren Application Load Balancer nicht aktiviert AWS WAF ist, wird die Zeit angerechnet target_processing_time , die der Client benötigt, um die erforderlichen Daten für POST-Anfragen zu senden.</p>
response_processing_time	<p>Die insgesamt verstrichene Zeit (in Sekunden, mit einer Genauigkeit in Millisekunden) ab dem Zeitpunkt, zu dem der Load Balancer den Antwort-Header vom Ziel erhalten hat, bis zu dem Zeitpunkt, zu dem er begonnen hat, die Antwort an den Client zu senden. Dies umfasst sowohl die Wartezeit am Load Balancer als auch die Zeit für die Herstellung der Verbindung vom Load Balancer zum Client.</p> <p>Dieser Wert wird auf -1 gesetzt, wenn der Load Balancer keine Antwort von einem Ziel erhält. Dies kann der Fall sein, wenn das Ziel die Verbindung vor dem Leerlaufzeitlimit schließt, oder wenn der Client eine falsch formatierte Anfrage sendet.</p>
elb_status_code	Der Statuscode der Antwort vom Load Balancer.
target_status_code	Der Statuscode der Antwort vom Ziel. Dieser Wert wird nur aufgezeichnet, wenn eine Verbindung zum Ziel hergestellt wurde und das Ziel eine Antwort gesendet hat. Andernfalls lautet der Wert –.

Feld	Beschreibung
received_bytes	Die Größe der Anforderung, in Byte, die vom Client (Auftraggeber) eingegangen ist. Bei HTTP-Anfragen sind die Header eingeschlossen. Denn dies ist die Gesamtzahl der Byte WebSockets, die vom Client über die Verbindung empfangen wurden.
sent_bytes	Die Größe der Antwort, in Byte, die an den Client (Auftraggeber) gesendet wurde. Bei HTTP-Anfragen sind die Header eingeschlossen. Denn dies ist die Gesamtzahl der Byte WebSockets, die über die Verbindung an den Client gesendet wurden.
„request“	Die Anfragezeile vom Client in Anführungszeichen und protokolliert mit folgendem Format: HTTP-Methode + Protokoll://Host:Port/URI + HTTP-Version. Der Load Balancer behält die vom Client gesendete URL bei der Aufnahme des Anforderungs-URI unverändert bei. Es wird kein Inhaltstyp für die Zugriffsprotokolldatei festgelegt. Berücksichtigen Sie bei der Verarbeitung des Feldes, wie der Client die URL gesendet hat.
„user_agent“	Eine User-Agent-Zeichenfolge, die den Client identifiziert, von dem die Anfrage stammt, in Anführungszeichen. Die Zeichenfolge besteht aus einem oder mehreren Produkt-IDs, Produkt[/Version]. Wenn die Zeichenfolge länger als 8 KB ist, wird sie gekürzt.
ssl_cipher	[HTTPS-Listener] Die SSL-Verschlüsselung. Dieser Wert ist auf „-“ festgelegt, wenn es sich bei dem Listener nicht um einen HTTPS-Listener handelt.
ssl_protocol	[HTTPS-Listener] Das SSL-Protokoll. Dieser Wert ist auf „-“ festgelegt, wenn es sich bei dem Listener nicht um einen HTTPS-Listener handelt.
target_group_arn	Der Amazon-Ressourcenname (ARN) der Zielgruppe.
"trace_id"	Der Inhalt des X-Amzn-Trace-Id-Headers in Anführungszeichen.
"domain_name"	[HTTPS-Listener] Die vom Client beim TLS-Handshake bereitgestellte SNI-Domain in Anführungszeichen. Dieser Wert wird auf - festgelegt, wenn der Client SNI nicht unterstützt oder die Domain keinem Zertifikat entspricht und dem Client das Standardzertifikat vorgelegt wird.

Feld	Beschreibung
"chosen_cert_arn"	[HTTPS-Listener] Der ARN des Zertifikats, das dem Client vorgelegt wird, in Anführungszeichen. Dieser Wert ist auf <code>session-reused</code> festgelegt, wenn die Sitzung wiederverwendet wird. Dieser Wert ist auf „-“ festgelegt, wenn es sich bei dem Listener nicht um einen HTTPS-Listener handelt.
matched_rule_priority	Der Prioritätswert der Regel, die eine Übereinstimmung mit der Anforderung erzeugt. Wenn eine Regel eine Übereinstimmung erzeugt hat, ist dies ein Wert zwischen 1 und 50.000. Hat keine Regel eine Übereinstimmung erzeugt und die Standardaktion wurde ausgeführt, ist dieser Wert 0. Wenn während der Regelauswertung ein Fehler auftritt, beträgt der Wert -1. Für alle anderen Fehler lautet der Wert -.
request_creation_time	Die Uhrzeit, zu der der Load Balancer die Anforderung vom Client erhalten hat, im ISO 8601-Format.
"actions_executed"	Die bei der Verarbeitung der Anfrage ergriffenen Maßnahmen in Anführungszeichen. Dieser Wert ist eine durch Kommas getrennte Liste, die die in Ergriffene Maßnahmen beschriebenen Werte enthalten kann. Wenn keine Maßnahmen ergriffen wurden, beispielsweise bei einer falsch formatierten Anfrage, wird dieser Wert auf - festgelegt.
"redirect_url"	Die URL des Weiterleitungsziels für den Location-Header der HTTP-Antwort in doppelten Anführungszeichen. Wenn keine Weiterleitungsaktionen ausgeführt wurden, wird dieser Wert auf "-" eingestellt.
"error_reason"	Der Ursachencode in doppelten Anführungszeichen. Wenn die Anforderung fehlschlug, ist dies einer der in Codes für die Fehlerursache beschriebenen Fehlercodes. Wenn die durchgeführten Aktionen keine Authentifizierungsaktion umfassen oder das Ziel keine Lambda-Funktion ist, wird dieser Wert auf „-“ festgelegt.

Feld	Beschreibung
„target:port_list“	<p>Eine durch Leerzeichen getrennte Liste von IP-Adressen und Ports für die Ziele, die diese Anforderung verarbeitet haben, eingeschlossen in doppelten Anführungszeichen. Derzeit kann diese Liste ein Element enthalten und es entspricht dem target:port-Feld.</p> <p>Wenn der Client keine vollständige Anfrage gesendet hat, kann der Load Balancer die Anfrage keinem Ziel zuteilen, und dieser Wert wird auf - festgelegt.</p> <p>Wenn das Ziel eine Lambda-Funktion ist, ist dieser Wert auf - einstellt.</p> <p>Wenn die Anfrage blockiert wird von AWS WAF, wird dieser Wert auf - und der Wert von elb_status_code auf 403 gesetzt.</p>
„target_status_code_list“	<p>Eine durch Leerzeichen getrennte Liste von Statuscodes aus den Antworten der Ziele, die in doppelte Anführungszeichen eingeschlossen sind. Derzeit kann diese Liste ein Element enthalten und entspricht dem target_status_code-Feld.</p> <p>Dieser Wert wird nur aufgezeichnet, wenn eine Verbindung zum Ziel hergestellt wurde und das Ziel eine Antwort gesendet hat. Andernfalls lautet der Wert -.</p>
„classification“	<p>Die Klassifikation für die desynchrone Mitigation in doppelten Anführungszeichen. Wenn die Anforderung nicht den Anforderungen von RFC 7230 entspricht, lauten die möglichen Werte „Acceptable“ (Akzeptabel), „Ambiguous“ (Mehrdeutig) und „Severe“ (Schwerwiegend).</p> <p>Wenn die Anforderung RFC 7230 entspricht, wird dieser Wert auf „-“ gesetzt.</p>
„classification_reason“	<p>Der Ursachencode für die Klassifizierung steht in doppelten Anführungszeichen. Wenn die Anforderung nicht RFC 7230 entspricht, ist dies einer der unter Gründe für die Klassifizierung beschriebenen Klassifizierungscodes. Wenn die Anforderung RFC 7230 entspricht, wird dieser Wert auf „-“ gesetzt.</p>

Feld	Beschreibung
conn_trace_id	Die Verbindungsrückverfolgbarkeits-ID ist eine eindeutige und durchsichtige ID, die zur Identifizierung jeder Verbindung verwendet wird. Nachdem eine Verbindung mit einem Client hergestellt wurde, enthalten nachfolgende Anfragen von diesem Client diese ID in ihren jeweiligen Zugriffsprotokolleinträgen. Diese ID fungiert als Fremdschlüssel, um eine Verbindung zwischen den Verbindungs- und Zugriffsprotokollen herzustellen.

Ergriffene Maßnahmen

Der Load Balancer speichert die ausgeführten Maßnahmen im Feld „actions_executed“ des Zugriffsprotokolls.

- `authenticate` – Der Load Balancer hat die Sitzung validiert, den Benutzer authentifiziert und die Benutzerinformationen den Anforderungs-Headern hinzugefügt, wie in der Regelkonfiguration angegeben.
- `fixed-response` – Der Load Balancer hat eine feste Antwort ausgegeben, wie in der Regelkonfiguration angegeben.
- `forward` – Der Load Balancer hat die Anforderung an ein Ziel weitergeleitet, wie in der Regelkonfiguration angegeben.
- `redirect` – Der Load Balancer hat die Anforderung an eine andere URL weitergeleitet, wie in der Regelkonfiguration angegeben.
- `waf` – Der Load Balancer hat die Anforderung an AWS WAF weitergeleitet, um festzustellen, ob die Anforderung an das Ziel weitergeleitet werden sollte. Wenn dies die letzte Aktion ist, wurde AWS WAF festgestellt, dass die Anfrage abgelehnt werden sollte.
- `waf-failed`— Der Load Balancer hat versucht, die Anfrage weiterzuleiten AWS WAF, aber dieser Vorgang ist fehlgeschlagen.

Gründe für die Klassifizierung

Wenn eine Anforderung nicht RFC 7230 entspricht, speichert der Load Balancer einen der folgenden Codes im Feld „classification_reason“ des Zugriffsprotokolls. Weitere Informationen finden Sie unter [Desynchroner Mitigationsmodus](#).

Code	Beschreibung	Klassifizierung
AmbiguousUri	Der Anforderungs-URI enthält Steuerzeichen.	Mehrdeutig
BadContentLength	Der Content-Length-Header enthält einen Wert, der nicht analysiert werden kann oder der keine gültige Zahl ist.	Schwerwiegend
BadHeader	Ein Header enthält ein Nullzeichen oder einen Zeilenumbruch.	Schwerwiegend
BadTransferEncoding	Der Transfer-Encoding-Header enthält einen ungültigen Wert.	Schwerwiegend
BadUri	Die Anforderungs-URI enthält ein Nullzeichen oder ein Zeilenumkehrzeichen.	Schwerwiegend
BadMethod	Die Anforderungsmethode ist falsch formatiert.	Schwerwiegend
BadVersion	Die Anforderungsversion ist falsch formatiert.	Schwerwiegend
BothTeClPresent	Die Anforderung enthält sowohl einen Transfer-Encoding-Header als auch einen Content-Length-Header.	Mehrdeutig
DuplicateContentLength	Es gibt mehrere Content-Length-Header mit demselben Wert.	Mehrdeutig
EmptyHeader	Eine Kopfzeile ist leer oder es gibt eine Zeile, die nur Leerzeichen enthält.	Mehrdeutig
GetHeaderZeroContentLength	Es gibt einen Content-Length-Header mit dem Wert 0 für eine GET- oder HEAD-Anforderung.	Akzeptabel
MultipleContentLength	Es gibt mehrere Content-Length-Header mit verschiedenen Werten.	Schwerwiegend

Code	Beschreibung	Klassifizierung
MultipleTransferEncodingChunked	Es gibt mehrere „Transfer-Encoding: chunked“-Header.	Schwerwiegend
NonCompliantHeader	Ein Header enthält ein Nicht-ASCII- oder Steuerzeichen.	Akzeptabel
NonCompliantVersion	Die Anforderungsversion enthält einen ungültigen Wert.	Akzeptabel
SpaceInUri	Die Anforderungs-URI enthält ein Leerzeichen, das nicht URL-codiert ist.	Akzeptabel
SuspiciousHeader	Es gibt einen Header, der mithilfe gängiger Textnormalisierungstechniken auf Transfer-Encoding oder Content-Length normalisiert werden kann.	Mehrdeutig
UndefinedContentLengthSemantics	Für eine GET- oder HEAD-Anforderung ist ein Content-Length-Header definiert.	Mehrdeutig
UndefinedTransferEncodingSemantics	Für eine GET- oder HEAD-Anforderung ist ein Transfer-Encoding-Header definiert.	Mehrdeutig

Codes für die Fehlerursache

Wenn der Load Balancer keine Authentifizierungsaktion abschließen kann, speichert er einen der folgenden Ursachencodes im Feld „error_reason“ im Aktionsprotokoll. Der Load Balancer erhöht auch die entsprechende Metrik. CloudWatch Weitere Informationen finden Sie unter [Authentifizieren von Benutzern mithilfe eines Application Load Balancers](#).

Code	Beschreibung	Metrik
AuthInvalidCookie	Das Authentifizierungs-Cookie ist ungültig.	ELBAuthFailure
AuthInvalidGrantError	Der Code zur Gewährung der Authentifizierung vom Token-Endpunkt ist ungültig.	ELBAuthFailure
AuthInvalidIdToken	Das ID-Token ist ungültig.	ELBAuthFailure
AuthInvalidStateParam	Der Zustandsparameter ist ungültig.	ELBAuthFailure
AuthInvalidTokenResponse	Die Antwort vom Token-Endpunkt ist ungültig.	ELBAuthFailure
AuthInvalidUserInfoResponse	Die Antwort vom Benutzerinformationsendpunkt ist ungültig.	ELBAuthFailure
AuthMissingCodeParam	Der Authentifizierungsantwort vom Authentifizierungsendpunkt fehlt ein Abfrageparameter namens „code“.	ELBAuthFailure
AuthMissingHostHeader	Der Authentifizierungsantwort vom Authentifizierungsendpunkt fehlt ein Host-Header-Feld.	ELBAuthError
AuthMissingStateParam	Der Authentifizierungsantwort vom Authentifizierungsendpunkt fehlt ein Abfrageparameter namens „state“.	ELBAuthFailure
AuthTokenEpRequestFailed	Es liegt eine Fehlerantwort (non-2XX) vom Token-Endpunkt vor.	ELBAuthError

Code	Beschreibung	Metrik
AuthTokenEpRequestTimeout	Der Load Balancer kann nicht mit dem Token-Endpoint kommunizieren.	ELBAuthError
AuthUnhandledException	Der Load Balancer hat eine unbehandelte Ausnahme festgestellt.	ELBAuthError
AuthUserInfoEndpointFailed	Es liegt eine Fehlerantwort (non-2XX) vom Identitätsanbieter-Benutzerinformationsendpunkt vor.	ELBAuthError
AuthUserInfoEndpointTimeout	Der Load Balancer kann nicht mit dem Identitätsanbieter-Benutzerinformationsendpunkt kommunizieren.	ELBAuthError
AuthUserInfoResponseSizeExceeded	Die Größe der vom Identitätsanbieter zurückgegebenen Ansprüche übersteigt 11 K Byte.	ELBAuthUserClaimsSizeExceeded

Wenn eine Anforderung an eine gewichtete Zielgruppe fehlschlägt, speichert der Load Balancer einen der folgenden Fehlercodes im Feld „error_reason“ des Zugriffsprotokolls.

Code	Beschreibung
AWSALBTGCookieInvalid	Das AWSALBTG Cookie, das bei gewichteten Zielgruppen verwendet wird, ist nicht gültig. Beispielsweise gibt der Load Balancer diesen Fehler zurück, wenn Cookie-Werte URL-codiert sind.
WeightedTargetGroupUnhandledException	Der Load Balancer hat eine unbehandelte Ausnahme festgestellt.

Wenn eine Anforderung an eine Lambda-Funktion fehlschlägt, speichert der Load Balancer eine der folgenden Ursachencodes im Feld "error_reason" des Zugriffsprotokolls. Der Load Balancer erhöht

auch die entsprechende CloudWatch Metrik. Weitere Informationen finden Sie unter der Lambda-Aktion [Invoke](#).

Code	Beschreibung	Metrik
<code>LambdaAccessDenied</code>	Der Load Balancer war nicht zum Aufrufen der Lambda-Funktion berechtigt.	<code>LambdaUserError</code>
<code>LambdaBadRequest</code>	Der Lambda-Aufruf ist fehlgeschlagen, da die Clientanforderungsheader oder der Hauptteil nicht nur UTF-8-Zeichen enthalten.	<code>LambdaUserError</code>
<code>LambdaConnectionError</code>	Der Load Balancer kann keine Verbindung mit Lambda herstellen.	<code>LambdaInternalError</code>
<code>LambdaConnectionTimeout</code>	Bei dem Versuch, eine Verbindung mit Lambda herzustellen, ist eine Zeitüberschreitung aufgetreten.	<code>LambdaInternalError</code>
<code>LambdaEC2AccessDeniedException</code>	Amazon EC2 hat während der Initialisierung der Funktion den Zugriff auf Lambda abgelehnt.	<code>LambdaUserError</code>
<code>LambdaEC2ThrottledException</code>	Amazon EC2 drosselte Lambda während der Initialisierung der Funktion.	<code>LambdaUserError</code>
<code>LambdaEC2UnexpectedException</code>	Bei Amazon EC2 trat während der Initialisierung der Funktion ein unerwarteter Fehler auf.	<code>LambdaUserError</code>
<code>LambdaENILimitReachedException</code>	Lambda konnte in der VPC, die in der Konfiguration der Lambda-Funktion angegeben wird, keine Netzwerkschnittstelle erstellen, da das Limit für Netzwerkschnittstellen überschritten wurde.	<code>LambdaUserError</code>

Code	Beschreibung	Metrik
<code>LambdaInvalidResponse</code>	Die Antwort von der Lambda-Funktion ist falsch formatiert oder enthält nicht alle erforderlichen Felder.	<code>LambdaUserError</code>
<code>LambdaInvalidRuntimeException</code>	Die angegebene Version der Lambda-Laufzeit wird nicht unterstützt.	<code>LambdaUserError</code>
<code>LambdaInvalidSecurityGroupIDException</code>	Die angegebene Sicherheitsgruppen-ID in der Konfiguration der Lambda-Funktion ist nicht gültig.	<code>LambdaUserError</code>
<code>LambdaInvalidSubnetIDException</code>	Die angegebene Subnetz-ID in der Konfiguration der Lambda-Funktion ist nicht gültig.	<code>LambdaUserError</code>
<code>LambdaInvalidZipFileException</code>	Lambda konnte die angegebene Funktions-Zip-Datei nicht entpacken.	<code>LambdaUserError</code>
<code>LambdaKMSAccessDeniedException</code>	Lambda konnte die Umgebungsvariablen nicht entschlüsseln, da der Zugriff auf den KMS-Schlüssel abgelehnt wurde. Überprüfen Sie die KMS-Berechtigungen der Lambda-Funktion.	<code>LambdaUserError</code>
<code>LambdaKMSDisabledException</code>	Lambda konnte die Umgebungsvariablen nicht entschlüsseln, da der angegebene KMS-Schlüssel deaktiviert ist. Überprüfen Sie die Einstellungen des KMS-Schlüssels der Lambda-Funktion.	<code>LambdaUserError</code>

Code	Beschreibung	Metrik
LambdaKMSInvalidStateException	Lambda konnte die Umgebungsvariablen nicht entschlüsseln, da der Zustand des KMS-Schlüssels nicht gültig ist. Überprüfen Sie die Einstellungen des KMS-Schlüssels der Lambda-Funktion.	LambdaUserError
LambdaKMSNotFoundException	Lambda konnte die Umgebungsvariablen nicht entschlüsseln, da der KMS-Schlüssel nicht gefunden wurde. Überprüfen Sie die Einstellungen des KMS-Schlüssels der Lambda-Funktion.	LambdaUserError
LambdaRequestTooLarge	Die Größe des Anfragetextes überschritt 1 MB.	LambdaUserError
LambdaResourceNotFound	Die Lambda-Funktion konnte nicht gefunden werden.	LambdaUserError
LambdaResponseTooLarge	Die Größe der Antwort überschritt 1 MB.	LambdaUserError
LambdaServiceException	Bei Lambda trat ein interner Fehler auf.	LambdaInternalError
LambdaSubnetIPAddressLimitReachedException	Lambda konnte den VPC-Zugriff für die Lambda-Funktion nicht einrichten, da ein oder mehrere konfigurierte Subnetze keine verfügbaren IP-Adressen haben.	LambdaUserError
LambdaThrottling	Die Lambda-Funktion wurde gedrosselt, da es zu viele Anfragen gab.	LambdaUserError
LambdaUnhandled	Bei der Lambda-Funktion trat eine unbehandelte Ausnahme auf.	LambdaUserError

Code	Beschreibung	Metrik
LambdaUnhandledException	Der Load Balancer hat eine unbehandelte Ausnahme festgestellt.	LambdaInternalError
LambdaWebSocketNotSupported	WebSockets werden von Lambda nicht unterstützt.	LambdaUserError

Wenn der Load Balancer bei der Weiterleitung von Anfragen an einen Fehler AWS WAF feststellt, speichert er einen der folgenden Fehlercodes im Feld `error_reason` des Zugriffsprotokolls.

Code	Beschreibung
WAFConnectionError	Der Load Balancer kann keine Verbindung zu herstellen. AWS WAF
WAFConnectionTimeout	Bei der Verbindung wurde das AWS WAF Zeitlimit überschritten.
WAFResponseReadTimeout	Eine Anfrage zum AWS WAF Timeout.
WAFServiceError	AWS WAF hat einen 5XX-Fehler zurückgegeben.
WAFUnhandledException	Der Load Balancer hat eine unbehandelte Ausnahme festgestellt.

Beispiel-Protokolleinträge

Es folgen beispielhafte Protokolleinträge. Beachten Sie, dass der Text nur aus Gründen der besseren Lesbarkeit auf mehrere Zeilen verteilt ist.

Beispiel für HTTP-Eintrag

Es folgt ein Beispiel für einen Protokolleintrag für einen HTTP-Listener (Port 80 zu Port 80):

```
http 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
```

```
192.168.131.39:2817 10.0.0.1:80 0.000 0.001 0.000 200 200 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337262-36d228ad5d99923122bbe354" "-" "-"
0 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.0.1:80" "200" "-" "-"
```

Beispiel für HTTPS-Eintrag

Es folgt ein Beispiel für einen Protokolleintrag für einen HTTPS-Listener (Port 443 zu Port 80):

```
https 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 10.0.0.1:80 0.086 0.048 0.037 200 200 0 57
"GET https://www.example.com:443/ HTTP/1.1" "curl/7.46.0" ECDHE-RSA-AES128-GCM-SHA256
  TLSv1.2
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337281-1d84f3d73c47ec4e58577259" "www.example.com" "arn:aws:acm:us-
east-2:123456789012:certificate/12345678-1234-1234-1234-123456789012"
1 2018-07-02T22:22:48.364000Z "authenticate,forward" "-" "-" "10.0.0.1:80" "200" "-"
  "-" TID_123456
```

Beispiel für HTTP/2-Eintrag

Es folgt ein Beispiel für einen Protokolleintrag für einen HTTP/2-Stream.

```
h2 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
10.0.1.252:48160 10.0.0.66:9000 0.000 0.002 0.000 200 200 5 257
"GET https://10.0.2.105:773/ HTTP/2.0" "curl/7.46.0" ECDHE-RSA-AES128-GCM-SHA256
  TLSv1.2
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337327-72bd00b0343d75b906739c42" "-" "-"
1 2018-07-02T22:22:48.364000Z "redirect" "https://example.com:80/" "-" "10.0.0.66:9000"
  "200" "-" "-"
```

Beispiel für WebSockets einen Eintrag

Im Folgenden finden Sie ein Beispiel für einen Protokolleintrag für eine WebSockets Verbindung.

```
ws 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
```

```

10.0.0.140:40914 10.0.1.192:8010 0.001 0.003 0.000 101 101 218 587
"GET http://10.0.0.30:80/ HTTP/1.1" "-" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
1 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.1.192:8010" "101" "-" "-"

```

Beispiel für einen gesicherten WebSockets Eintrag

Im Folgenden finden Sie ein Beispiel für einen Protokolleintrag für eine sichere WebSockets Verbindung.

```

wss 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
10.0.0.140:44244 10.0.0.171:8010 0.000 0.001 0.000 101 101 218 786
"GET https://10.0.0.30:443/ HTTP/1.1" "-" ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2
arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
1 2018-07-02T22:22:48.364000Z "forward" "-" "-" "10.0.0.171:8010" "101" "-" "-"

```

Beispieleinträge für Lambda-Funktionen

Es folgt ein Beispiel eines Protokolleintrags für eine Anfrage an eine Lambda-Funktion, die erfolgreich war:

```

http 2018-11-30T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 - 0.000 0.001 0.000 200 200 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"
0 2018-11-30T22:22:48.364000Z "forward" "-" "-" "-" "-" "-" "-"

```

Es folgt ein Beispiel eines Protokolleintrags für eine Anfrage an eine Lambda-Funktion, die fehlgeschlagen ist:

```

http 2018-11-30T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 - 0.000 0.001 0.000 502 - 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" - -
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067

```

```
"Root=1-58337364-23a8c76965a2ef7629b185e3" "-" "-"  
0 2018-11-30T22:22:48.364000Z "forward" "-" "LambdaInvalidResponse" "-" "-" "-" "
```

Verarbeiten von Zugriffsprotokolldateien

Die Zugriffsprotokolldateien werden komprimiert. Wenn Sie die Dateien mithilfe der Amazon-S3-Konsole öffnen, werden sie dekomprimiert und die Informationen werden angezeigt. Wenn Sie die Dateien herunterladen, müssen Sie sie dekomprimieren, um die Informationen anzuzeigen.

Falls es viele Zugriff auf Ihre Website gibt, kann der Load Balancer Protokolldateien mit mehreren Gigabyte an Daten generieren. Möglicherweise sind Sie nicht in der Lage, eine so große Datenmenge mithilfe von line-by-line Processing zu verarbeiten. Daher müssen Sie möglicherweise Tools zur Datenanalyse verwenden, die parallele Verarbeitungslösungen bieten. Beispielsweise können Sie die folgenden analytischen Tools zum Analysieren und Verarbeiten von Zugriffsprotokollen verwenden:

- Amazon Athena ist ein interaktiver Abfrageservice, der die Analyse von Daten in Amazon S3 mit Standard-SQL erleichtert. Weitere Informationen finden Sie unter [Abfragen von Application-Load-Balancer-Protokollen](#) im Benutzerhandbuch zu Amazon Athena.
- [Loggly](#)
- [Splunk](#)
- [Sumo Logic](#)

Aktivieren der Zugriffsprotokolle für Ihren Application Load Balancer

Wenn Sie die Zugriffsprotokolle für Ihren Load Balancer aktivieren, müssen Sie den Namen des S3-Bucket angeben, in dem der Load Balancer die Protokolle speichert. Der Bucket muss über eine Bucket-Richtlinie verfügen, die Elastic Load Balancing die Berechtigung zum Schreiben in den Bucket gewährt.

Aufgaben

- [Schritt 1: Einen S3-Bucket erstellen](#)
- [Schritt 2: Hinzufügen von Richtlinien zu Ihrem S3-Bucket](#)
- [Schritt 3: Konfigurieren von Zugriffsprotokollen](#)
- [Schritt 4: Überprüfen der Bucket-Berechtigungen](#)
- [Fehlerbehebung](#)

Schritt 1: Einen S3-Bucket erstellen

Wenn Sie Zugriffsprotokolle aktivieren, müssen Sie einen S3-Bucket für die Zugriffsprotokolle angeben. Sie können einen vorhandenen Bucket verwenden oder einen Bucket speziell für Zugriffsprotokolle erstellen. Der Bucket muss die folgenden Anforderungen erfüllen.

Voraussetzungen

- Der Bucket muss sich in derselben Region wie der Load Balancer befinden. Der Bucket und der Load Balancer können verschiedenen Konten gehören.
- Die einzige serverseitige Verschlüsselungsoption, die unterstützt wird, sind von Amazon S3 verwaltete Schlüssel (SSE-S3). Weitere Informationen finden Sie unter [Amazon-S3-verwaltete Verschlüsselungsschlüssel \(SSE-S3\)](#).

Erstellen eines S3-Buckets mithilfe der Amazon-S3-Konsole

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie Bucket erstellen aus.
3. Führen Sie auf der Seite Create bucket (Bucket erstellen) die folgenden Schritte aus:
 - a. Geben Sie unter Bucket Name (Bucket-Name) einen Namen für den Bucket ein. Dieser Name muss unter den in Amazon S3 vorhandenen Bucket-Namen eindeutig sein. In einigen Regionen kann es zusätzliche Einschränkungen für Bucket-Namen geben. Weitere Informationen finden Sie unter [Bucket-Einschränkungen und -Limits](#) im Benutzerhandbuch zu Amazon Simple Storage Service.
 - b. Wählen Sie unter AWS -Region die Region aus, in der Sie Ihren Load Balancer erstellt haben.
 - c. Wählen Sie für Standardverschlüsselung die Option Von Amazon S3 verwaltete Schlüssel (SSE-S3) aus.
 - d. Wählen Sie Bucket erstellen aus.

Schritt 2: Hinzufügen von Richtlinien zu Ihrem S3-Bucket

Der S3-Bucket muss über eine Bucket-Richtlinie verfügen, die Elastic Load Balancing die Berechtigung zum Schreiben von Zugriffsprotokollen in den Bucket gewährt. Bucket-Richtlinien sind eine Sammlung von JSON-Anweisungen, die in der Sprache der Zugriffsrichtlinie geschrieben sind,

um Zugriffsberechtigungen für Ihre Buckets zu definieren. Jeder Anweisung enthält Informationen über eine einzelne Berechtigung und besteht aus einer Reihe von Elementen.

Wenn Sie einen vorhandenen Bucket verwenden, dem bereits eine Richtlinie angehängt ist, können Sie die Anweisung für Zugriffsprotokolle von Elastic Load Balancing zu der Richtlinie hinzufügen. Wenn Sie dies tun, empfehlen wir, dass Sie eine Beurteilung der daraus resultierenden Berechtigungen vornehmen, um sicherzustellen, dass sie für die Benutzer geeignet sind, die Zugriff auf die Bucket-Zugriffsprotokolle benötigen.

Verfügbare Bucket-Richtlinien

Welche Bucket-Richtlinie Sie verwenden, hängt von der Zone und dem Typ der Zone ab. AWS-Region

Regionen, die ab August 2022 oder später verfügbar sind

Diese Richtlinie erteilt Berechtigungen für den angegebenen Protokoll-Bereitstellungsdienst. Verwenden Sie diese Richtlinie für Load Balancer in Availability Zones und Local Zonen in den folgenden Regionen:

- Asien-Pazifik (Hyderabad)
- Asien-Pazifik (Melbourne)
- Kanada West (Calgary)
- Europa (Spain)
- Europa (Zürich)
- Israel (Tel Aviv)
- Naher Osten (VAE)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*"
    }
  ]
}
```

```
}  
]  
}
```

Regionen, die vor August 2022 verfügbar sind

Diese Richtlinie gewährt Berechtigungen für die angegebene Konto-ID von Elastic Load Balancing. Verwenden Sie diese Richtlinie für Load Balancer in Availability Zones oder Local Zonen in den Regionen in der Liste unten.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::elb-account-id:root"  
      },  
      "Action": "s3:PutObject",  
      "Resource": "arn:aws:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*"  
    }  
  ]  
}
```

Ersetzen Sie *elb-account-id* durch die ID von AWS-Konto für Elastic Load Balancing für Ihre Region:

- USA Ost (Nord-Virginia) – 127311923021
- USA Ost (Ohio) – 033677994240
- USA West (Nordkalifornien) – 027434742980
- USA West (Oregon) – 797873946194
- Afrika (Kapstadt) – 098369216593
- Asien-Pazifik (Hongkong) – 754344448648
- Asien-Pazifik (Jakarta) – 589379963580
- Asien-Pazifik (Mumbai) – 718504428378
- Asien-Pazifik (Osaka) – 383597477331
- Asien-Pazifik (Seoul) – 600734575887

- Asien-Pazifik (Singapur) – 114774131450
- Asien-Pazifik (Sydney) – 783225319266
- Asien-Pazifik (Tokio) – 582318560864
- Kanada (Zentral) – 985666609251
- Europa (Frankfurt) – 054676820928
- Europa (Irland) – 156460612806
- Europa (London) – 652711504416
- Europa (Mailand) – 635631232127
- Europa (Paris) – 009996457667
- Europa (Stockholm) – 897822967062
- Naher Osten (Bahrain) – 076674570225
- Südamerika (São Paulo) – 507241528517

Ersetzen Sie *my-s3-arn* durch den ARN des Speicherorts für Ihre Zugriffsprotokolle. Der von Ihnen angegebene ARN hängt davon ab, ob Sie bei der Aktivierung von Zugriffsprotokollen in [Schritt 3](#) ein Präfix angeben möchten.

- ARN-Beispiel mit einem Präfix

```
arn:aws:s3::bucket-name/prefix/AWSLogs/aws-account-id/*
```

- ARN-Beispiel ohne Präfix

```
arn:aws:s3::bucket-name/AWSLogs/aws-account-id/*
```

Verwenden wann ist. `NotPrincipalEffectDeny`

Wenn die Amazon S3 S3-Bucket-Richtlinie `Effect` mit dem Wert `Deny` und dem `Include`-Wert verwendet, `NotPrincipal` wie im Beispiel unten gezeigt, stellen Sie sicher, dass dieser Wert in der Service Liste enthalten `logdelivery.elasticloadbalancing.amazonaws.com` ist.

```
{  
  "Effect": "Deny",  
  "NotPrincipal": {
```

```

    "Service": [
      "logdelivery.elasticloadbalancing.amazonaws.com",
      "example.com"
    ]
  }
},

```

AWS GovCloud (US) Regions

Diese Richtlinie gewährt Berechtigungen für die angegebene Konto-ID von Elastic Load Balancing. Verwenden Sie diese Richtlinie für Load Balancer in Availability Zones oder Local Zones in den AWS GovCloud (US) Regionen in der folgenden Liste.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws-us-gov:iam::elb-account-id:root"
      },
      "Action": "s3:PutObject",
      "Resource": "my-s3-arn"
    }
  ]
}

```

Ersetzen Sie *elb-account-id* durch die ID von AWS-Konto für Elastic Load Balancing für Ihre Region: AWS GovCloud (US)

- AWS GovCloud (US-West) — 048591011584
- AWS GovCloud (US-Ost) — 190560391635

Ersetzen Sie *my-s3-arn* durch den ARN des Speicherorts für Ihre Zugriffsprotokolle. Der von Ihnen angegebene ARN hängt davon ab, ob Sie bei der Aktivierung von Zugriffsprotokollen in [Schritt 3](#) ein Präfix angeben möchten.

- ARN-Beispiel mit einem Präfix

```
arn:aws-us-gov:s3::bucket-name/prefix/AWSLogs/aws-account-id/*
```

- ARN-Beispiel ohne Präfix

```
arn:aws-us-gov:s3::bucket-name/AWSLogs/aws-account-id/*
```

Outposts-Zonen

Die folgende Richtlinie erteilt Berechtigungen für den angegebenen Protokoll-Bereitstellungsdienst. Verwenden Sie diese Richtlinie für Load Balancer in Outposts-Zonen.

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "logdelivery.elb.amazonaws.com"
  },
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::bucket-name/prefix/AWSLogs/your-aws-account-id/*",
  "Condition": {
    "StringEquals": {
      "s3:x-amz-acl": "bucket-owner-full-control"
    }
  }
}
```

So fügen Sie Ihrem Bucket über die Amazon S3-Konsole eine Bucket-Richtlinie für Zugriffsprotokolle hinzu

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Buckets aus, um seine Detailseite zu öffnen.
3. Wählen Sie Berechtigungen und anschließend Bucket-Richtlinie, Bearbeiten aus.
4. Aktualisieren Sie die Bucket-Richtlinie, um die erforderlichen Berechtigungen zu gewähren.
5. Wählen Sie Änderungen speichern aus.

Schritt 3: Konfigurieren von Zugriffsprotokollen

Verwenden Sie das folgende Verfahren, um Zugriffsprotokolle für die Erfassung und Bereitstellung von Protokolldateien in Ihrem S3-Bucket zu konfigurieren.

Voraussetzungen

Der Bucket muss die in [Schritt 1](#) beschriebenen Anforderungen erfüllen und Sie müssen eine Bucket-Richtlinie wie in [Schritt 2](#) beschrieben anhängen. Wenn Sie ein Präfix angeben, darf es die Zeichenfolge "" nicht enthalten. AWSLogs

So aktivieren Sie die Zugriffsprotokolle für Ihren Load Balancer mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie im Navigationsbereich auf Load Balancers.
3. Wählen Sie den Namen Ihres Load Balancers aus, um die Detailseite zu öffnen.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Aktivieren Sie für die Überwachung die Option Zugriffsprotokolle.
6. Geben Sie für S3-URI den S3-URI für Ihre Protokolldateien ein. Der URI, den Sie angeben, hängt davon ab, ob Sie ein Präfix verwenden.
 - URI mit einem Präfix: `s3://bucket-name/prefix`
 - URI ohne Präfix: `s3://bucket-name`
7. Wählen Sie Änderungen speichern aus.

Um Zugriffsprotokolle mit dem zu aktivieren AWS CLI

Verwenden Sie den Befehl [modify-load-balancer-attributes](#).

So verwalten Sie den S3-Bucket für Ihre Zugriffsprotokolle

Stellen Sie sicher, dass Sie die Zugriffsprotokolle deaktivieren, bevor Sie den Bucket löschen, den Sie für Zugriffsprotokolle konfiguriert haben. Andernfalls kann Elastic Load Balancing keine Zugriffsprotokolle für Ihren Load Balancer in diesem neuen Bucket schreiben, wenn es einen neuen Bucket mit demselben Namen und den erforderlichen Bucket-Richtlinien in einem AWS-Konto gibt, dessen Eigentümer nicht Sie sind.

Schritt 4: Überprüfen der Bucket-Berechtigungen

Nachdem Zugriffsprotokolle für den Load Balancer aktiviert ist, überprüft Elastic Load Balancing den S3-Bucket und erstellt eine Testdatei, um sicherzustellen, dass die Bucket-Richtlinie die erforderlichen Berechtigungen angibt. Sie können die Amazon-S3-Konsole verwenden, um sicherzustellen, dass die Testdatei erstellt wurde. Die Testdatei ist keine tatsächliche Zugriffsprotokolldatei; sie enthält keine Beispieldatensätze.

So überprüfen Sie, ob Elastic Load Balancing eine Testdatei in Ihrem S3-Bucket erstellt hat

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Buckets aus, den Sie für Zugriffsprotokolle angegeben haben.
3. Navigieren Sie zur Testdatei, ELBAccessLogTestFile. Der Standort hängt davon ab, ob Sie ein Präfix verwenden.
 - Standort mit einem Präfix: *my-bucket/prefix/AWSLogs/123456789012/ELBAccessLogTestFile*
 - Standort ohne Präfix: *my-bucket/AWSLogs/123456789012/ELBAccessLogTestFile*

Fehlerbehebung

Wenn Sie einen Fehler aufgrund Zugriffsverweigerung erhalten, kann dies die folgenden möglichen Ursachen haben.

- Die Bucket-Policy gewährt Elastic Load Balancing nicht die Berechtigung, Zugriffsprotokolle in den Bucket zu schreiben. Stellen Sie sicher, dass Sie die richtige Bucket-Richtlinie für die Region verwenden. Stellen Sie sicher, dass der Ressourcen-ARN denselben Bucket-Namen verwendet, den Sie bei der Aktivierung von Zugriffsprotokollen angegeben haben. Stellen Sie sicher, dass der Ressourcen-ARN kein Präfix enthält, wenn Sie bei der Aktivierung von Zugriffsprotokollen kein Präfix angegeben haben.
- Der Bucket verwendet eine nicht unterstützte serverseitige Verschlüsselungsoption. Der Bucket muss von Amazon S3 verwaltete Schlüssel (SSE-S3) verwenden.

Deaktivieren der Zugriffsprotokolle für Ihren Application Load Balancer

Sie können die Zugriffsprotokolle für Ihren Load Balancer jederzeit deaktivieren. Nachdem Sie Zugriffsprotokolle deaktiviert haben, verbleiben Ihre Zugriffsprotokolle in Ihrem S3-Bucket, bis Sie sie löschen. Weitere Informationen finden Sie unter [Arbeiten mit Buckets](#) im Benutzerhandbuch für Amazon Simple Storage Service.

So deaktivieren Sie die Zugriffsprotokolle mithilfe der Konsole

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie im Navigationsbereich auf Load Balancers.
3. Wählen Sie den Namen Ihres Load Balancers aus, um die Detailseite zu öffnen.

4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Deaktivieren Sie für die Überwachung die Zugriffsprotokolle.
6. Wählen Sie Änderungen speichern aus.

Um Zugriffsprotokolle zu deaktivieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-load-balancer-attributes](#).

Verbindungsprotokolle für Ihren Application Load Balancer

Elastic Load Balancing stellt Verbindungsprotokolle bereit, die detaillierte Informationen über Anfragen erfassen, die an Ihren Load Balancer gesendet wurden. Jedes Protokoll enthält Informationen wie die IP-Adresse und den Port des Clients, den Listener-Port, die verwendete TLS-Chiffre und das verwendete Protokoll, die TLS-Handshake-Latenz, den Verbindungsstatus und Details zum Client-Zertifikat. Sie können diese Verbindungsprotokolle verwenden, um Anforderungsmuster zu analysieren und Probleme zu beheben.

Verbindungsprotokolle sind eine optionale Funktion von Elastic Load Balancing, die standardmäßig deaktiviert ist. Nachdem Sie Verbindungsprotokolle für Ihren Load Balancer aktiviert haben, erfasst Elastic Load Balancing die Protokolle und speichert sie in dem von Ihnen angegebenen Amazon S3 S3-Bucket als komprimierte Dateien. Sie können Verbindungsprotokolle jederzeit deaktivieren.

Sie zahlen Speicherkosten für Amazon S3, aber Sie zahlen nicht für die Bandbreite, die von Elastic Load Balancing zum Senden von Protokolldateien an Amazon S3 verwendet wird. Weitere Information zu Speicherkosten finden Sie unter [Amazon S3 – Preise](#).

Inhalt

- [Verbindungsprotokolldateien](#)
- [Verbindungsprotokolleinträge](#)
- [Beispiel-Protokolleinträge](#)
- [Verbindungsprotokolldateien werden verarbeitet](#)
- [Verbindungsprotokolle für Ihren Application Load Balancer aktivieren](#)
- [Verbindungsprotokolle für Ihren Application Load Balancer deaktivieren](#)

Verbindungsprotokolldateien

Elastic Load Balancing veröffentlicht alle 5 Minuten eine Protokolldatei für jeden Load-Balancer-Knoten. Die Protokollbereitstellung ist letztendlich konsistent. Der Load Balancer kann mehrere Protokolle für denselben Zeitraum bereitstellen. Dies passiert in der Regel, wenn die Website hohen Datenverkehr aufweist.

Die Dateinamen der Verbindungsprotokolle verwenden das folgende Format:

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/  
conn_log.aws-account-id_elasticloadbalancing_region_app.load-balancer-id_end-time_ip-  
address_random-string.log.gz
```

bucket

Der Name des S3-Buckets.

prefix

(Optional) Das Präfix (logische Hierarchie) für den Bucket. Das von Ihnen angegebene Präfix darf die Zeichenfolge AWSLogs nicht enthalten. Weitere Informationen finden Sie unter [Organisieren von Objekten mit Präfixen](#).

AWSLogs

Wir fügen den Teil des Dateinamens hinzu, der mit AWSLogs nach dem von Ihnen angegebenen Bucket-Namen und dem optionalen Präfix beginnt.

aws-account-id

Die AWS Konto-ID des Besitzers.

Region

Die Region für Ihren Load Balancer und den S3-Bucket.

JJJJ/MM/TT

Das Datum, an dem das Protokoll übermittelt wurde.

load-balancer-id

Die Ressourcen-ID des Load Balancer. Wenn die Ressourcen-ID Schrägstriche (/) enthält, werden sie durch Punkte (.) ersetzt.

end-time

Das Datum und die Uhrzeit, an dem das Protokollierungsintervall endete. Beispiel: Die Endzeit 20140215T2340Z enthält Einträge für Anforderungen, die zwischen 23:35 und 23:40 in UTC- oder Zulu-Zeit durchgeführt wurden.

ip-address

Die IP-Adresse des Load Balancer-Knotens, der die Anforderung verarbeitet hat. Für einen internen Load Balancer handelt es sich hierbei um eine private IP-Adresse.

random-string

Eine vom System generierte zufällige Zeichenfolge.

Im Folgenden finden Sie ein Beispiel für einen Protokolldateinamen mit Präfix:

```
s3://my-bucket/my-prefix/AWSLogs/123456789012/elasticloadbalancing/us-east-2/2022/05/01/conn_log.123456789012_elasticloadbalancing_us-east-2_app.my-loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Im Folgenden finden Sie ein Beispiel für einen Protokolldateinamen ohne Präfix:

```
s3://my-bucket/AWSLogs/123456789012/elasticloadbalancing/us-east-2/2022/05/01/conn_log.123456789012_elasticloadbalancing_us-east-2_app.my-loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Sie können Ihre Protokolldateien beliebig lange im Bucket speichern. Sie können aber auch Amazon S3-Lebenszyklusregeln aufstellen, anhand derer die Protokolldateien automatisch archiviert oder gelöscht werden. Weitere Informationen finden Sie unter [Objekt-Lebenszyklusmanagement](#) im Benutzerhandbuch zu Amazon Simple Storage Service.

Verbindungsprotokolleinträge

Jeder Verbindungsversuch hat einen Eintrag in einer Verbindungsprotokolldatei. Wie Client-Anfragen gesendet werden, hängt davon ab, ob die Verbindung persistent oder nicht persistent ist. Nicht persistente Verbindungen haben eine einzige Anfrage, wodurch ein einziger Eintrag im Zugriffs- und Verbindungsprotokoll erstellt wird. Persistente Verbindungen haben mehrere Anfragen, wodurch mehrere Einträge im Zugriffs- und ein einziger Eintrag im Verbindungsprotokoll erstellt werden.

Inhalt

- [Syntax](#)
- [Codes für die Fehlerursache](#)

Syntax

Die Einträge im Verbindungsprotokoll verwenden das folgende Format:

```
[timestamp] [client_ip] [client_port] [listener_port] [tls_protocol] [tls_cipher]
[tls_handshake_latency] [leaf_client_cert_subject] [leaf_client_cert_validity]
[leaf_client_cert_serial_number] [tls_verify_status]
```

In der folgenden Tabelle werden die Felder eines Verbindungsprotokolleintrags der Reihe nach beschrieben. Alle Felder werden durch Leerzeichen voneinander getrennt. Wenn neue Felder eingeführt werden, werden sie am Ende des Protokolleintrags hinzugefügt. Sie sollten alle Felder am Ende des Protokolleintrags ignorieren, die Sie nicht erwartet haben.

Feld	Beschreibung
Zeitstempel	Der Zeitpunkt im ISO 8601-Format, zu dem der Load Balancer erfolgreich eine Verbindung hergestellt hat oder nicht.
client_ip	Die IP-Adresse des anfragenden Clients.
client_port	Der Port des anfragenden Clients.
listener_port	Der Port des Load Balancer-Listeners, der die Client-Anfrage empfängt.
tls_protocol	[HTTPS-Listener] Das SSL/TLS-Protokoll, das bei Handshakes verwendet wird. Dieses Feld ist für Nicht-SSL/TLS-Anfragen auf - gesetzt.
tls_cipher	[HTTPS-Listener] Das bei Handshakes verwendete SSL/TLS-Protokoll. Dieses Feld ist für Nicht-SSL/TLS-Anfragen auf - gesetzt.
tls_handshake_latency	[HTTPS-Listener] Die Gesamtzeit in Sekunden, mit einer Genauigkeit von Millisekunden, die beim Aufbau eines erfolgreichen Handshakes verstrichen ist. Dieses Feld ist auf den Zeitpunkt gesetzt, wenn: - <ul style="list-style-type: none"> • Die eingehende Anfrage ist keine SSL/TLS-Anfrage.

Feld	Beschreibung
	<ul style="list-style-type: none"> • Der Handshake wurde nicht erfolgreich eingerichtet.
leaf_client_cert_subject	<p>[HTTPS-Listener] Der Betreffname des Leaf-Client-Zertifikats. Dieses Feld ist auf Folgendes gesetzt-:</p> <ul style="list-style-type: none"> • Die eingehende Anfrage ist keine SSL/TLS-Anfrage. • Der Load Balancer-Listener ist nicht mit aktiviertem mTLS konfiguriert. • Der Server ist nicht in der Lage, das Leaf-Client-Zertifikat zu laden/zu analysieren.
leaf_client_cert_validity	<p>[HTTPS-Listener] Die Gültigkeit des Leaf-Client-Zertifikats mit <code>not-before</code> und <code>not-after</code> im ISO 8601-Format. Dieses Feld ist auf Folgendes gesetzt-:</p> <ul style="list-style-type: none"> • Die eingehende Anfrage ist keine SSL/TLS-Anfrage. • Der Load Balancer-Listener ist nicht mit aktiviertem mTLS konfiguriert. • Der Server ist nicht in der Lage, das Leaf-Client-Zertifikat zu laden/zu analysieren.
leaf_client_cert_serial_number	<p>[HTTPS-Listener] Die Seriennummer des Leaf-Client-Zertifikats. Dieses Feld ist auf Folgendes gesetzt-:</p> <ul style="list-style-type: none"> • Die eingehende Anfrage ist keine SSL/TLS-Anfrage. • Der Load Balancer-Listener ist nicht mit aktiviertem mTLS konfiguriert. • Der Server ist nicht in der Lage, das Leaf-Client-Zertifikat zu laden/zu analysieren.
tls_verify_status	<p>[HTTPS-Listener] Der Status der Verbindungsanfrage. Dieser Wert gibt an <code>Success</code>, ob die Verbindung erfolgreich hergestellt wurde. Bei einer erfolglosen Verbindung ist der Wert <code>Failed:\$error_code</code>.</p>

Feld	Beschreibung
conn_trace_id	Die Verbindungsrückverfolgbarkeits-ID ist eine eindeutige und durchsichtige ID, die zur Identifizierung jeder Verbindung verwendet wird. Nachdem eine Verbindung mit einem Client hergestellt wurde, enthalten nachfolgende Anfragen von diesem Client diese ID in ihren jeweiligen Zugriffsprotokolleinträgen. Diese ID fungiert als Fremdschlüssel, um eine Verbindung zwischen den Verbindungs- und Zugriffsprotokollen herzustellen.

Codes für die Fehlerursache

Wenn der Load Balancer keine Verbindung herstellen kann, speichert der Load Balancer einen der folgenden Ursachencodes im Verbindungsprotokoll.

Code	Beschreibung
ClientCertificateMaximumChainDepthExceeded	Die maximale Tiefe der Client-Zertifikatskette wurde überschritten
ClientCertificateMaximumSizeExceeded	Die maximale Größe des Client-Zertifikats wurde überschritten
ClientCertificateRevoked	Das Client-Zertifikat wurde von der CA gesperrt
ClientCertificateRevocationProcessingError	Fehler bei der CRL-Verarbeitung
ClientCertificateUntrusted	Das Client-Zertifikat ist nicht vertrauenswürdig
ClientCertificateNotYetValid	Das Client-Zertifikat ist noch nicht gültig

Code	Beschreibung
ClientCertificateExpired	Das Client-Zertifikat ist abgelaufen
ClientCertificateTypeUnsupported	Der Typ des Client-Zertifikats wird nicht unterstützt
ClientCertificateInvalid	Das Client-Zertifikat ist ungültig
ClientCertificateRejected	Das Client-Zertifikat wurde durch die benutzerdefinierte Servervalidierung abgelehnt
UnmappedConnectionError	Fehler bei der Verbindung zur Laufzeit bei nicht zugeordneter Zuordnung

Beispiel-Protokolleinträge

Im Folgenden finden Sie Beispiele für Verbindungsprotokolleinträge.

Im Folgenden finden Sie ein Beispiel für einen Protokolleintrag für eine erfolgreiche Verbindung mit einem HTTPS-Listener, bei dem der Modus für die gegenseitige TLS-Überprüfung auf Port 443 aktiviert ist:

```
2023-10-04T17:05:15.514108Z 203.0.113.1 36280 443 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 4.036 "CN=amazondomains.com,O=endEntity,L=Seattle,ST=Washington,C=US"
NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z FEF257372D5C14D4 Success
```

Im Folgenden finden Sie ein Beispiel für einen Protokolleintrag für eine fehlgeschlagene Verbindung mit einem HTTPS-Listener, bei dem der gegenseitige TLS-Überprüfungsmodus auf Port 443 aktiviert ist. :

```
2023-10-04T17:05:15.514108Z 203.0.113.1 36280 443 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 - "CN=amazondomains.com,O=endEntity,L=Seattle,ST=Washington,C=US"
NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z FEF257372D5C14D4
Failed:ClientCertUntrusted
```

Verbindungsprotokolldateien werden verarbeitet

Die Verbindungsprotokolldateien sind komprimiert. Wenn Sie die Dateien mithilfe der Amazon-S3-Konsole öffnen, werden sie dekomprimiert und die Informationen werden angezeigt. Wenn Sie die Dateien herunterladen, müssen Sie sie dekomprimieren, um die Informationen anzuzeigen.

Falls es viele Zugriff auf Ihre Website gibt, kann der Load Balancer Protokolldateien mit mehreren Gigabyte an Daten generieren. Möglicherweise sind Sie nicht in der Lage, eine so große Datenmenge mithilfe von line-by-line Processing zu verarbeiten. Daher müssen Sie möglicherweise Tools zur Datenanalyse verwenden, die parallele Verarbeitungslösungen bieten. Sie können beispielsweise die folgenden Analysetools verwenden, um Verbindungsprotokolle zu analysieren und zu verarbeiten:

- Amazon Athena ist ein interaktiver Abfrageservice, der die Analyse von Daten in Amazon S3 mit Standard-SQL erleichtert.
- [Loggly](#)
- [Splunk](#)
- [Sumo Logic](#)

Verbindungsprotokolle für Ihren Application Load Balancer aktivieren

Wenn Sie Verbindungsprotokolle für Ihren Load Balancer aktivieren, müssen Sie den Namen des S3-Buckets angeben, in dem der Load Balancer die Protokolle speichert. Der Bucket muss über eine Bucket-Richtlinie verfügen, die Elastic Load Balancing die Berechtigung zum Schreiben in den Bucket gewährt.

Aufgaben

- [Schritt 1: Einen S3-Bucket erstellen](#)
- [Schritt 2: Hinzufügen von Richtlinien zu Ihrem S3-Bucket](#)
- [Schritt 3: Verbindungsprotokolle konfigurieren](#)
- [Schritt 4: Überprüfen der Bucket-Berechtigungen](#)
- [Fehlerbehebung](#)

Schritt 1: Einen S3-Bucket erstellen

Wenn Sie Verbindungsprotokolle aktivieren, müssen Sie einen S3-Bucket für die Verbindungsprotokolle angeben. Sie können einen vorhandenen Bucket verwenden oder einen

Bucket speziell für Verbindungsprotokolle erstellen. Der Bucket muss die folgenden Anforderungen erfüllen.

Voraussetzungen

- Der Bucket muss sich in derselben Region wie der Load Balancer befinden. Der Bucket und der Load Balancer können verschiedenen Konten gehören.
- Die einzige serverseitige Verschlüsselungsoption, die unterstützt wird, sind von Amazon S3 verwaltete Schlüssel (SSE-S3). Weitere Informationen finden Sie unter [Amazon-S3-verwaltete Verschlüsselungsschlüssel \(SSE-S3\)](#).

Erstellen eines S3-Buckets mithilfe der Amazon-S3-Konsole

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie Bucket erstellen aus.
3. Führen Sie auf der Seite Create bucket (Bucket erstellen) die folgenden Schritte aus:
 - a. Geben Sie unter Bucket Name (Bucket-Name) einen Namen für den Bucket ein. Dieser Name muss unter den in Amazon S3 vorhandenen Bucket-Namen eindeutig sein. In einigen Regionen kann es zusätzliche Einschränkungen für Bucket-Namen geben. Weitere Informationen finden Sie unter [Bucket-Einschränkungen und -Limits](#) im Benutzerhandbuch zu Amazon Simple Storage Service.
 - b. Wählen Sie unter AWS -Region die Region aus, in der Sie Ihren Load Balancer erstellt haben.
 - c. Wählen Sie für Standardverschlüsselung die Option Von Amazon S3 verwaltete Schlüssel (SSE-S3) aus.
 - d. Wählen Sie Bucket erstellen aus.

Schritt 2: Hinzufügen von Richtlinien zu Ihrem S3-Bucket

Ihr S3-Bucket muss über eine Bucket-Richtlinie verfügen, die Elastic Load Balancing die Erlaubnis erteilt, die Verbindungsprotokolle in den Bucket zu schreiben. Bucket-Richtlinien sind eine Sammlung von JSON-Anweisungen, die in der Sprache der Zugriffsrichtlinie geschrieben sind, um Zugriffsberechtigungen für Ihre Buckets zu definieren. Jeder Anweisung enthält Informationen über eine einzelne Berechtigung und besteht aus einer Reihe von Elementen.

Wenn Sie einen vorhandenen Bucket verwenden, dem bereits eine Richtlinie angehängt ist, können Sie der Richtlinie die Anweisung für Elastic Load Balancing Balancing-Verbindungsprotokolle hinzufügen. Wenn Sie dies tun, empfehlen wir Ihnen, die resultierenden Berechtigungen auszuwerten, um sicherzustellen, dass sie für die Benutzer geeignet sind, die Zugriff auf den Bucket für Verbindungsprotokolle benötigen.

Verfügbare Bucket-Richtlinien

Welche Bucket-Richtlinie Sie verwenden, hängt von der AWS-Region und der Art der Zone ab.

Regionen, die ab August 2022 oder später verfügbar sind

Diese Richtlinie erteilt Berechtigungen für den angegebenen Protokoll-Bereitstellungsdienst. Verwenden Sie diese Richtlinie für Load Balancer in Availability Zones und Local Zonen in den folgenden Regionen:

- Asien-Pazifik (Hyderabad)
- Asien-Pazifik (Melbourne)
- Europa (Spain)
- Europa (Zürich)
- Israel (Tel Aviv)
- Naher Osten (VAE)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*"
    }
  ]
}
```

Regionen, die vor August 2022 verfügbar sind

Diese Richtlinie gewährt Berechtigungen für die angegebene Konto-ID von Elastic Load Balancing. Verwenden Sie diese Richtlinie für Load Balancer in Availability Zones oder Local Zonen in den Regionen in der Liste unten.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::elb-account-id:root"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*"
    }
  ]
}
```

Ersetzen Sie *elb-account-id* durch die ID von AWS-Konto für Elastic Load Balancing für Ihre Region:

- USA Ost (Nord-Virginia) – 127311923021
- USA Ost (Ohio) – 033677994240
- USA West (Nordkalifornien) – 027434742980
- USA West (Oregon) – 797873946194
- Afrika (Kapstadt) – 098369216593
- Asien-Pazifik (Hongkong) – 754344448648
- Asien-Pazifik (Jakarta) – 589379963580
- Asien-Pazifik (Mumbai) – 718504428378
- Asien-Pazifik (Osaka) – 383597477331
- Asien-Pazifik (Seoul) – 600734575887
- Asien-Pazifik (Singapur) – 114774131450
- Asien-Pazifik (Sydney) – 783225319266
- Asien-Pazifik (Tokio) – 582318560864
- Kanada (Zentral) – 985666609251

- Europa (Frankfurt) – 054676820928
- Europa (Irland) – 156460612806
- Europa (London) – 652711504416
- Europa (Mailand) – 635631232127
- Europa (Paris) – 009996457667
- Europa (Stockholm) – 897822967062
- Naher Osten (Bahrain) – 076674570225
- Südamerika (São Paulo) – 507241528517
- AWS GovCloud (US-West) — 048591011584
- AWS GovCloud (US-Ost) — 190560391635

Ersetzen Sie *my-s3-arn* durch den ARN des Speicherorts für Ihre Verbindungsprotokolle. Der von Ihnen angegebene ARN hängt davon ab, ob Sie bei der Aktivierung von Verbindungsprotokollen in [Schritt 3](#) ein Präfix angeben möchten.

- ARN-Beispiel mit einem Präfix

```
arn:aws:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*
```

- ARN-Beispiel ohne Präfix

```
arn:aws:s3:::bucket-name/AWSLogs/aws-account-id/*
```

Verwenden **NotPrincipal**, wann **Effect** ist **Deny**.

Wenn die Amazon S3 S3-Bucket-Richtlinie Effect mit dem Wert Deny und dem Include-Wert verwendet, NotPrincipal wie im Beispiel unten gezeigt, stellen Sie sicher, dass dieser Wert in der Service Liste enthalten `logdelivery.elasticloadbalancing.amazonaws.com` ist.

```
{
  "Effect": "Deny",
  "NotPrincipal": {
    "Service": [
      "logdelivery.elasticloadbalancing.amazonaws.com",
      "example.com"
    ]
  },
}
```

So hängen Sie mithilfe der Amazon S3 S3-Konsole eine Bucket-Richtlinie für Verbindungsprotokolle an Ihren Bucket an

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Buckets aus, um seine Detailseite zu öffnen.
3. Wählen Sie Berechtigungen und anschließend Bucket-Richtlinie, Bearbeiten aus.
4. Aktualisieren Sie die Bucket-Richtlinie, um die erforderlichen Berechtigungen zu gewähren.
5. Wählen Sie Änderungen speichern aus.

Schritt 3: Verbindungsprotokolle konfigurieren

Gehen Sie wie folgt vor, um Verbindungsprotokolle für die Erfassung und Übermittlung von Protokolldateien an Ihren S3-Bucket zu konfigurieren.

Voraussetzungen

Der Bucket muss die in [Schritt 1](#) beschriebenen Anforderungen erfüllen und Sie müssen eine Bucket-Richtlinie wie in [Schritt 2](#) beschrieben anhängen. Wenn Sie ein Präfix angeben, darf es die Zeichenfolge "AWSLogs" nicht enthalten.

Um Verbindungsprotokolle für Ihren Load Balancer mithilfe der Konsole zu aktivieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie im Navigationsbereich auf Load Balancers.
3. Wählen Sie den Namen Ihres Load Balancers aus, um die Detailseite zu öffnen.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Aktivieren Sie für die Überwachung die Verbindungsprotokolle.
6. Geben Sie für S3 URI (S3-URI) den S3-URI für Ihre Protokolldateien ein. Der URI, den Sie angeben, hängt davon ab, ob Sie ein Präfix verwenden.
 - URI mit einem Präfix: `s3://bucket-name/prefix`
 - URI ohne Präfix: `s3://bucket-name`
7. Wählen Sie Änderungen speichern aus.

Um Verbindungsprotokolle zu aktivieren, verwenden Sie AWS CLI

Verwenden Sie den Befehl [modify-load-balancer-attributes](#).

Um den S3-Bucket für Ihre Verbindungsprotokolle zu verwalten

Stellen Sie sicher, dass Sie die Verbindungsprotokolle deaktivieren, bevor Sie den Bucket löschen, den Sie für Verbindungsprotokolle konfiguriert haben. Andernfalls könnte Elastic Load Balancing die Verbindungsprotokolle für Ihren Load Balancer in AWS-Konto diesen neuen Bucket schreiben, wenn es einen neuen Bucket mit demselben Namen und der erforderlichen Bucket-Richtlinie gibt, aber in einem Bucket erstellt wurde, den Sie nicht besitzen.

Schritt 4: Überprüfen der Bucket-Berechtigungen

Nachdem die Verbindungsprotokolle für Ihren Load Balancer aktiviert wurden, validiert Elastic Load Balancing den S3-Bucket und erstellt eine Testdatei, um sicherzustellen, dass die Bucket-Richtlinie die erforderlichen Berechtigungen festlegt. Sie können die Amazon-S3-Konsole verwenden, um sicherzustellen, dass die Testdatei erstellt wurde. Die Testdatei ist keine eigentliche Verbindungsprotokolldatei; sie enthält keine Beispieldatensätze.

So überprüfen Sie, ob Elastic Load Balancing eine Testdatei in Ihrem S3-Bucket erstellt hat

1. Öffnen Sie die Amazon-S3-Konsole unter <https://console.aws.amazon.com/s3/>.
2. Wählen Sie den Namen des Buckets aus, den Sie für Verbindungsprotokolle angegeben haben.
3. Navigieren Sie zur Testdatei, `ELBConnectionLogTestFile`. Der Standort hängt davon ab, ob Sie ein Präfix verwenden.
 - Standort mit einem Präfix: `my-bucket/prefix/AWSLogs/123456789012/ELBConnectionLogTestFile`
 - Standort ohne Präfix: `my-bucket/AWSLogs/123456789012/ELBConnectionLogTestFile`

Fehlerbehebung

Wenn Sie einen Fehler aufgrund Zugriffsverweigerung erhalten, kann dies die folgenden möglichen Ursachen haben.

- Die Bucket-Richtlinie gewährt Elastic Load Balancing nicht die Erlaubnis, Verbindungsprotokolle in den Bucket zu schreiben. Stellen Sie sicher, dass Sie die richtige Bucket-Richtlinie für die Region verwenden. Stellen Sie sicher, dass der Ressourcen-ARN denselben Bucket-Namen verwendet, den Sie bei der Aktivierung von Verbindungsprotokollen angegeben haben. Stellen Sie sicher, dass

der Ressourcen-ARN kein Präfix enthält, wenn Sie bei der Aktivierung von Verbindungsprotokollen kein Präfix angegeben haben.

- Der Bucket verwendet eine nicht unterstützte serverseitige Verschlüsselungsoption. Der Bucket muss von Amazon S3 verwaltete Schlüssel (SSE-S3) verwenden.

Verbindungsprotokolle für Ihren Application Load Balancer deaktivieren

Sie können die Verbindungsprotokolle für Ihren Load Balancer jederzeit deaktivieren. Nachdem Sie die Verbindungsprotokolle deaktiviert haben, verbleiben Ihre Verbindungsprotokolle in Ihrem S3-Bucket, bis Sie sie löschen. Weitere Informationen finden Sie unter [Arbeiten mit Buckets](#) im Benutzerhandbuch für Amazon Simple Storage Service.

Um Verbindungsprotokolle mit der Konsole zu deaktivieren

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Klicken Sie im Navigationsbereich auf Load Balancers.
3. Wählen Sie den Namen Ihres Load Balancers aus, um die Detailseite zu öffnen.
4. Klicken Sie auf der Registerkarte Attribute auf Bearbeiten.
5. Deaktivieren Sie für die Überwachung die Verbindungsprotokolle.
6. Wählen Sie Änderungen speichern aus.

Um Verbindungsprotokolle zu deaktivieren, verwenden Sie den AWS CLI

Verwenden Sie den Befehl [modify-load-balancer-attributes](#).

Anfragenachverfolgung für Ihren Application Load Balancer

Wenn der Load Balancer eine Anfrage von einem Client erhält, fügt er die Kopfzeile X-Amzn-Trace-Id hinzu oder aktualisiert diese, bevor er die Anfrage an das Ziel sendet. Jeder Service oder jede Anwendung zwischen dem Load Balancer und dem Ziel kann diese Kopfzeile ebenfalls hinzufügen oder aktualisieren.

Sie können Anfragenachverfolgung verwenden, um HTTP-Anfragen von Clients an Ziele oder andere Services nachzuverfolgen. Wenn Sie Zugriffsprotokolle aktivieren, wird der Inhalt der Kopfzeile X-Amzn-Trace-Id protokolliert. Weitere Informationen finden Sie unter [Zugriffsprotokolle für Ihre Application Load Balancer](#).

Syntax

Die Kopfzeile X-Amzn-Trace-Id enthält Felder mit folgendem Format:

```
Field=version-time-id
```

Feld

Der Name des Felds. Die unterstützten Werte sind Root und Self.

Eine Anwendung kann beliebige Felder für eigene Zwecke hinzufügen. Der Load Balancer behält diese Felder bei, verwendet sie jedoch nicht.

version

Die Versionsnummer.

time

Die Epoch-Zeit in Sekunden.

id

Die Nachverfolgungskennung.

Beispiele

Wenn die Kopfzeile X-Amzn-Trace-Id in einer eingehenden Anfrage nicht vorhanden ist, generiert der Load Balancer eine Kopfzeile mit einem Root-Feld und leitet die Anfrage weiter. Beispielsweise:

```
X-Amzn-Trace-Id: Root=1-67891233-abcdef012345678912345678
```

Wenn die X-Amzn-Trace-Id-Kopfzeile vorhanden ist und ein Root-Feld aufweist, fügt der Load Balancer ein Self-Feld ein und leitet die Anfrage weiter. Beispielsweise:

```
X-Amzn-Trace-Id: Self=1-67891233-12456789abcdef012345678;Root=1-67891233-abcdef012345678912345678
```

Wenn eine Anwendung eine Kopfzeile mit einem Root-Feld und ein benutzerdefiniertes Feld hinzufügt, behält der Load Balancer beide Felder bei, fügt ein Self-Feld hinzu und leitet die Anfrage weiter:

```
X-Amzn-Trace-Id: Self=1-67891233-12456789abcdef012345678;Root=1-67891233-abcdef012345678912345678;CalledFrom=app
```

Wenn die X-Amzn-Trace-Id-Kopfzeile vorhanden ist und ein Self-Feld aufweist, aktualisiert der Load Balancer den Wert des Self-Felds.

Einschränkungen

- Der Load Balancer aktualisiert die Kopfzeile, wenn er eine eingehende Anfrage erhält, nicht wenn er eine Antwort erhält.
- Wenn die HTTP-Header größer als 7 KB sind, schreibt der Load Balancer die Kopfzeile X-Amzn-Trace-Id neu mit einem Root-Feld.
- Mit können Sie nur verfolgen WebSockets, bis die Upgrade-Anfrage erfolgreich ist.

Protokollierung von API-Aufrufen für Ihren Application Load Balancer mit AWS CloudTrail

Elastic Load Balancing ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in Elastic Load Balancing ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für Elastic Load Balancing als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von AWS Management Console und Codeaufrufen an die Elastic Load Balancing API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für Elastic Load Balancing. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Event-Verlauf einsehen. Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an Elastic Load Balancing gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen CloudTrail dazu finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

Zum Überwachen von anderen Aktionen für Ihren Load Balancer, z. B. wenn ein Client eine Anforderung an Ihren Load Balancer sendet, verwenden Sie Zugriffsprotokolle. Weitere Informationen finden Sie unter [Zugriffsprotokolle für Ihre Application Load Balancer](#).

Informationen zu Elastic Load Balancing in CloudTrail

CloudTrail ist in Ihrem AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn in Elastic Load Balancing eine Aktivität auftritt, wird diese Aktivität zusammen mit anderen CloudTrail AWS Dienstereignissen in der Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter [Ereignisse mit CloudTrail Ereignisverlauf anzeigen](#).

Für eine fortlaufende Aufzeichnung von Ereignissen in Ihrem AWS Konto, einschließlich Ereignissen für Elastic Load Balancing, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Trail in der Konsole erstellen, gilt der Trail standardmäßig für alle AWS Regionen. Der Trail protokolliert Ereignisse aus allen Regionen in der AWS -Partition und stellt die Protokolldateien in dem von Ihnen angegebenen Amazon-S3-Bucket bereit. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail unterstützte Dienste und Integrationen](#)
- [Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#) und [Empfangen von CloudTrail Protokolldateien von mehreren Konten](#)

Alle Elastic Load Balancing-Aktionen für Application Load Balancers werden von der [Elastic Load Balancing API-Referenzversion 2015-12-01](#) protokolliert CloudTrail und sind in dieser Dokumentation dokumentiert. Aufrufe von `DeleteLoadBalancer` Aktionen `CreateLoadBalancer` und generieren beispielsweise Einträge in den CloudTrail Protokolldateien.

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 Anfrage mit Root-Anmeldeinformationen ausgeführt wurde
- 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.

Weitere Informationen finden Sie unter dem [CloudTrailUserIdentity-Element](#).

Verstehen der Elastic-Load-Balancing-Protokolldateieinträge

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält unter anderem Informationen über die angeforderte Aktion, das Datum und die Uhrzeit der Aktion sowie über die Anforderungsparameter. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Die Protokolldateien enthalten Ereignisse für alle AWS API-Aufrufe für Ihre AWS-Konto, nicht nur für Elastic Load Balancing API-Aufrufe. Sie können Aufrufe an die Elastic-Load-Balancing-API finden, indem Sie nach eventSource-Elementen mit dem Wert `elasticloadbalancing.amazonaws.com` suchen. Um einen Datensatz für eine bestimmte Aktion anzuzeigen, z. B. `CreateLoadBalancer`, suchen Sie nach eventName-Elementen mit dem Aktionsnamen.

Im Folgenden finden CloudTrail Sie Beispielprotokolldatensätze für Elastic Load Balancing für einen Benutzer, der einen Application 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: CreateLoadBalancer

```
{
  "eventVersion": "1.03",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:user/Alice",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Alice"
  },
  "eventTime": "2016-04-01T15:31:48Z",
  "eventSource": "elasticloadbalancing.amazonaws.com",
  "eventName": "CreateLoadBalancer",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "198.51.100.1",
```



```

"userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 boto3/1.4.1",
"requestParameters": {
  "subnets": ["subnet-8360a9e7","subnet-b7d581c0"],
  "securityGroups": ["sg-5943793c"],
  "name": "my-load-balancer",
  "scheme": "internet-facing"
},
"responseElements": {
  "loadBalancers": [{
    "type": "application",
    "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": "Z2P70J7HTTTPU",
    "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: DeleteLoadBalancer

```

{
  "eventVersion": "1.03",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:user/Alice",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",

```

```
    "userName": "Alice"
  },
  "eventTime": "2016-04-01T15:31:48Z",
  "eventSource": "elasticloadbalancing.amazonaws.com",
  "eventName": "DeleteLoadBalancer",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "198.51.100.1",
  "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
  "requestParameters": {
    "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/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"
}
```

Fehlerbehebung bei Ihren Application Load Balancern

Die folgenden Informationen können Ihnen helfen, Probleme bei Ihren Application Load Balancern zu beheben.

Problembereiche

- [Ein registriertes Ziel ist nicht in Betrieb](#)
- [Clients können keine Verbindung zu einem mit dem Internet verbundenen Load Balancer herstellen](#)
- [Anfragen, die an eine benutzerdefinierte Domain gesendet werden, werden vom Load Balancer nicht empfangen](#)
- [An den Load Balancer gesendete HTTPS-Anfragen geben „NET::ERR_CERT_COMMON_NAME_INVALID“ zurück](#)
- [Der Load Balancer zeigt erhöhte Verarbeitungszeiten an](#)
- [Der Load Balancer sendet als Antwortcode „000“.](#)
- [Der Load Balancer generiert einen HTTP-Fehler](#)
- [Ein Ziel generiert einen HTTP-Fehler](#)
- [Ein AWS Certificate Manager Zertifikat kann nicht verwendet werden](#)
- [Header mit mehreren Zeilen werden nicht unterstützt](#)
- [Beheben Sie fehlerhafte Ziele mithilfe der Ressourcenübersicht](#)

Ein registriertes Ziel ist nicht in Betrieb

Wenn es länger als erwartet dauert, bis ein Ziel den Zustand InService aufweist, besteht es möglicherweise Zustandsprüfungen nicht. Ihr Ziel ist erst betriebsbereit, wenn es eine Zustandsprüfung besteht. Weitere Informationen finden Sie unter [Zustandsprüfungen für Ihre Zielgruppen](#).

Überprüfen Sie, ob Ihre Instance Zustandsprüfungen nicht besteht und prüfen Sie dann Folgendes:

Eine Sicherheitsgruppe erlaubt keinen Datenverkehr

Die mit einer Instance verbundene Sicherheitsgruppe muss mithilfe des Zustandsprüfungs-Ports und Zustandsprüfungs-Protokolls Datenverkehr vom Load Balancer zulassen. Sie können eine Regel zur Sicherheitsgruppe der Instance hinzufügen, um den gesamten Datenverkehr von der

Load Balancer-Sicherheitsgruppe zuzulassen. Außerdem muss die Sicherheitsgruppe für Ihren Load Balancer Datenverkehr zu den Instances zulassen.

Eine Netzwerk-Zugriffskontrollliste (ACL) erlaubt keinen Datenverkehr

Die mit den Subnetzen verbundene Netzwerk-ACL für Ihre Instances muss eingehenden Datenverkehr am Zustandsprüfungs-Port und ausgehenden Datenverkehr an den flüchtigen Ports (1024-65535) zuzulassen. Die mit den Subnetzen verbundene Netzwerk-ACL für Ihre Load Balancer-Knoten muss eingehenden Datenverkehr an den flüchtigen Ports und ausgehenden Datenverkehr am Zustandsprüfungs-Port und an den flüchtigen Ports zulassen.

Der Ping-Pfad ist nicht vorhanden

Erstellen Sie eine Zielseite für die Zustandsprüfung und geben Sie den Pfad als den Ping-Pfad an.

Die Verbindung läuft ab

Vergewissern Sie sich zunächst, dass Sie mithilfe der privaten IP-Adresse des Ziels und des Zustandsprüfungs-Protokolls im Netzwerk eine direkte Verbindung zum Ziel herstellen können. Wenn Sie keine Verbindung herstellen können, prüfen Sie, ob die Instance überlastet ist, und fügen Sie weitere Ziele zu Ihrer Zielgruppe hinzu, wenn sie zu stark ausgelastet ist, um zu reagieren. Wenn Sie eine Verbindung herstellen können, ist es möglich, dass die Zielseite vor Ablauf des Zeitüberschreitungszeitraums der Zustandsprüfung nicht reagiert. Wählen Sie eine einfachere Zielseite für die Zustandsprüfung aus oder passen Sie die Zustandsprüfungseinstellungen an.

Das Ziel hat keinen Erfolgsmeldungscode zurückgegeben

Der Erfolgscode lautet standardmäßig 200. Sie können jedoch optional zusätzliche ErfolgsCodes angeben, wenn Sie Zustandsprüfungen konfigurieren. Überprüfen Sie die ErfolgsCodes, die der Load Balancer erwartet, und stellen Sie sicher, dass Ihre Anwendung so konfiguriert ist, dass sie diese Codes bei Erfolg zurückgibt.

Der Antwortcode des Ziels war fehlerhaft oder bei der Verbindung zum Ziel ist ein Fehler aufgetreten

Überprüfen Sie, ob Ihre Anwendung auf die Zustandsprüfungsanfragen des Load Balancers antwortet. Einige Anwendungen benötigen zusätzliche Konfigurationen, um auf Zustandsprüfungen zu reagieren, z. B. eine Konfiguration des virtuellen Hosts, um auf den vom Load Balancer gesendeten HTTP-Host-Header zu reagieren. Der Host-Header-Wert enthält die private IP-Adresse des Ziels, gefolgt vom Health Check-Port, wenn kein Standardport verwendet wird. Wenn das Ziel einen standardmäßigen Port für die Integritätsprüfung verwendet, enthält der Host-Header-Wert nur die private IP-Adresse des Ziels. Wenn die private IP-Adresse Ihres

Ziels beispielsweise lautet `10.0.0.10` und der Port für die Zustandsprüfung lautet, lautet der HTTP-Host-Header `8080`, der vom Load Balancer bei Integritätsprüfungen gesendet wird `Host : 10.0.0.10:8080`. Wenn die private IP-Adresse Ihres Ziels lautet `10.0.0.10` und der Port für die Zustandsprüfung der HTTP-Host-Header ist, ist `80` dies der HTTP-Host-Header, der vom Load Balancer bei Integritätsprüfungen gesendet wird. `Host : 10.0.0.10` Möglicherweise ist eine virtuelle Hostkonfiguration, die auf diesen Host reagiert, oder eine Standardkonfiguration erforderlich, um den Zustand Ihrer Anwendung zu überprüfen. Anfragen für Zustandsprüfungen haben die folgenden Attribute: `User-Agent` ist auf `ELB-HealthChecker/2.0` gesetzt, das Zeilenende für die Felder des Nachrichtenkopfes ist die Sequenz CRLF und der Header endet an der ersten leeren Zeile, gefolgt von einem CRLF.

Clients können keine Verbindung zu einem mit dem Internet verbundenen Load Balancer herstellen

Wenn der Load Balancer auf Anfragen nicht reagiert, überprüfen Sie Folgendes:

Ihr mit dem Internet verbundener Load Balancer ist mit einem privaten Subnetz verbunden.

Sie müssen öffentliche Subnetze für Ihren Load Balancer angeben. Ein öffentliches Subnetz verfügt über einen Zugang zum Internet-Gateway für Ihre Virtual Private Cloud (VPC).

Eine Sicherheitsgruppe oder Netzwerk-ACL erlaubt keinen Datenverkehr

Die Sicherheitsgruppe für den Load Balancer und jede Netzwerk-ACL für die Load Balancer-Subnetze muss an den Listener-Ports eingehenden Datenverkehr von den Clients und ausgehenden Datenverkehr zu den Clients erlauben.

Anfragen, die an eine benutzerdefinierte Domain gesendet werden, werden vom Load Balancer nicht empfangen

Wenn der Load Balancer keine Anfragen empfängt, die an eine benutzerdefinierte Domain gesendet werden, überprüfen Sie Folgendes:

Der benutzerdefinierte Domainname kann nicht in die IP-Adresse des Load Balancers aufgelöst werden

- Bestätigen Sie mithilfe einer Befehlszeilenschnittstelle, auf welche IP-Adresse der benutzerdefinierte Domainname aufgelöst wird.

- Linux, macOS oder Unix: Sie können den `dig`-Befehl im Terminal verwenden. Beispiel: `dig example.com`
- Windows: Sie können den `nslookup`-Befehl in der Eingabeaufforderung verwenden. Beispiel: `nslookup example.com`
- Bestätigen Sie die IP-Adresse, zu der der DNS-Name des Load Balancers über eine Befehlszeilenschnittstelle aufgelöst wird.
- Vergleichen Sie die Ergebnisse der beiden Ausgaben. Die IP-Adressen müssen übereinstimmen.

Weitere Informationen zur Verwendung von Route 53 zum Hosten Ihrer benutzerdefinierten Domain finden Sie unter [Meine Domain ist im Internet nicht verfügbar](#) im Entwicklerhandbuch zu Amazon Route 53.

An den Load Balancer gesendete HTTPS-Anfragen geben „NET::ERR_CERT_COMMON_NAME_INVALID“ zurück

Wird auf HTTPS-Anfragen vom Load Balancer `NET::ERR_CERT_COMMON_NAME_INVALID` zurückgegeben, überprüfen Sie die folgenden möglichen Ursachen:

- Der in der HTTPS-Anfrage verwendete Domainname stimmt nicht mit dem im ACM-Zertifikat des Listeners angegebenen alternativen Namen überein.
- Der Standard-DNS-Name des Load Balancers wird verwendet. Der Standard-DNS-Name kann nicht für HTTPS-Anfragen verwendet werden, da für die `*.amazonaws.com`-Domain kein öffentliches Zertifikat angefordert werden kann.

Der Load Balancer zeigt erhöhte Verarbeitungszeiten an

Der Load Balancer berechnet die Verarbeitungszeit je nach Konfiguration unterschiedlich.

- Wenn mit Ihrem Application Load Balancer verknüpft AWS WAF ist und ein Client eine HTTP-POST-Anfrage sendet, wird die Zeit zum Senden der Daten für POST-Anfragen in dem `request_processing_time` Feld in den Load Balancer-Zugriffsprotokollen wiedergegeben. Dieses Verhalten wird für HTTP-POST-Anfragen erwartet.
- Wenn AWS WAF es nicht mit Ihrem Application Load Balancer verknüpft ist und ein Client eine HTTP-POST-Anforderung sendet, wird die Zeit zum Senden der Daten für POST-Anfragen in

dem `target_processing_time` Feld in den Load Balancer-Zugriffsprotokollen wiedergegeben. Dieses Verhalten wird für HTTP-POST-Anfragen erwartet.

Der Load Balancer sendet als Antwortcode „000“.

Wenn bei HTTP/2-Verbindungen die komprimierte Länge eines der Header 8 K überschreitet oder wenn mehr als 10 000 Anfragen über eine Verbindung übermittelt werden, sendet der Load Balancer einen GOAWAY-Frame und schließt die Verbindung mit einer TCP-FIN.

Der Load Balancer generiert einen HTTP-Fehler

Die folgenden HTTP-Fehler werden vom Load Balancer generiert. Der Load Balancer sendet den HTTP-Code an den Client, speichert die Anforderung im Zugriffsprotokoll und erhöht die `HTTPCode_ELB_4XX_Count`- oder `HTTPCode_ELB_5XX_Count`-Metrik schrittweise.

Fehler

- [HTTP 400: Schlechte Anfrage](#)
- [HTTP 401: Unauthorized \(Nicht autorisiert\)](#)
- [HTTP 403: Forbidden \(Verboten\)](#)
- [HTTP 405: Methode nicht erlaubt](#)
- [HTTP 408: Anfrage-Timeout](#)
- [HTTP 413: Nutzlast zu hoch](#)
- [HTTP 414: URI zu lang](#)
- [HTTP 460](#)
- [HTTP 463](#)
- [HTTP 464](#)
- [HTTP 500: Interner Serverfehler](#)
- [HTTP 501: Nicht implementiert](#)
- [HTTP 502: Schlechtes Gateway](#)
- [HTTP 503: Dienst nicht verfügbar](#)
- [HTTP 504: Gateway-Timeout](#)
- [HTTP 505: Version wird nicht unterstützt](#)
- [HTTP 507: Nicht genügend Speicherplatz](#)

- [HTTP 561: Unauthorized \(Nicht autorisiert\)](#)

HTTP 400: Schlechte Anfrage

Mögliche Ursachen:

- Der Client hat eine falsch formatierte Anforderung gesendet, die die HTTP-Spezifikation nicht erfüllt.
- Der Anforderungsheader hat 16 K pro Anforderungszeile, 16 K pro individuelm Header oder 64 K pro gesamtem Anfrage-Header überschritten.
- Der Client hat die Verbindung beendet, bevor er den vollständigen Anfragetext gesendet hat.

HTTP 401: Unauthorized (Nicht autorisiert)

Sie haben eine Listener-Regel zum Authentifizieren von Benutzern konfiguriert, es trifft aber eine der folgenden Bedingungen zu:

- Entweder haben Sie `OnUnauthenticatedRequest` so konfiguriert, dass nicht authentifizierte Benutzer abgelehnt werden, oder der Identitätsanbieter hat den Zugriff abgelehnt.
- Die Größe der vom Identitätsanbieter zurückgegebenen Ansprüche überschreitet die maximale Größe, die vom Load Balancer unterstützt wird.
- Ein Client hat eine HTTP/1.0-Anfrage ohne Host-Header eingereicht und der Load Balancer konnte keine Umleitungs-URL generieren.
- Der angeforderte Bereich gibt kein ID-Token zurück.
- Sie schließen den Anmeldevorgang nicht ab, bevor das Client-Login-Timeout abgelaufen ist. Weitere Informationen finden Sie unter [Client-Login-Timeout](#).

HTTP 403: Forbidden (Verboten)

Sie haben eine AWS WAF Web Access Control List (Web ACL) konfiguriert, um Anfragen an Ihren Application Load Balancer zu überwachen, und dieser hat eine Anfrage blockiert.

HTTP 405: Methode nicht erlaubt

Der Client hat die TRACE-Methode verwendet, die von Application Load Balancern nicht unterstützt wird.

HTTP 408: Anfrage-Timeout

Der Client hat vor Ablauf des Zeitraums der Leerlaufzeitüberschreitung keine Daten gesendet. Durch Senden eines TCP-Keepalive-Pakets wird dieses Timeout nicht verhindert. Senden Sie mindestens 1 Datenbyte vor Ablauf jeder Leerlaufzeitüberschreitung. Erhöhen Sie bei Bedarf die Länge des Timeout-Limits.

HTTP 413: Nutzlast zu hoch

Mögliche Ursachen:

- Das Ziel ist eine Lambda-Funktion und der Anfragekörper übersteigt 1 MB.
- Der Anforderungsheader hat 16 K pro Anforderungszeile, 16 K pro einzeltem Header oder 64 K pro gesamtem Anfrage-Header überschritten.

HTTP 414: URI zu lang

Die Anfrage-URL oder Abfragezeichenfolgen-Parameter sind zu groß.

HTTP 460

Der Load Balancer hat eine Anforderung von einem Client erhalten, aber der Client hat die Verbindung mit dem Load Balancer getrennt, bevor die Leerlaufzeitüberschreitung abgelaufen war.

Überprüfen Sie, ob der Zeitraum der Client-Zeitüberschreitung größer ist als der Zeitraum der Leerlaufzeitüberschreitung für den Load Balancer. Stellen Sie sicher, dass Ihr Ziel eine Antwort an den Client liefert, bevor der Zeitraum der Client-Zeitüberschreitung verstreicht, oder erhöhen Sie den Zeitraum der Client-Zeitüberschreitung entsprechend der Leerlaufzeitüberschreitung des Load Balancer, wenn dies vom Client unterstützt wird.

HTTP 463

Der Load Balancer hat einen X-Forwarded-For-Anfrage-Header mit zu vielen IP-Adressen erhalten. Die Obergrenze für IP-Adressen liegt bei 30.

HTTP 464

Der Load Balancer hat ein eingehendes Anfrageprotokoll erhalten, das mit der Versionskonfiguration des Zielgruppenprotokolls nicht kompatibel ist.

Mögliche Ursachen:

- Das Anfrageprotokoll ist ein HTTP/1.1, während die Zielgruppenprotokollversion ein gRPC oder HTTP/2 ist.
- Das Anfrageprotokoll ist ein gRPC, während die Zielgruppenprotokollversion ein HTTP/1.1 ist.
- Das Anfrageprotokoll ist ein HTTP/2 und die Anfrage ist nicht POST, während die Zielgruppenprotokollversion ein gRPC ist.

HTTP 500: Interner Serverfehler

Mögliche Ursachen:

- Sie haben eine AWS WAF Web-Zugriffskontrollliste (Web-ACL) konfiguriert und bei der Ausführung der Web-ACL-Regeln ist ein Fehler aufgetreten.
- Der Load Balancer kann nicht mit dem Identitätsanbieter-Tokenendpunkt oder mit dem Endpunkt mit den Benutzerinformationen des Identitätsanbieters kommunizieren.
 - Überprüfen Sie, ob der DNS des IdP öffentlich auflösbar ist.
 - Stellen Sie sicher, dass die Sicherheitsgruppen für Ihren Load Balancer und die Netzwerk-ACLs für Ihren VPC den ausgehenden Zugriff auf diese Endpunkte zulassen.
 - Stellen Sie sicher, dass Ihre VPC auf das Internet zugreifen kann. Wenn Sie einen internen Load Balancer verwenden, aktivieren Sie den Internetzugriff mithilfe eines NAT-Gateways.
- Der vom IdP erhaltene Benutzeranspruch ist größer als 11 KB.

HTTP 501: Nicht implementiert

Der Load Balancer erhielt einen Transfer-Encoding (Transferverschlüsselung)-Header mit einem nicht unterstützten Wert. Die unterstützten Werte für Transfer-Encoding (Transferverschlüsselung) sind `chunked` und `identity`. Als Alternative können Sie den Content-Encoding (Inhaltsverschlüsselung)-Header verwenden.

HTTP 502: Schlechtes Gateway

Mögliche Ursachen:

- Der Load Balancer erhielt beim Versuch, eine Verbindung herzustellen, einen TCP-RST vom Ziel.

- Der Load Balancer hat beim Versuch, eine Verbindung herzustellen, eine unerwartete Antwort vom Ziel empfangen, z. B. "ICMP Destination unreachable (Host unreachable)" (ICMP-Ziel nicht erreichbar (Host nicht erreichbar)). Überprüfen Sie, ob Datenverkehr von den Subnetzen des Load Balancers an die Ziele im Ziel-Port zugelassen wird.
- Das Ziel hat die Verbindung mit einem TCP-RST oder einem TCP-FIN geschlossen, während der Load Balancer eine ausstehende Anforderung an das Ziel hatte. Überprüfen Sie, ob die Keep-Alive-Dauer des Ziels kürzer ist als der Timeoutwert für die Leerlaufzeit des Load Balancers.
- Die Zielantwort ist falsch formatiert oder enthält ungültige HTTP-Header.
- Der Ziel-Antwort-Header hat für den gesamten Antwort-Header 32 K überschritten.
- Die Verzögerungszeit der Registrierungsaufhebung einer Anfrage, die von einem Ziel verarbeitet wird, dessen Registrierung aufgehoben wurde, ist abgelaufen. Erhöhen Sie die Verzögerungszeit, sodass langwierige Vorgänge abgeschlossen werden können.
- Das Ziel ist eine Lambda-Funktion und der Anfragekörper übersteigt 1 MB.
- Das Ziel ist eine Lambda-Funktion, die nicht geantwortet hat, bevor die Zeitüberschreitung erreicht wurde.
- Das Ziel ist eine Lambda-Funktion, die einen Fehler zurückgegeben hat, oder die Funktion wurde vom Lambda-Service gedrosselt.
- Der Load Balancer ist beim Herstellen einer Verbindung zu einem Ziel auf einen SSL-Handshake-Fehler gestoßen.

Weitere Informationen finden Sie im AWS Support Knowledge Center unter [Wie behebe ich HTTP 502-Fehler im Application Load Balancer?](#)

HTTP 503: Dienst nicht verfügbar

Die Zielgruppen für den Load Balancer weisen keine registrierten Ziele auf.

HTTP 504: Gateway-Timeout

Mögliche Ursachen:

- Der Load Balancer konnte vor Ablauf des Verbindungstimeouts (10 Sekunden) keine Verbindung zum Ziel herstellen.
- Der Load Balancer hat eine Verbindung zum Ziel hergestellt, aber das Ziel hat nicht vor Ablauf des Verbindungstimeouts reagiert.

- Die Netzwerk-ACL für das Subnetz hat an den flüchtigen Ports (1024-65535) keinen Datenverkehr von den Zielen zu den Load Balancer-Knoten zugelassen.
- Das Ziel gibt einen Inhaltslängen-Header zurück, der größer ist als der Entitätskörper. Beim Warten auf die fehlenden Byte wurde das Zeitlimit des Load Balancer überschritten.
- Das Ziel ist eine Lambda-Funktion und der Lambda-Service hat vor Ablauf des Verbindungszeitlimits nicht reagiert.
- Der Load Balancer hat beim Herstellen einer Verbindung zu einem Ziel ein SSL-Handshake-Timeout (10 Sekunden) festgestellt.

HTTP 505: Version wird nicht unterstützt

Der Load Balancer hat eine unerwartete HTTP-Versionsanfrage erhalten. Beispielsweise hat der Load Balancer eine HTTP/1-Verbindung hergestellt, aber eine HTTP/2-Anfrage erhalten.

HTTP 507: Nicht genügend Speicherplatz

Die Weiterleitungs-URL ist zu lang.

HTTP 561: Unauthorized (Nicht autorisiert)

Sie haben eine Listener-Regel zum Authentifizieren von Benutzern konfiguriert, aber der Identitätsanbieter hat beim Authentifizieren des Benutzers einen Fehlercode zurückgegeben. Suchen Sie in Ihren Zugriffsprotokollen nach dem entsprechenden [Fehlerursachencode](#).

Ein Ziel generiert einen HTTP-Fehler

Der Load Balancer leitet gültige HTTP-Antworten von Zielen an den Client weiter, einschließlich HTTP-Fehlern. Die von einem Ziel generierten HTTP-Fehler werden in der HTTPCode_Target_4XX_Count- und der HTTPCode_Target_5XX_Count-Metrik aufgezeichnet.

Ein AWS Certificate Manager Zertifikat kann nicht verwendet werden

Wenn Sie sich für die Verwendung eines HTTPS-Listeners mit Ihrem Application Load Balancer entscheiden, AWS Certificate Manager müssen Sie vor der Ausstellung eines Zertifikats den

Domainbesitz überprüfen. Wenn dieser Schritt bei der Einrichtung versäumt wird, verbleibt das Zertifikat im Status Pending Validation und kann erst verwendet werden, wenn es validiert wurde.

- Wenn Sie die E-Mail-Validierung verwenden, finden Sie weitere Informationen unter [E-Mail-Validierung](#) im AWS Certificate Manager -Benutzerhandbuch.
- Weitere Informationen zur DNS-Validierung finden Sie unter [DNS-Validierung](#) im AWS Certificate Manager -Benutzerhandbuch.

Header mit mehreren Zeilen werden nicht unterstützt

Application Load Balancer unterstützen keine mehrzeiligen Header, einschließlich des message/http-Medientyp-Headers. Wenn ein mehrzeiliger Header bereitgestellt wird, hängt der Application Load Balancer einen Doppelpunkt „:“ an, bevor er ihn an das Ziel weitergibt.

Beheben Sie fehlerhafte Ziele mithilfe der Ressourcenübersicht

Wenn Ihre Application Load Balancer-Balancer-Ziele die Integritätsprüfungen nicht bestehen, können Sie die Ressourcenübersicht verwenden, um fehlerhafte Ziele zu finden und auf der Grundlage des Fehlerursachencodes Maßnahmen zu ergreifen. Weitere Informationen finden Sie unter [Ressourcenübersicht für Application Load Balancer](#).

Die Ressourcenübersicht bietet zwei Ansichten: „Übersicht“ und „Ungesunde Zielübersicht“. Overview ist standardmäßig ausgewählt und zeigt alle Ressourcen Ihres Load Balancers an. Wenn Sie die Ansicht Unhealthy Target Map auswählen, werden nur die fehlerhaften Ziele in jeder Zielgruppe angezeigt, die dem Application Load Balancer zugeordnet ist.

Note

Sie müssen die Option „Ressourcendetails anzeigen“ aktivieren, um die Zusammenfassung der Integritätsprüfung und die Fehlermeldungen für alle entsprechenden Ressourcen in der Ressourcenübersicht anzuzeigen. Wenn diese Option nicht aktiviert ist, müssen Sie jede Ressource auswählen, um ihre Details anzuzeigen.

In der Spalte Zielgruppen wird eine Zusammenfassung der gesunden und ungesunden Ziele für jede Zielgruppe angezeigt. Auf diese Weise kann festgestellt werden, ob alle Ziele die Zustandsprüfungen

nicht bestehen oder nur bestimmte Ziele nicht bestehen. Wenn alle Ziele in einer Zielgruppe die Integritätsprüfungen nicht bestehen, überprüfen Sie die Konfiguration der Zielgruppe. Wählen Sie den Namen einer Zielgruppe aus, um deren Detailseite in einem neuen Tab zu öffnen.

In der Spalte Ziele werden die TargetID und der aktuelle Status der Integritätsprüfung für jedes Ziel angezeigt. Wenn ein Ziel fehlerhaft ist, wird der Code für die Ursache des Fehlers bei der Integritätsprüfung angezeigt. Wenn ein einzelnes Ziel eine Integritätsprüfung nicht besteht, stellen Sie sicher, dass das Ziel über ausreichende Ressourcen verfügt, und stellen Sie sicher, dass die auf dem Ziel ausgeführten Anwendungen verfügbar sind. Wählen Sie die ID eines Ziels aus, um die zugehörige Detailseite in einer neuen Registerkarte zu öffnen.

Wenn Sie Exportieren auswählen, haben Sie die Möglichkeit, die aktuelle Ansicht der Ressourcenübersicht Ihres Application Load Balancers als PDF zu exportieren.

Stellen Sie sicher, dass Ihre Instance die Integritätsprüfungen nicht besteht, und überprüfen Sie dann anhand des Fehlerursachencodes auf die folgenden Probleme:

- Fehlerhaft: Die HTTP-Antwort stimmt nicht überein
 - Stellen Sie sicher, dass die auf dem Ziel ausgeführte Anwendung die richtige HTTP-Antwort auf die Integritätsprüfanforderungen des Application Load Balancers sendet.
 - Alternativ können Sie die Integritätsprüfanforderung des Application Load Balancers so aktualisieren, dass sie mit der Antwort der Anwendung übereinstimmt, die auf dem Ziel ausgeführt wird.
- Fehlerhaft: Das Zeitlimit für die Anfrage wurde überschritten
 - Stellen Sie sicher, dass die Sicherheitsgruppen und Network Access Control Lists (ACL), die Ihren Zielen und dem Application Load Balancer zugeordnet sind, die Konnektivität nicht blockieren.
 - Stellen Sie sicher, dass das Ziel über ausreichende Ressourcen verfügt, um Verbindungen vom Application Load Balancer anzunehmen.
 - Überprüfen Sie den Status aller Anwendungen, die auf dem Ziel ausgeführt werden.
 - Die Antworten des Application Load Balancers auf die Integritätsprüfung können in den Anwendungsprotokollen der einzelnen Ziele eingesehen werden. Weitere Informationen finden Sie unter [Ursachencodes für Gesundheitschecks](#).
- Ungesund: FailedHealthChecks
 - Überprüfen Sie den Status aller Anwendungen, die auf dem Ziel ausgeführt werden.
 - Stellen Sie sicher, dass das Ziel auf dem Health Check-Port auf Datenverkehr wartet.

i Bei Verwendung eines HTTPS-Listeners

Sie wählen aus, welche Sicherheitsrichtlinie für Front-End-Verbindungen verwendet wird. Die für Back-End-Verbindungen verwendete Sicherheitsrichtlinie wird automatisch auf der Grundlage der verwendeten Front-End-Sicherheitsrichtlinie ausgewählt.

- Wenn Ihr HTTPS-Listener eine TLS 1.3-Sicherheitsrichtlinie für Front-End-Verbindungen verwendet, wird die `ELBSecurityPolicy-TLS13-1-0-2021-06` Sicherheitsrichtlinie für Back-End-Verbindungen verwendet.
- Wenn Ihr HTTPS-Listener keine TLS 1.3-Sicherheitsrichtlinie für Front-End-Verbindungen verwendet, wird die `ELBSecurityPolicy-2016-08` Sicherheitsrichtlinie für Back-End-Verbindungen verwendet.

[Weitere Informationen finden Sie unter Sicherheitsrichtlinien.](#)

- Stellen Sie sicher, dass das Ziel ein Serverzertifikat und einen Schlüssel im richtigen Format bereitstellt, das in der Sicherheitsrichtlinie angegeben ist.
- Stellen Sie sicher, dass das Ziel eine oder mehrere übereinstimmende Chiffren und ein vom Application Load Balancer bereitgestelltes Protokoll zur Einrichtung von TLS-Handshakes unterstützt.

Kontingente für Ihre Application Load Balancer

Das AWS-Konto verfügt über Standardkontingente (früher als Limits bezeichnet) für jeden AWS-Service. Wenn nicht anders angegeben, gilt jedes Kontingent spezifisch für eine Region. Sie können Erhöhungen für einige Kontingente beantragen und andere Kontingente können nicht erhöht werden.

Um die Kontingente für Ihre Application Load Balancer anzuzeigen, öffnen Sie die [Konsole Service Quotas](#). Wählen Sie im Navigationsbereich AWS services (Dienste) und wählen Sie Elastic Load Balancing aus. Sie können auch den Befehl [describe-account-limits](#) (AWS CLI) für Elastic Load Balancing verwenden.

Informationen zur Erhöhung eines Kontingents finden Sie unter [Anfordern einer Kontingenterhöhung](#) im Service-Quotas-Benutzerhandbuch. Wenn das Kontingent unter Service Quotas noch nicht verfügbar ist, verwenden Sie das [Formular zur Erhöhung des Service-Limits für Elastic Load Balancing](#).

Load Balancers

Ihr AWS-Konto verfügt über die folgenden Kontingente in Bezug auf Application Load Balancers.

Name	Standard	Anpassbar
Application Load Balancer pro Region	50	Ja
Zertifikate pro Application Load Balancer (ausgenommen Standardzertifikate)	25	Ja
Listener pro Application Load Balancer	50	Ja
Zielgruppen pro Aktion und Application Load Balancer	5	Nein
Zielgruppen pro Application Load Balancer	100	Nein
Ziele pro Application Load Balancer	1.000	Ja

Zielgruppen

Die folgenden Kontingente gelten für Zielgruppen.

Name	Standard	Anpassbar
Zielgruppen pro Region	3.000*	Ja
Ziele pro Zielgruppe pro Region (Instances oder IP-Adressen)	1.000	Ja
Ziele pro Zielgruppe (Lambda-Funktionen)	1	Nein
Load Balancer pro Zielgruppe	1	Nein

* Dieses Kontingent wird von Application Load Balancern und Network Load Balancern geteilt.

Regeln

Die folgenden Kontingente gelten für Regeln.

Name	Standard	Anpassbar
Regeln pro Application Load Balancer (ohne Standardregeln)	100	Ja
Bedingungswerte pro Regel	5	Nein
Bedingungsplatzhalter pro Regel	5	Nein
Übereinstimmungsauswertungen pro Regel	5	Nein

Vertrauensspeicher

Die folgenden Kontingente gelten für Vertrauensspeicher.

Name	Standard	Anpassbar
Vertrauensspeicher pro Konto	20	Ja
Anzahl der Listener, die mTLS im Verifizierungsmodus verwenden, pro Load Balancer.	2	Nein

Zertifizierungsstellenzertifikate

Die folgenden Kontingente gelten für CA-Zertifikate.

Name	Standard	Anpassbar
CA-Zertifikate pro Trust Store	25	Ja
Größe des CA-Zertifikats	16KB	Nein
Maximale Tiefe der Zertifikatskette	4	Nein

Zertifikatsperrlisten

Die folgenden Kontingente gelten für Zertifikatsperrlisten.

Name	Standard	Anpassbar
Widerrufslisten pro Vertrauensspeicher	30	Ja
Widerrufseinträge pro Vertrauensspeicher	500 000	Ja
Dateigröße der Widerrufsliste	50MB	Nein

HTTP-Header

Für HTTP-Header gilt die folgende Größenbeschränkung.

Name	Standard	Anpassbar
Anforderungszeile	16 K	Nein
Einzelner Header	16 K	Nein
Gesamter Antwort-Header	32 K	Nein
Gesamter Anfrage-Header	64 K	Nein

Dokumentverlauf für Application Load Balancer

In der folgenden Tabelle werden die Versionen für Application Load Balancer beschrieben.

Änderung	Beschreibung	Datum
Karte der Ressourcen	Diese Version bietet Unterstützung für die Anzeige Ihrer Load Balancer-Ressourcen und -Beziehungen in einem visuellen Format.	8. März 2024
WAF mit einem Klick	Diese Version bietet Unterstützung für die Konfiguration des Verhaltens Ihres Load Balancers, wenn er mit einem Klick integriert wird. AWS WAF	6. Februar 2024
Gegenseitiges TLS	Diese Version bietet Unterstützung für die gegenseitige TLS-Authentifizierung.	26. November 2023
Automatische Zielgewichte	Diese Version bietet Unterstützung für den Algorithmus für automatische Zielgewichte.	26. November 2023
FIPS 140-3 TLS-Terminierung	Diese Version fügt Sicherheitsrichtlinien hinzu, die beim Beenden von TLS-Verbindungen kryptografische FIPS 140-3-MODULE verwenden.	20. November 2023
Registrieren Sie Ziele mithilfe von IPv6	Diese Version bietet Unterstützung für die Registrierung von Instances als Ziele, wenn sie über IPv6 adressiert werden.	2. Oktober 2023

Sicherheitsrichtlinien, die TLS 1.3 unterstützen	Diese Version bietet Unterstützung für vordefinierte Sicherheitsrichtlinien mit TLS 1.3.	22. März 2023
Zonale Verschiebung	Diese Version bietet Unterstützung für die Weiterleitung von Datenverkehr von einer einzelnen beeinträchtigten Availability Zone weg durch die Integration mit der Amazon Route 53 Application Recovery Controller.	28. November 2022
Schalten Sie den zonenübergreifenden Lastenausgleich aus	Diese Version bietet Unterstützung für die Deaktivierung des zonenübergreifenden Lastenausgleichs.	28. November 2022
Zustand der Zielgruppe	Mit dieser Version lässt sich die Mindestanzahl oder der Prozentsatz der Ziele konfigurieren, die fehlerfrei sein müssen. Außerdem können Sie festlegen, welche Maßnahmen der Load Balancer ergreift, wenn der Schwellenwert nicht erreicht wird.	28. November 2022
Zonenübergreifendes Load Balancing	Diese Version bietet Unterstützung für die Konfiguration des zonenübergreifenden Lastenausgleichs auf Zielgruppenebene.	17. November 2022

IPv6-Zielgruppen	Diese Version bietet Unterstützung für die Konfiguration von IPv6-Zielgruppen für Application Load Balancer.	23. November 2021
Interne IPv6-Loadbalancer	Diese Version bietet Unterstützung für die Konfiguration von IPv6-Zielgruppen für Application Load Balancer.	23. November 2021
AWS PrivateLink und statische IP-Adressen	Diese Version bietet Unterstützung für die Verwendung AWS PrivateLink und Offenlegung statischer IP-Adressen, indem der Datenverkehr direkt von Network Load Balancers an Application Load Balancers weitergeleitet wird.	27. September 2021
Erhaltung des Client-Ports	In dieser Version wurde ein Attribut hinzugefügt, um den Quellport zu erhalten, den der Client zur Verbindung mit dem Load Balancer verwendet hat.	29. Juli 2021
TLS-Header	In dieser Version wird ein Attribut hinzugefügt, das angibt, dass die TLS-Header, die Informationen über die ausgehandelte TLS-Version und die Verschlüsselungssuite enthalten, der Client-Anfrage hinzugefügt werden, bevor sie an das Ziel gesendet wird.	21. Juli 2021

Zusätzliche ACM-Zertifikate	Diese Version unterstützt RSA-Zertifikate mit Schlüssel­längen von 2048, 3072 und 4096 Bit sowie alle ECDSA-Zertifikate.	14. Juli 2021
Anwendungsbasierte Sticky Sessions	Diese Version fügt ein anwendungsbasiertes Cookie hinzu, um Sticky Sessions für Ihren Load Balancer zu unterstützen.	8. Februar 2021
Sicherheitsrichtlinie für FS mit Unterstützung für TLS Version 1.2	Diese Version umfasst eine Sicherheitsrichtli­nie für Forward Secrecy (FS) mit Unterstützung für TLS Version 1.2.	24. November 2020
Unterstützung für WAF Fail Open	Diese Version bietet Unterstüt­zung für die Konfiguration des Verhaltens Ihres Load Balancers, wenn dieser mit integriert wird. AWS WAF	13. November 2020
Unterstützung für gRPC und HTTP/2	Diese Version bietet Unterstüt­zung für gRPC-Workloads und end-to-end HTTP/2.	29. Oktober 2020
Outpost-Unterstützung	Sie können einen Application Load Balancer auf Ihrem AWS Outposts bereitstellen.	8. September 2020
Desynchroner Mitigatio­nsmodus	Diese Version bietet Unterstüt­zung für den desynchronen Mitigationsmodus.	17. August 2020

Am wenigsten ausstehende Anfragen	Mit dieser Version wird die Unterstützung für den Algorithmus „Am wenigsten ausstehende Anfragen“ hinzugefügt.	25. November 2019
Gewichtete Zielgruppen	Diese Version bietet neue Unterstützung für Weiterleitungsaaktionen mit mehreren Zielgruppen. Anforderungen werden basierend auf der Gewichtung, die Sie für jede Zielgruppe angeben, an diese Zielgruppen verteilt.	19. November 2019
New Attribute	Diese Version bietet neue Unterstützung für das Attribut <code>routing.http.drop_invalid_header_fields.enabled</code> .	15. November 2019
Sicherheitsrichtlinien für FS	Diese Version bietet Unterstützung für drei weitere vordefinierte Forward Secrecy-Sicherheitsrichtlinien.	8. Oktober 2019
Erweiterte Anfrageweiterleitung	In dieser Version wurde Unterstützung für weitere Bedingungstypen für Ihre Listener-Regeln hinzugefügt.	27. März 2019
Lambda-Funktionen als Ziel	Diese Version fügt Unterstützung für die Registrierung von Lambda-Funktionen als Ziel hinzu.	29. November 2018

Weiterleitungsaktionen	In dieser Version wurde Support für den Load Balancer zum Weiterleiten von Anfragen an eine andere URL hinzugefügt.	25. Juli 2018
Aktionen mit feststehender Antwort	In dieser Version wurde Support für den Load Balancer zum Zurückgeben benutzerdefinierter HTTP-Antworten hinzugefügt.	25. Juli 2018
Sicherheitsrichtlinien für FS und TLS 1.2	Mit dieser Version wird die Unterstützung für zwei zusätzliche vordefinierte Sicherheitsrichtlinien hinzugefügt.	6. Juni 2018
Benutzerauthentifizierung	Mit dieser Version wird für den Load Balancer die Unterstützung zum Authentifizieren von Benutzern Ihrer Anwendungen mit ihren Unternehmensidentitäten oder Social Identities vor dem Weiterleiten von Anfragen hinzugefügt.	30. Mai 2018
Berechtigungen auf Ressourcenebene	Mit dieser Version wird Unterstützung für Berechtigungen auf Ressourcenebene und Bedingungsschlüssel für das Markieren hinzugefügt.	10. Mai 2018

Langsamer Startmodus	Diese Version bietet Unterstützung für den Modus des langsamen Hochfahrens, der den Anteil an Anfragen, die vom Load Balancer an ein neu registriertes Ziel gesendet werden, während dieses warmläuft, schrittweise erhöht.	24. März 2018
SNI-Unterstützung	In dieser Version wurde SNI-Unterstützung (Server Name Indication) hinzugefügt.	10. Oktober 2017
IP-Adressen als Ziele	Diese Version bietet Unterstützung für die Registrierung von IP-Adressen als Ziele.	31. August 2017
Host-basiertes Routing	Mit dieser Version wird Unterstützung für Weiterleitungsanfragen basierend auf den Host-Namen im Host-Header hinzugefügt.	5. April 2017
Sicherheitsrichtlinien für TLS 1.1 und TLS 1.2	Mit dieser Version werden Sicherheitsrichtlinien für TLS 1.1 und TLS 1.2 eingeführt.	6. Februar 2017
IPv6-Support	Diese Version bietet Unterstützung für IPv6-Adressen.	25. Januar 2017
Ablaufverfolgung anfordern	Diese Version bietet Unterstützung für Anfragenachverfolgung.	22. November 2016

[Unterstützung von Perzentilen für die Metrik TargetResponseTime](#)

Diese Version bietet Unterstützung für die neuen Perzentilstatistiken, die von Amazon unterstützt werden. CloudWatch

17. November 2016

[Neuer Typ von Load Balancern](#)

In dieser Version von Elastic Load Balancing werden Application Load Balancer eingeführt.

11. August 2016

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.