



Application Load Balancer

# Elastic Load Balancing



# Elastic Load Balancing: Application Load Balancer

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

---

# Table of Contents

Apa itu Application Load Balancer? .....	1
Komponen Application Load Balancer .....	1
Gambaran umum Application Load Balancer .....	2
Manfaat migrasi dari Classic Load Balancer .....	3
Layanan terkait .....	4
Harga .....	5
Application Load Balancer .....	6
Subnet untuk penyeimbang beban Anda .....	7
Subnet Zona Ketersediaan .....	7
Subnet Zona Lokal .....	8
Subnet pos terdepan .....	8
Grup keamanan penyeimbang beban .....	10
Status penyeimbang beban .....	10
Atribut penyeimbang beban .....	11
Jenis alamat IP .....	13
Manajemen Alamat IP Application Load Balancer .....	14
Kumpulan alamat IP IPAM .....	15
Koneksi penyeimbang beban .....	16
Penyeimbangan beban lintas zona .....	16
Nama DNS .....	16
Membuat penyeimbang beban .....	17
Prasyarat .....	18
Buat penyeimbang beban .....	18
Uji penyeimbang beban .....	23
Langkah selanjutnya .....	24
Memperbarui Availability Zone .....	24
Memperbarui grup keamanan .....	26
Aturan yang disarankan .....	26
Memperbarui grup keamanan terkait .....	28
Perbarui jenis alamat IP .....	30
Perbarui kumpulan alamat IP IPAM .....	31
Edit atribut penyeimbang beban .....	33
Batas waktu idle koneksi .....	33
Durasi keepalive klien HTTP .....	35

---

Perlindungan penghapusan .....	37
Mode mitigasi desync .....	39
Pelestarian header host .....	41
Tandai penyeimbang beban .....	44
Menghapus penyeimbang beban .....	46
Lihat peta sumber daya .....	47
Komponen peta sumber daya .....	48
Peralihan zona .....	49
Sebelum Anda mulai .....	50
Penyeimbangan beban lintas zona .....	50
Pengesampingan administratif .....	51
Aktifkan pergeseran zona .....	51
Memulai peralihan zona .....	53
Perbarui pergeseran zona .....	54
Batalkan pergeseran zona .....	55
Reservasi LCU .....	56
Permintaan reservasi .....	57
Perbarui atau batalkan reservasi .....	59
Pantau reservasi .....	60
Integrasi penyeimbang beban .....	61
Pengontrol Pemulihan Aplikasi Amazon (ARC) .....	61
Amazon CloudFront + AWS WAF .....	61
AWS Global Accelerator .....	62
AWS Config .....	63
AWS WAF .....	63
Pendengar dan aturan .....	65
Konfigurasi listener .....	65
Atribut pendengar .....	66
Tindakan default .....	69
Membuat listener HTTP .....	69
Prasyarat .....	69
Menambahkan listener HTTP .....	69
Sertifikat SSL .....	72
Sertifikat default .....	73
Daftar sertifikat .....	73
Perpanjangan sertifikat .....	74

Kebijakan keamanan .....	75
Contoh describe-ssl-policies perintah .....	78
Kebijakan keamanan TLS .....	79
Kebijakan keamanan FIPS .....	110
Kebijakan yang didukung FS .....	134
Buat listener HTTPS .....	140
Prasyarat .....	140
Menambahkan pendengar HTTPS .....	141
Memperbarui listener HTTPS .....	144
Mengganti sertifikat default .....	144
Menambahkan sertifikat ke daftar sertifikat .....	145
Menghapus sertifikat dari daftar sertifikat .....	147
Memperbarui kebijakan keamanan .....	148
Modifikasi header HTTP .....	150
Aturan pendengar .....	150
Jenis tindakan .....	151
Jenis kondisi .....	160
Mengubah .....	167
Tambahkan peraturan .....	170
Mengedit peraturan .....	176
Menghapus peraturan .....	182
Autentikasi TLS bersama .....	183
Sebelum Anda mulai .....	183
Header HTTP .....	186
Iklankan nama subjek CA .....	188
Log koneksi .....	188
Konfigurasi TLS timbal balik .....	189
Bagikan toko kepercayaan .....	197
Otentikasi pengguna .....	202
Bersiap menggunakan IdP yang sesuai dengan OID .....	203
Bersiap menggunakan Amazon Cognito .....	203
Bersiaplah untuk menggunakan Amazon CloudFront .....	205
Mengonfigurasi autentikasi pengguna .....	206
Alur autentikasi .....	209
Pengkodean klaim pengguna dan verifikasi tanda tangan .....	211
Waktu habis .....	213

---

Logout autentikasi .....	214
Verifikasi JWT .....	214
Bersiaplah untuk menggunakan verifikasi JWT .....	215
Batas validasi JWT .....	215
Untuk mengkonfigurasi verifikasi JWT menggunakan CLI .....	216
Header X-diteruskan .....	218
X-Diteruskan-Untuk .....	219
X-Diteruskan-Proto .....	223
Port-X-Diteruskan .....	224
Modifikasi header HTTP .....	224
Ganti nama header mTLS/TLS .....	224
Tambahkan header respons .....	226
Nonaktifkan header .....	228
Batasan .....	228
Aktifkan modifikasi header .....	228
Menghapus listener .....	232
Kelompok-kelompok target .....	234
Konfigurasi perutean .....	235
Tipe target .....	235
Jenis alamat IP .....	237
Versi protokol .....	238
Target-target terdaftar .....	239
Pengoptimal Target .....	240
Atribut grup target .....	240
Kesehatan kelompok sasaran .....	243
Tindakan negara yang tidak sehat .....	243
Persyaratan dan pertimbangan .....	243
Memantau .....	245
Contoh .....	245
Menggunakan failover DNS Route 53 untuk menyeimbangkan beban Anda .....	246
Buat grup target .....	247
Konfigurasi pemeriksaan kondisi .....	251
Pengaturan pemeriksaan kondisi .....	252
Status kondisi target .....	254
Kode alasan pemeriksaan kondisi .....	255
Periksa kesehatan target .....	257

Perbarui pengaturan pemeriksaan kesehatan .....	259
Edit atribut grup target .....	260
Penundaan Pembatalan Pendaftaran .....	261
Algoritma perutean .....	262
Mode mulai lambat .....	265
Pengaturan Kesehatan .....	267
Penyeimbangan beban lintas zona .....	268
Bobot Target Otomatis (ATW) .....	272
Sesi lengket .....	276
Daftarkan Target-target. ....	283
Menargetkan grup keamanan .....	284
Pengoimal Target .....	285
Subnet bersama .....	286
Daftarkan target .....	286
Target deregister .....	289
Gunakan fungsi Lambda sebagai target .....	290
Siapkan fungsi Lambda .....	291
Buat grup target untuk fungsi Lambda .....	292
Menerima peristiwa dari load balancer .....	293
Menanggapi load balancer .....	294
Header nilai ganda .....	295
Aktifkan pemeriksaan kesehatan .....	299
Daftarkan fungsi Lambda .....	301
Deregistrasi fungsi Lambda .....	302
Menandai grup sasaran .....	303
Menghapus grup target .....	305
Memantau penyeimbang beban Anda .....	306
CloudWatch metrik .....	307
Metrik Application Load Balancer .....	308
Dimensi metrik untuk Application Load Balancer .....	333
Statistik untuk metrik Application Load Balancer .....	333
Lihat CloudWatch metrik untuk penyeimbang beban .....	335
Log akses .....	337
Berkas log akses .....	338
Entri log akses .....	339
Contoh Entri log .....	357

Konfigurasi pemberitahuan pengiriman log .....	359
Memproses berkas log akses .....	360
Aktifkan log akses .....	360
Nonaktifkan log akses .....	370
Log koneksi .....	370
File log koneksi .....	371
Entri log koneksi .....	373
Contoh Entri log .....	377
Memproses file log koneksi .....	377
Aktifkan log koneksi .....	378
Nonaktifkan log koneksi .....	386
Log pemeriksaan kesehatan .....	387
File log pemeriksaan kesehatan .....	387
Entri log pemeriksaan kesehatan .....	389
Contoh Entri log .....	392
Konfigurasi pemberitahuan pengiriman log .....	392
Memproses file log pemeriksaan kesehatan .....	392
Aktifkan log pemeriksaan kesehatan .....	393
Nonaktifkan log pemeriksaan kesehatan .....	401
Pelacakan permintaan .....	402
Sintaksis .....	402
Batasan .....	403
Memecahkan masalah Load Balancer .....	404
Target terdaftar tidak dalam layanan .....	404
Klien tidak dapat menyambung ke Load Balancer yang menghadap internet .....	406
Permintaan yang dikirim ke domain kustom tidak diterima oleh penyeimbang beban .....	406
Permintaan HTTPS yang dikirim ke penyeimbang beban mengembalikan “NET: :ERR_CERT_COMMON_NAME_INVALID” .....	407
Load balancer menunjukkan peningkatan waktu pemrosesan .....	407
Load Balancer mengirimkan kode respon 000 .....	408
Load Balancer menghasilkan kesalahan HTTP .....	408
HTTP 400: Permintaan buruk .....	409
HTTP 401: Tidak sah .....	409
HTTP 403: Terlarang .....	410
HTTP 405: Metode tidak diperbolehkan .....	410
HTTP 408: Waktu habis permintaan .....	410

HTTP 413: Muatan terlalu besar .....	410
HTTP 414: URI terlalu panjang .....	410
HTTP 460 .....	410
HTTP 463 .....	411
HTTP 464 .....	411
HTTP 500: Kesalahan peladen internal .....	411
HTTP 501: Tidak diimplementasikan .....	412
HTTP 502: Gateway buruk .....	412
503 Layanan Tidak Tersedia .....	413
HTTP 504: Waktu habis gateway .....	413
HTTP 505: Versi tidak didukung .....	414
HTTP 507: Penyimpanan Tidak Cukup .....	414
HTTP 561: Tidak sah .....	414
HTTP 562: Permintaan JWKS Gagal .....	414
Target menghasilkan kesalahan HTTP .....	414
AWS Certificate Manager Sertifikat tidak tersedia untuk digunakan .....	415
Header Multi-Line tidak didukung .....	415
Memecahkan masalah target yang tidak sehat menggunakan peta sumber daya .....	415
Memecahkan masalah pengoptimal target .....	417
Kuota .....	419
Penyeimbang beban .....	419
Kelompok-kelompok target .....	420
Aturan .....	420
Penyimpanan kepercayaan .....	421
Sertifikat .....	421
Header HTTP .....	422
Unit Kapasitas Load Balancer .....	422
Riwayat dokumen .....	423
.....	cdxxxi

# Apa itu Application Load Balancer?

Elastic Load Balancing secara otomatis mendistribusikan lalu lintas masuk Anda ke beberapa target, seperti instans EC2, kontainer, dan alamat IP, dalam satu atau beberapa Availability Zone. Ini memantau kesehatan target terdaftarnya, dan mengarahkan lalu lintas hanya ke target yang sehat. Elastic Load Balancing menskalakan load balancer Anda saat lalu lintas masuk Anda berubah seiring waktu. Ini dapat secara otomatis menskalakan sebagian besar beban kerja.

Elastic Load Balancing mendukung penyeimbang beban berikut: Application Load Balancer, Penyeimbang Beban Jaringan, Gateway Load Balancer, dan Classic Load Balancer. Anda dapat memilih jenis penyeimbang beban yang paling sesuai dengan kebutuhan Anda. Panduan ini membahas Application Load Balancer. Untuk informasi selengkapnya tentang penyeimbang beban lainnya, lihat [Panduan Pengguna untuk Penyeimbang Beban Jaringan](#), [Panduan Pengguna untuk Gateway Load Balancer](#), dan [Panduan Pengguna untuk Classic Load Balancer](#).

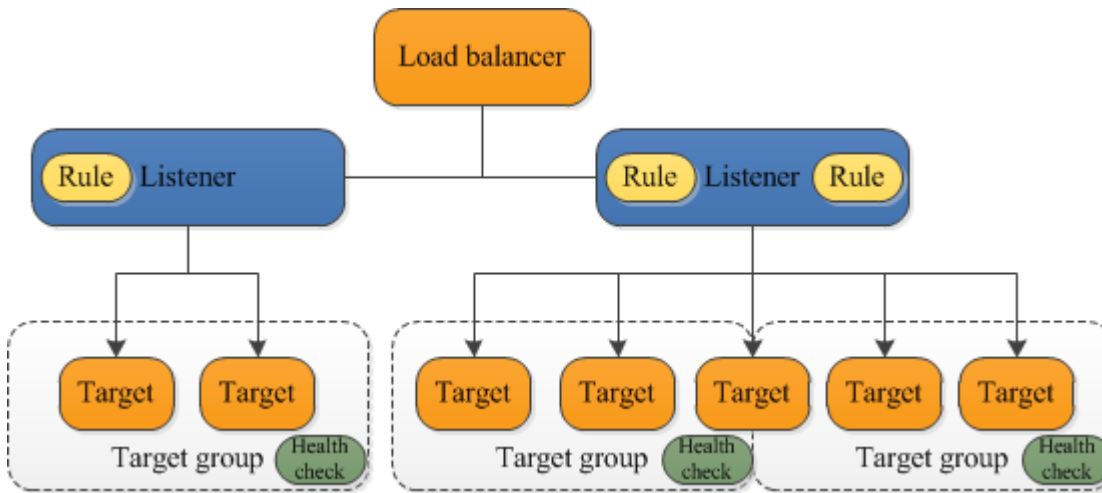
## Komponen Application Load Balancer

Penyeimbang beban berfungsi sebagai titik kontak tunggal untuk klien. Penyeimbang beban mendistribusikan lalu lintas aplikasi yang masuk ke beberapa target, seperti instans EC2, di beberapa Availability Zone. Hal ini akan meningkatkan ketersediaan aplikasi Anda. Anda menambahkan satu listener atau lebih ke penyeimbang beban Anda.

Listener memeriksa permintaan koneksi dari klien, menggunakan protokol dan port yang Anda konfigurasi. Peraturan yang Anda tetapkan untuk listener menentukan cara penyeimbang beban merutekan permintaan untuk target terdaftar. Setiap peraturan terdiri dari prioritas, satu tindakan atau lebih, dan satu syarat atau lebih. Bila syarat untuk suatu peraturan terpenuhi, maka tindakannya dilakukan. Anda harus menentukan aturan default untuk setiap listener, dan Anda dapat menentukan aturan tambahan secara opsional.

Setiap grup target merutekan permintaan ke satu atau beberapa target yang terdaftar, seperti instans EC2, menggunakan protokol dan nomor port yang Anda tentukan. Anda dapat mendaftarkan target dengan beberapa grup target. Anda dapat mengonfigurasi pemeriksaan kondisi berdasarkan per grup target. Pemeriksaan kondisi dilakukan pada semua target yang terdaftar ke grup target yang ditentukan dalam aturan listener untuk penyeimbang beban Anda.

Diagram berikut menggambarkan komponen dasar. Perhatikan bahwa setiap listener berisi aturan default, dan satu listener berisi aturan lain yang merutekan permintaan ke grup target yang berbeda. Satu target terdaftar dengan dua kelompok sasaran.



Untuk informasi lebih lanjut, lihat dokumentasi berikut ini:

- [Penyeimbang beban](#)
- [Pendengar](#)
- [Kelompok sasaran](#)

## Gambaran umum Application Load Balancer

Application Load Balancer berfungsi pada lapisan aplikasi, lapisan ketujuh dari model Open Systems Interconnection (OSI). Setelah penyeimbang beban menerima permintaan, penyeimbang beban mengevaluasi aturan listener dalam urutan prioritas untuk menentukan aturan yang akan diterapkan, dan kemudian memilih target dari grup target untuk tindakan aturan. Anda dapat mengonfigurasi aturan listener untuk merutekan permintaan ke grup target yang berbeda berdasarkan isi lalu lintas aplikasi. Perutean dilakukan secara independen untuk setiap grup target, bahkan ketika target terdaftar dengan beberapa grup target. Anda dapat mengonfigurasi algoritme perutean yang digunakan pada tingkat grup target. Algoritma perutean default adalah round robin; sebagai alternatif, Anda dapat menentukan algoritme perutean permintaan yang paling sedikit menonjol.

Anda dapat menambah dan menghapus target dari penyeimbang beban saat kebutuhan Anda berubah, tanpa mengganggu keseluruhan aliran permintaan ke aplikasi Anda. Elastic Load Balancing menskalakan penyeimbang beban Anda saat lalu lintas ke aplikasi Anda berubah seiring waktu. Elastic Load Balancing dapat menskalakan sebagian besar beban kerja secara otomatis.

Anda dapat mengonfigurasi pemeriksaan kondisi, yang digunakan untuk memantau kondisi target terdaftar sehingga penyeimbang beban hanya dapat mengirim permintaan ke target yang sehat.

Untuk informasi lebih lanjut, lihat [Cara kerja Elastic Load Balancing](#) di Panduan Pengguna Elastic Load Balancing.

## Manfaat migrasi dari Classic Load Balancer

Menggunakan Application Load Balancer alih-alih Classic Load Balancer memiliki manfaat sebagai berikut:

- Dukungan untuk [Syarat jalur](#). Anda dapat mengonfigurasi aturan untuk listener Anda yang meneruskan permintaan berdasarkan URL dalam permintaan tersebut. Ini memungkinkan Anda untuk menyusun aplikasi Anda sebagai layanan yang lebih kecil, dan mengarahkan permintaan ke layanan yang benar berdasarkan konten URL.
- Dukungan untuk [Syarat host](#). Anda dapat mengonfigurasi aturan untuk listener Anda yang meneruskan permintaan berdasarkan bidang host di header HTTP. Ini memungkinkan Anda merutekan permintaan ke beberapa domain menggunakan penyeimbang beban tunggal.
- Dukungan untuk perutean berdasarkan bidang dalam permintaan, seperti [Syarat header HTTP](#) dan metode, parameter kueri, dan alamat IP sumber.
- Dukungan untuk merutekan permintaan ke beberapa aplikasi pada satu instans EC2. Anda dapat mendaftarkan instans atau alamat IP dengan beberapa grup target, masing-masing pada port yang berbeda.
- Dukungan untuk mengarahkan permintaan dari satu URL ke URL lainnya.
- Dukungan untuk mengembalikan respons HTTP kustom.
- Dukungan untuk mendaftarkan target berdasarkan alamat IP, termasuk target di luar VPC untuk penyeimbang beban.
- Dukungan untuk mendaftarkan fungsi Lambda sebagai target.
- Dukungan untuk penyeimbang beban untuk mengotentikasi pengguna aplikasi Anda melalui identitas perusahaan atau sosial mereka sebelum merutekan permintaan.
- Dukungan untuk aplikasi kontainer. Amazon Elastic Container Service (Amazon ECS) dapat memilih port yang tidak terpakai ketika penjadwalan tugas dan mendaftarkan tugas dengan grup target menggunakan port ini. Hal ini memungkinkan Anda untuk memanfaatkan klaster Anda secara efisien.
- Support untuk memantau kesehatan setiap layanan secara independen, karena pemeriksaan kesehatan didefinisikan pada tingkat kelompok sasaran dan banyak CloudWatch metrik dilaporkan pada tingkat kelompok sasaran. Melampirkan grup target ke grup Auto Scaling memungkinkan Anda menskalakan setiap layanan secara dinamis berdasarkan permintaan.

- Log akses berisi informasi tambahan dan disimpan dalam format terkompresi.
- Peningkatan performa penyeimbang beban.

Untuk informasi selengkapnya tentang fitur yang didukung oleh masing-masing tipe load balancer, lihat fitur [Elastic Load Balancing](#).

## Layanan terkait

Elastic Load Balancing bekerja dengan layanan berikut untuk meningkatkan ketersediaan dan skalabilitas aplikasi Anda.

- Amazon EC2— Server virtual yang menjalankan aplikasi Anda di cloud. Anda dapat mengonfigurasi penyeimbang beban Anda untuk mengarahkan lalu lintas ke instans EC2 Anda.
- Amazon EC2 Auto Scaling — Memastikan bahwa Anda menjalankan jumlah instans yang Anda inginkan, bahkan jika sebuah instans gagal, dan memungkinkan Anda untuk secara otomatis menambah atau mengurangi jumlah instans saat permintaan pada instans Anda berubah. Jika Anda mengaktifkan Auto Scaling dengan Elastic Load Balancing, instans yang diluncurkan oleh Auto Scaling secara otomatis terdaftar dengan grup target, dan instance yang diakhiri oleh Auto Scaling secara otomatis dibatalkan registrasi dari grup target.
- AWS Certificate Manager— Ketika Anda membuat pendengar HTTPS, Anda dapat menentukan sertifikat yang disediakan oleh ACM. Penyeimbang beban menggunakan sertifikat untuk mengakhiri koneksi dan mendekripsi permintaan dari klien. Untuk informasi selengkapnya, lihat [Sertifikat SSL untuk Application Load Balancer](#).
- Amazon CloudWatch - Memungkinkan Anda memantau penyeimbang beban dan mengambil tindakan sesuai kebutuhan. Untuk informasi selengkapnya, lihat [CloudWatch metrik untuk Application Load Balancer](#).
- Amazon ECS — Memungkinkan Anda untuk menjalankan, menghentikan, dan mengelola kontainer Docker pada kluster instans EC2. Anda dapat mengonfigurasi penyeimbang beban Anda untuk mengarahkan lalu lintas ke kontainer Anda. Untuk informasi lebih lanjut, lihat [Penyeimbang beban layanan](#) di Panduan Developer Layanan Amazon Elastic Container.
- AWS Global Accelerator — Meningkatkan ketersediaan dan performa aplikasi Anda. Gunakan akselerator untuk mendistribusikan lalu lintas di beberapa penyeimbang beban di satu atau beberapa Wilayah. AWS Lihat informasi selengkapnya di [Panduan Developer AWS Global Accelerator](#).

- Route 53 — Menyediakan cara yang andal dan hemat biaya untuk mengarahkan pengunjung ke situs web dengan menerjemahkan nama domain (seperti `www.example.com`) ke alamat IP numerik (seperti `192.0.2.1`) yang digunakan komputer untuk terhubung satu sama lain. AWS menetapkan sumber URLs daya Anda, seperti penyeimbang beban. Namun, Anda mungkin ingin sebuah URL yang mudah diingat pengguna. Misalnya, Anda dapat memetakan nama domain Anda ke sebuah load balancer. Untuk informasi selengkapnya, lihat [Merutekan lalu lintas ke penyeimbang beban ELB](#) di Panduan Pengembang Amazon Route 53.
- AWS WAF— Anda dapat menggunakan AWS WAF Application Load Balancer Anda untuk mengizinkan atau memblokir permintaan berdasarkan aturan dalam daftar kontrol akses web (web ACL). Untuk informasi selengkapnya, lihat [AWS WAF](#).

Untuk melihat informasi tentang layanan yang terintegrasi dengan penyeimbang beban Anda, pilih penyeimbang beban Anda di Konsol Manajemen AWS dan pilih tab Layanan terintegrasi.

## Harga

Dengan penyeimbang beban, Anda hanya membayar apa yang Anda gunakan. Untuk informasi lebih lanjut, lihat [Harga Elastic Load Balancing?](#)

# Application Load Balancer

Penyeimbang beban berfungsi sebagai titik kontak tunggal untuk klien. Klien mengirimkan permintaan ke penyeimbang beban dan penyeimbang beban mengirimkannya ke target, seperti instans EC2. Untuk mengonfigurasi penyeimbang beban, Anda membuat [grup target](#), lalu mendaftarkan target dengan grup target Anda. Anda juga membuat [listener](#) untuk memeriksa permintaan koneksi dari klien dan aturan listener untuk merutekan permintaan dari klien ke target di satu atau beberapa grup target.

Untuk informasi selengkapnya, lihat [Cara kerja Elastic Load Balancing](#) di Panduan Pengguna Elastic Load Balancing.

## Daftar Isi

- [Subnet untuk penyeimbang beban Anda](#)
- [Grup keamanan penyeimbang beban](#)
- [Status penyeimbang beban](#)
- [Atribut penyeimbang beban](#)
- [Jenis alamat IP](#)
- [Manajemen Alamat IP Application Load Balancer](#)
- [Kumpulan alamat IP IPAM](#)
- [Koneksi penyeimbang beban](#)
- [Penyeimbangan beban lintas zona](#)
- [Nama DNS](#)
- [Membuat Application Load Balancer](#)
- [Perbarui Availability Zone untuk Application Load Balancer](#)
- [Grup keamanan untuk Application Load Balancer Anda](#)
- [Perbarui jenis alamat IP untuk Application Load Balancer](#)
- [Perbarui kumpulan alamat IP IPAM untuk Application Load Balancer Anda](#)
- [Mengedit atribut untuk Application Load Balancer](#)
- [Menandai Application Load Balancer](#)
- [Menghapus Application Load Balancer](#)
- [Lihat peta sumber daya Application Load Balancer](#)

- [Pergeseran zona untuk Application Load Balancer Anda](#)
- [Pemesanan kapasitas untuk Application Load Balancer Anda](#)
- [Integrasi untuk Application Load Balancer](#)

## Subnet untuk penyeimbang beban Anda

Saat Anda membuat Application Load Balancer, Anda harus mengaktifkan zona yang berisi target Anda. Untuk mengaktifkan zona, tentukan subnet di zona tersebut. Elastic Load Balancing menciptakan simpul penyeimbang beban di setiap zona yang Anda tentukan.

### Pertimbangan-pertimbangan

- Penyeimbang beban Anda paling efektif ketika Anda memastikan bahwa setiap zona yang diaktifkan memiliki setidaknya satu target terdaftar.
- Jika Anda mendaftarkan target di zona tetapi tidak mengaktifkan zona tersebut, target terdaftar ini tidak menerima lalu lintas dari penyeimbang beban.
- Jika Anda mengaktifkan beberapa zona untuk penyeimbang beban Anda, zona harus dari jenis yang sama. Misalnya, Anda tidak dapat mengaktifkan Availability Zone dan Local Zone.
- Anda dapat menentukan subnet yang dibagikan dengan Anda.
- Elastic Load Balancing menciptakan antarmuka jaringan di subnet tempat Anda mengonfigurasi penyeimbang beban. Antarmuka jaringan ini dicadangkan sehingga penyeimbang beban dapat menyelesaikan tindakan pemeliharaan bahkan ketika subnet hampir habis pada alamat IP yang tersedia. Mereka memiliki deskripsi “ENI dicadangkan oleh ELB untuk subnet”.

Aplikasi Load Balancers mendukung jenis subnet berikut.

### Jenis subnet

- [Subnet Zona Ketersediaan](#)
- [Subnet Zona Lokal](#)
- [Subnet pos terdepan](#)

## Subnet Zona Ketersediaan

Anda harus memilih setidaknya dua subnet Availability Zone. Pembatasan berikut berlaku:

- Setiap subnet harus berasal dari Availability Zone yang berbeda.
- Untuk memastikan penyeimbang beban Anda dapat menskalakan dengan benar, verifikasi bahwa setiap subnet Availability Zone untuk penyeimbang beban Anda memiliki blok CIDR dengan setidaknya /27 bitmask (misalnya, 10.0.0.0/27) dan setidaknya delapan alamat IP gratis per subnet. Kedelapan alamat IP ini diperlukan untuk memungkinkan penyeimbang beban skala jika diperlukan. Penyeimbang beban Anda menggunakan alamat IP ini untuk membuat koneksi dengan target. Tanpa mereka Application Load Balancer Anda dapat mengalami kesulitan dengan upaya penggantian node, menyebabkannya memasuki status gagal.

Catatan: Jika subnet Application Load Balancers kehabisan alamat IP yang dapat digunakan saat mencoba menskalakan, Application Load Balancer akan berjalan dengan kapasitas yang tidak mencukupi. Selama waktu ini, node lama terus melayani lalu lintas, tetapi upaya penskalaan yang macet dapat menyebabkan kesalahan 5xx atau batas waktu ketika mencoba membuat koneksi.

## Subnet Zona Lokal

Anda dapat menentukan subnet Zona Lokal. Fitur berikut tidak didukung dengan subnet zona lokal:

- Lambda berfungsi sebagai target
- Autentikasi TLS bersama
- AWS WAF integrasi

## Subnet pos terdepan

Anda dapat menentukan subnet Outpost tunggal. Pembatasan berikut berlaku:

- Anda harus menginstal dan mengonfigurasi Outpost di pusat data On-Premise Anda. Anda harus memiliki koneksi jaringan yang dapat diandalkan antara Outpost Anda dan Wilayah AWS . Untuk informasi selengkapnya, silakan lihat [Panduan Pengguna AWS Outposts](#).
- Penyeimbang beban membutuhkan dua `large` instance di Outpost untuk node penyeimbang beban. Jenis instance yang didukung ditampilkan dalam tabel berikut. Timbangan penyeimbang beban sesuai kebutuhan, mengubah ukuran node satu ukuran pada satu waktu (dari `large` ke `xlarge`, lalu `xlarge` ke `2xlarge`, dan kemudian `2xlarge` ke `4xlarge`). Setelah menskalakan node ke ukuran instans terbesar, jika Anda membutuhkan kapasitas tambahan, penyeimbang beban menambahkan `4xlarge` instance sebagai node penyeimbang beban. Jika Anda tidak memiliki kapasitas instans yang memadai atau alamat IP yang tersedia untuk menskalakan

penyeimbang beban, penyeimbang beban melaporkan peristiwa ke [Dasbor AWS Health](#) dan status penyeimbang beban adalah `active_impaired`.

- Anda dapat mendaftarkan target dengan ID instans atau alamat IP. Jika Anda mendaftarkan target di AWS Wilayah untuk Pos Luar, mereka tidak digunakan.
- Fitur berikut tidak didukung:
  - AWS Global Accelerator integrasi
  - Lambda berfungsi sebagai target
  - Autentikasi TLS bersama
  - Sesi lengket
  - Otentikasi pengguna
  - AWS WAF integrasi

Application Load Balancer dapat di-deploy pada instans `c5/c5d`, `m5/m5d`, atau `r5/r5d` pada Outpost. Tabel berikut menunjukkan ukuran dan volume EBS per tipe instans yang dapat digunakan oleh penyeimbang beban pada Outpost:

Tipe dan ukuran instans	Volume EBS (GB)
<b>c5/c5d</b>	
large	50
xlarge	50
2xlarge	50
4xlarge	100
<b>m5/m5d</b>	
large	50
xlarge	50
2xlarge	100
4xlarge	100

Tipe dan ukuran instans	Volume EBS (GB)	
r5/r5d		
large	50	
xlarge	100	
2xlarge	100	
4xlarge	100	

## Grup keamanan penyeimbang beban

Grup keamanan bertindak sebagai firewall yang mengontrol lalu lintas yang diizinkan ke dan dari penyeimbang beban Anda. Anda dapat memilih port dan protokol untuk mengizinkan lalu lintas masuk dan keluar.

Aturan untuk grup keamanan yang terkait dengan penyeimbang beban Anda harus mengizinkan lalu lintas di kedua arah pada listener dan port pemeriksaan kondisi. Setiap kali menambahkan listener ke penyeimbang beban atau memperbarui port pemeriksaan kondisi untuk grup target, Anda harus meninjau aturan grup keamanan untuk memastikan bahwa mereka mengizinkan lalu lintas pada port baru di kedua arah. Untuk informasi selengkapnya, lihat [Aturan yang disarankan](#).

## Status penyeimbang beban

Penyeimbang beban dapat berada dalam salah satu status berikut:

### provisioning

Penyeimbang beban sedang disiapkan.

### active

Penyeimbang beban telah sepenuhnya disiapkan dan siap untuk merutekan lalu lintas.

### active\_impaired

Penyeimbang beban merutekan lalu lintas, tetapi tidak memiliki sumber daya yang dibutuhkan untuk menskalakan.

failed

Penyeimbang beban tidak dapat disiapkan.

## Atribut penyeimbang beban

Anda dapat mengonfigurasi Application Load Balancer Anda dengan mengedit atributnya. Untuk informasi selengkapnya, lihat [Edit atribut penyeimbang beban](#).

Berikut adalah atribut penyeimbang beban:

`access_logs.s3.enabled`

Menunjukkan apakah log akses yang disimpan di Amazon S3 diaktifkan. Nilai default-nya `false`.

`access_logs.s3.bucket`

Nama bucket Amazon S3 untuk log akses. Atribut ini diperlukan jika log akses diaktifkan. Untuk informasi selengkapnya, lihat [Aktifkan log akses](#).

`access_logs.s3.prefix`

Prefiks untuk lokasi di bucket Amazon S3.

`client_keep_alive.seconds`

Nilai klien keepalive, dalam hitungan detik. Defaultnya adalah 3600 detik.

`deletion_protection.enabled`

Menunjukkan apakah perlindungan penghapusan diaktifkan. Nilai default-nya `false`.

`idle_timeout.timeout_seconds`

Nilai batas waktu idle dalam detik. Nilai default-nya adalah 60 detik.

`ipv6.deny_all_igw_traffic`

Memblokir akses internet gateway (IGW) ke penyeimbang beban, mencegah akses yang tidak diinginkan ke penyeimbang beban internal Anda melalui gateway internet. Ini diatur `false` untuk penyeimbang beban yang menghadap ke internet dan `true` untuk penyeimbang beban internal. Atribut ini tidak mencegah akses internet non-IGW (seperti, melalui peering, Transit Gateway AWS Direct Connect, atau). Site-to-Site VPN

`routing.http.desync_mitigation_mode`

Menentukan bagaimana penyeimbang beban menangani permintaan yang mungkin menimbulkan risiko keamanan pada aplikasi Anda. Nilai yang mungkin adalah `monitor`, `defensive`, dan `strictest`. Nilai default-nya `defensive`.

`routing.http.drop_invalid_header_fields.enabled`

Menunjukkan apakah header HTTP dengan kolom header yang tidak valid dihapus oleh penyeimbang beban (`true`) atau dirutekan ke target (`false`). Nilai default-nya `false`. Elastic Load Balancing mengharuskan nama header HTTP yang valid sesuai dengan ekspresi reguler `[A-Za-z0-9]+`, seperti yang dijelaskan dalam Registri Nama Bidang HTTP. Setiap nama terdiri dari karakter alfanumerik atau tanda hubung. Pilih `true` jika Anda ingin header HTTP yang tidak sesuai dengan pola ini, dihapus dari permintaan.

`routing.http.preserve_host_header.enabled`

Menunjukkan apakah Application Load Balancer harus mempertahankan Host header dalam permintaan HTTP dan mengirimkannya ke target tanpa perubahan apa pun. Nilai yang mungkin adalah `true` dan `false`. Nilai default-nya `false`.

`routing.http.x_amzn_tls_version_and_cipher_suite.enabled`

Menunjukkan apakah dua header (`x-amzn-tls-version` dan `x-amzn-tls-cipher-suite`), yang berisi informasi tentang versi TLS yang dinegosiasikan dan cipher suite, ditambahkan ke permintaan klien sebelum mengirimnya ke target. `x-amzn-tls-version` header memiliki informasi tentang versi protokol TLS yang dinegosiasikan dengan klien, dan `x-amzn-tls-cipher-suite` header memiliki informasi tentang cipher suite yang dinegosiasikan dengan klien. Kedua header dalam format OpenSSL. Nilai yang mungkin untuk atribut adalah `true` dan `false`. Nilai default-nya `false`.

`routing.http.xff_client_port.enabled`

Menunjukkan apakah X-Forwarded-For header harus mempertahankan port sumber yang digunakan klien untuk terhubung ke penyeimbang beban. Nilai yang mungkin adalah `true` dan `false`. Nilai default-nya `false`.

`routing.http.xff_header_processing.mode`

Memungkinkan Anda untuk memodifikasi, mempertahankan, atau menghapus X-Forwarded-For header dalam permintaan HTTP sebelum Application Load Balancer mengirimkan permintaan ke target. Nilai yang mungkin adalah `append`, `preserve`, dan `remove`. Nilai default-nya `append`.

- Jika nilainya `append`, Application Load Balancer menambahkan alamat IP klien (dari hop terakhir) ke `X-Forwarded-For` header dalam permintaan HTTP sebelum mengirimkannya ke target.
- Jika nilainya `preserve`, Application Load Balancer mempertahankan `X-Forwarded-For` header dalam permintaan HTTP, dan mengirimkannya ke target tanpa perubahan apa pun.
- Jika nilainya `remove`, Application Load Balancer menghapus `X-Forwarded-For` header dalam permintaan HTTP sebelum mengirimkannya ke target.

`routing.http2.enabled`

Menunjukkan apakah klien dapat terhubung ke penyeimbang beban menggunakan HTTP/2. Jika `true`, klien dapat terhubung menggunakan HTTP/2 atau HTTP/1.1. Jika `false`, klien harus terhubung menggunakan HTTP/1.1. Nilai default-nya `true`.

`waf.fail_open.enabled`

Menunjukkan apakah akan mengizinkan penyeimbang beban yang AWS WAF diaktifkan untuk merutekan permintaan ke target jika tidak dapat meneruskan permintaan ke AWS WAF. Nilai yang mungkin adalah `true` dan `false`. Nilai default-nya `false`.

#### Note

`routing.http.drop_invalid_header_fields.enabled` Atribut diperkenalkan untuk menawarkan perlindungan desync HTTP.

`routing.http.desync_mitigation_mode` Atribut ditambahkan untuk memberikan perlindungan yang lebih komprehensif dari desync HTTP untuk aplikasi Anda. Anda tidak diharuskan untuk menggunakan kedua atribut dan dapat memilih atribut mana yang paling sesuai dengan persyaratan aplikasi Anda.

## Jenis alamat IP

Anda dapat mengatur jenis alamat IP yang dapat digunakan klien untuk mengakses penyeimbang beban internal dan internet-facing Anda.

Application Load Balancers mendukung jenis alamat IP berikut:

### **ipv4**

Klien harus terhubung ke penyeimbang beban menggunakan IPv4 alamat (misalnya, 192.0.2.1).

## dualstack

Klien dapat terhubung ke penyeimbang beban menggunakan kedua IPv4 alamat (misalnya, 192.0.2.1) dan IPv6 alamat (misalnya, 2001:0 db 8:85 a 3:0:0:8 a2e: 0370:7334).

## dualstack-without-public-ipv4

Klien harus terhubung ke penyeimbang beban menggunakan IPv6 alamat (misalnya, 2001:0 db 8:85 a 3:0:0:8 a2e: 0370:7334).

### Pertimbangan-pertimbangan

- Load balancer berkomunikasi dengan target berdasarkan jenis alamat IP dari kelompok target.
- Saat Anda mengaktifkan mode dualstack untuk penyeimbang beban, Elastic Load Balancing menyediakan catatan DNS AAAA untuk penyeimbang beban. Klien yang berkomunikasi dengan penyeimbang beban menggunakan IPv4 alamat menyelesaikan catatan DNS A. Klien yang berkomunikasi dengan penyeimbang beban menggunakan IPv6 alamat menyelesaikan catatan DNS AAAA.
- Akses ke penyeimbang beban dualstack internal Anda melalui gateway internet diblokir untuk mencegah akses internet yang tidak diinginkan. Namun, ini tidak mencegah akses internet non-IGW (seperti, melalui peering, Transit Gateway AWS Direct Connect, atau). Site-to-Site VPN
- Autentikasi Application Load Balancer hanya mendukung IPv4 saat menghubungkan ke Endpoint Penyedia Identitas (IDP) atau Amazon Cognito. Tanpa IPv4 alamat publik, penyeimbang beban tidak dapat menyelesaikan proses otentikasi, menghasilkan kesalahan HTTP 500.

Untuk informasi selengkapnya, lihat [Perbarui jenis alamat IP untuk Application Load Balancer](#).

## Manajemen Alamat IP Application Load Balancer

Application Load Balancer menggunakan alamat Public Elastic dari [kumpulan IPv4 alamat publik IPv4 EC2](#). [Alamat IP ini terlihat di AWS akun Anda saat menggunakan CLI, API, atau melihat bagian IPs Elastic \(EIP\) di Konsol](#). AWS Setiap alamat IP terkait ALB ditandai dengan atribut `service_managed` yang disetel ke "ALB".

Meskipun ini IPs terlihat di akun Anda, mereka tetap dikelola sepenuhnya oleh layanan Application Load Balancer dan tidak dapat dimodifikasi atau dirilis. Application Load Balancer dirilis IPs kembali ke kumpulan IPv4 alamat publik saat tidak lagi digunakan.

CloudTrail log panggilan API yang terkait dengan EIP Application Load Balancer, seperti `AllocateAddress`. Panggilan API ini dipanggil oleh Principal Layanan `'elasticloadbalancing.amazonaws.com'`.

#### Note

Catatan: IPs dialokasikan oleh Application Load Balancer tidak dihitung terhadap batas EIP akun Anda.

## Kumpulan alamat IP IPAM

Kumpulan alamat IP IPAM adalah kumpulan rentang alamat IP yang berdekatan (atau CIDRs) yang Anda buat menggunakan Amazon VPC IP Address Manager (IPAM). Menggunakan kumpulan alamat IP IPAM dengan Application Load Balancer memungkinkan Anda mengatur alamat sesuai dengan IPv4 kebutuhan perutean dan keamanan Anda. Kumpulan alamat IP IPAM memberi Anda pilihan untuk membawa beberapa atau semua rentang IPv4 alamat publik Anda AWS dan menggunakannya dengan Application Load Balancers Anda. Kumpulan alamat IP IPAM Anda selalu diprioritaskan saat meluncurkan instans EC2 dan membuat Application Load Balancers. Ketika alamat IP Anda tidak lagi digunakan, mereka segera tersedia untuk digunakan lagi.

Untuk memulai, buat kumpulan alamat IP IPAM. Untuk informasi selengkapnya, lihat [Membawa alamat IP Anda ke IPAM](#).

### Pertimbangan-pertimbangan

- Kumpulan IPv6 alamat IPAM tidak didukung.
- Kumpulan IPv4 alamat IPAM tidak didukung dengan penyeimbang beban internal atau jenis `dualstack-without-public-ipv4` alamat IP.
- Anda tidak dapat menghapus alamat IP di kumpulan alamat IP IPAM jika saat ini digunakan oleh penyeimbang beban.
- Selama transisi ke kumpulan alamat IP IPAM yang berbeda, koneksi yang ada dihentikan sesuai dengan durasi keepalive klien HTTP penyeimbang beban.
- Kumpulan alamat IP IPAM dapat dibagikan di beberapa akun. Untuk informasi selengkapnya, lihat [Mengkonfigurasi opsi integrasi untuk IPAM Anda](#).

- Tidak ada biaya tambahan yang terkait dengan penggunaan kumpulan alamat IP IPAM dengan penyeimbang beban Anda. Namun, mungkin ada biaya yang terkait dengan IPAM, tergantung pada tingkat mana yang Anda gunakan.

Jika tidak ada lagi alamat IP yang dapat ditetapkan di kumpulan alamat IP IPAM Anda, Elastic Load Balancing AWS menggunakan IPv4 alamat terkelola sebagai gantinya. Ada biaya tambahan untuk menggunakan IPv4 alamat AWS terkelola. Untuk menghindari biaya ini, Anda dapat menambahkan rentang alamat IP ke kumpulan alamat IP IPAM yang ada.

Untuk informasi selengkapnya, lihat [harga Amazon VPC](#).

## Koneksi penyeimbang beban

Saat memproses permintaan, penyeimbang beban mempertahankan dua koneksi: satu koneksi dengan klien dan satu koneksi dengan target. Koneksi antara penyeimbang beban dan klien juga disebut sebagai koneksi front-end. Koneksi antara penyeimbang beban dan target juga disebut sebagai koneksi back-end.

## Penyeimbangan beban lintas zona

Dengan Application Load Balancers, penyeimbangan beban lintas zona aktif secara default dan tidak dapat diubah pada tingkat penyeimbang beban. Untuk informasi selengkapnya, lihat bagian [penyeimbangan beban lintas zona](#) di Panduan Pengguna Elastic Load Balancing.

Mematikan penyeimbangan beban lintas zona dimungkinkan di tingkat kelompok sasaran. Untuk informasi selengkapnya, lihat [the section called “Matikan penyeimbangan beban lintas zona”](#).

## Nama DNS

Setiap Application Load Balancer menerima nama Domain Name System (DNS) default dengan sintaks berikut: - .elb. *name id region*.amazonaws.com. Misalnya, my-load-balancer-1234567890abcdef. elb.us-east-2.amazonaws.com.

Jika Anda lebih suka menggunakan nama DNS yang lebih mudah diingat, Anda dapat membuat nama domain khusus dan mengaitkannya dengan nama DNS untuk Application Load Balancer Anda. Ketika klien membuat permintaan menggunakan nama domain kustom ini, server DNS menyelesaikannya ke nama DNS untuk Application Load Balancer Anda.

Pertama, daftarkan nama domain dengan registrar nama domain terakreditasi. Selanjutnya, gunakan layanan DNS Anda, seperti registrar domain Anda, untuk membuat catatan DNS untuk merutekan permintaan ke Application Load Balancer Anda. Untuk informasi lebih lanjut, lihat dokumentasi untuk server DNS Anda. Misalnya, jika Anda menggunakan Amazon Route 53 sebagai layanan DNS, Anda membuat catatan alias yang menunjuk ke Application Load Balancer Anda. Untuk informasi selengkapnya, lihat [Merutekan lalu lintas ke penyeimbang beban ELB](#) di Panduan Pengembang Amazon Route 53.

Application Load Balancer memiliki satu alamat IP per Availability Zone yang diaktifkan. Ini adalah alamat IP dari node Application Load Balancer. Nama DNS dari Application Load Balancer diselesaikan ke alamat-alamat ini. Misalnya, misalkan nama domain kustom untuk Application Load Balancer Anda adalah `example.applicationloadbalancer.com`. Gunakan `nslookup` perintah berikut di `g` atau untuk menentukan alamat IP dari node Application Load Balancer.

Linux atau Mac

```
$ dig +short example.applicationloadbalancer.com
```

Windows

```
C:\> nslookup example.applicationloadbalancer.com
```

Application Load Balancer memiliki catatan DNS untuk node-nya. Anda dapat menggunakan nama DNS dengan sintaks berikut untuk menentukan alamat IP dari node Application Load Balancer: `az name-id.elb.region.amazonaws.com`.

Linux atau Mac

```
$ dig +short us-east-2b.my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com
```

Windows

```
C:\> nslookup us-east-2b.my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com
```

## Membuat Application Load Balancer

Application Load Balancer mengambil permintaan dari klien dan mendistribusikannya ke seluruh target dalam grup target, seperti instans EC2. Untuk informasi selengkapnya, lihat [Cara Kerja Elastic Load Balancing](#) dalam Panduan Pengguna Elastic Load Balancing.

## Tugas

- [Prasyarat](#)
- [Buat penyeimbang beban](#)
- [Uji penyeimbang beban](#)
- [Langkah selanjutnya](#)

## Prasyarat

- Tentukan Availability Zones dan jenis alamat IP mana yang akan didukung aplikasi Anda. Konfigurasi VPC penyeimbang beban dengan subnet di masing-masing Availability Zone ini. Jika aplikasi akan mendukung keduanya IPv4 dan IPv6 lalu lintas, pastikan bahwa subnet memiliki keduanya IPv4 dan IPv6 CIDRs. Terapkan setidaknya satu target di setiap Availability Zone. Untuk informasi selengkapnya, lihat [the section called “Subnet untuk penyeimbang beban Anda”](#).
- Pastikan bahwa grup keamanan untuk instance target mengizinkan lalu lintas pada port listener dari alamat IP klien (jika target ditentukan oleh ID instance) atau node penyeimbang beban (jika target ditentukan oleh alamat IP). Untuk informasi selengkapnya, lihat [Aturan yang disarankan](#).
- Pastikan bahwa kelompok keamanan untuk contoh target memungkinkan lalu lintas dari penyeimbang beban pada port pemeriksaan kesehatan menggunakan protokol pemeriksaan kesehatan.

## Buat penyeimbang beban

Sebagai bagian dari pembuatan Application Load Balancer, Anda akan membuat penyeimbang beban, setidaknya satu pendengar, dan setidaknya satu grup target. Penyeimbang beban Anda siap menangani permintaan klien ketika setidaknya ada satu target terdaftar yang sehat di setiap Availability Zone yang diaktifkan.

### Console

Untuk membuat Application Load Balancer

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Di panel navigasi, pilih Load Balancers.
3. Pilih Buat Penyeimbang Beban.
4. Di bawah Application Load Balancer, pilih Buat.

## 5. Konfigurasi dasar

- a. Untuk Name, masukkan nama untuk penyeimbang beban Anda. Nama harus unik dalam set penyeimbang beban Anda untuk Wilayah. Nama dapat memiliki maksimal 32 karakter, dan hanya dapat berisi karakter alfanumerik dan tanda hubung. Mereka tidak dapat memulai atau mengakhiri dengan tanda hubung, atau dengan `internal`- Anda tidak dapat mengubah nama Application Load Balancer setelah dibuat.
- b. Untuk Skema, pilih Mengakses Internet atau Internal. Penyeimbang beban yang menghadap ke internet merutekan permintaan dari klien ke target melalui internet. Penyeimbang beban internal merutekan permintaan ke target menggunakan alamat IP privat.
- c. Untuk jenis alamat IP Load balancer, pilih IPv4 apakah klien Anda menggunakan IPv4 alamat untuk berkomunikasi dengan penyeimbang beban atau Dualstack jika klien Anda menggunakan keduanya IPv4 dan IPv6 alamat untuk berkomunikasi dengan penyeimbang beban. Pilih Dualstack tanpa publik IPv4 jika klien Anda hanya menggunakan IPv6 alamat untuk berkomunikasi dengan penyeimbang beban.

## 6. Pemetaan jaringan

- a. Untuk VPC, pilih VPC yang Anda siapkan untuk penyeimbang beban Anda. Dengan penyeimbang beban yang menghadap ke internet, hanya VPCs dengan gateway internet yang tersedia untuk dipilih.
- b. (Opsional) Untuk kolom IP, Anda dapat memilih Gunakan kolam IPAM untuk IPv4 alamat publik. Untuk informasi selengkapnya, lihat [the section called “Kumpulan alamat IP IPAM”](#).
- c. Untuk Availability Zone dan subnet, aktifkan zona untuk penyeimbang beban Anda sebagai berikut:
  - Pilih subnet dari setidaknya dua Availability Zone
  - Pilih subnet dari setidaknya satu Zona Lokal
  - Pilih satu Subnet Outpost

Untuk informasi selengkapnya, lihat [the section called “Subnet untuk penyeimbang beban Anda”](#).

Dengan penyeimbang beban Dualstack, Anda harus memilih subnet dengan keduanya IPv4 dan blok CIDR. IPv6

## 7. Grup keamanan

Kami memilih grup keamanan default untuk VPC penyeimbang beban. Anda dapat memilih grup keamanan tambahan sesuai kebutuhan. Jika Anda tidak memiliki grup keamanan yang memenuhi kebutuhan Anda, pilih buat grup keamanan baru untuk membuatnya sekarang. Untuk informasi selengkapnya, lihat [Membuat grup keamanan](#) di Panduan Pengguna Amazon VPC.

## 8. Pendengar dan perutean

- a. Defaultnya adalah pendengar yang menerima lalu lintas HTTP pada port 80. Anda dapat menyimpan pengaturan pendengar default, atau memodifikasi Protokol dan Port sesuai kebutuhan.
- b. Untuk Tindakan Bawaan, pilih grup target untuk meneruskan lalu lintas. Jika Anda tidak memiliki grup target yang memenuhi kebutuhan Anda, pilih Buat grup target untuk membuatnya sekarang. Untuk informasi selengkapnya, lihat [Buat grup target](#).
- c. (Opsional) Pilih Tambahkan tag pendengar dan masukkan kunci tag dan nilai tag.
- d. (Opsional) Pilih Tambahkan pendengar untuk menambahkan pendengar lain (misalnya, pendengar HTTPS).

## 9. Pengaturan pendengar yang aman

Bagian ini hanya muncul jika Anda menambahkan pendengar HTTPS.

- a. Untuk kebijakan Keamanan, pilih kebijakan keamanan yang memenuhi persyaratan Anda. Untuk informasi selengkapnya, lihat [Kebijakan keamanan](#).
- b. Untuk SSL/TLS sertifikat Default, opsi berikut tersedia:
  - Jika Anda membuat atau mengimpor sertifikat menggunakan AWS Certificate Manager, pilih Dari ACM, lalu pilih sertifikat.
  - Jika Anda mengimpor sertifikat menggunakan IAM, pilih Dari IAM, lalu pilih sertifikat Anda.
  - Jika Anda tidak memiliki sertifikat yang tersedia di ACM tetapi memiliki sertifikat untuk digunakan dengan penyeimbang beban Anda, pilih Impor sertifikat dan berikan informasi yang diperlukan. Jika tidak, pilih Minta sertifikat ACM baru. Untuk informasi selengkapnya, lihat [AWS Certificate Manager sertifikat](#) di Panduan AWS Certificate Manager Pengguna.
- c. (Opsional) Pilih Mutual Authentication (mTLS), pilih kebijakan untuk mengaktifkan ALPN.

Untuk informasi selengkapnya, lihat [Autentikasi TLS bersama](#).

#### 10. Optimalkan dengan integrasi layanan

(Opsional) Anda dapat mengintegrasikan yang lain AWS dengan penyeimbang beban Anda. Untuk informasi selengkapnya, lihat [Integrasi penyeimbang beban](#).

#### 11. Tag penyeimbang beban

(Opsional) Perluas tag penyeimbang beban. Pilih Tambahkan tag baru dan masukkan kunci tag dan nilai tag. Untuk informasi selengkapnya, lihat [Tag](#).

#### 12. Ringkasan

Tinjau konfigurasi Anda, dan pilih Buat penyeimbang beban. Beberapa atribut default diterapkan ke Network Load Balancer Anda selama pembuatan. Anda dapat melihat dan mengeditnya setelah membuat Network Load Balancer. Untuk informasi selengkapnya, lihat [Atribut penyeimbang beban](#).

## AWS CLI

Untuk membuat Application Load Balancer

Gunakan perintah [create-load-balancer](#).

Contoh berikut membuat penyeimbang beban yang menghadap ke internet dengan dua Availability Zone yang diaktifkan dan grup keamanan.

```
aws elbv2 create-load-balancer \  
  --name my-load-balancer \  
  --type application \  
  --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \  
  --security-groups sg-1111222233334444
```

Untuk membuat Application Load Balancer internal

Sertakan `--scheme` opsi seperti yang ditunjukkan pada contoh berikut.

```
aws elbv2 create-load-balancer \  
  --name my-load-balancer \  
  --type application \  
  --scheme internal
```

```
--scheme internal \
--subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
--security-groups sg-1111222233334444
```

Untuk membuat Application Load Balancer dualstack

Sertakan `--ip-address-type` opsi seperti yang ditunjukkan pada contoh berikut.

```
aws elbv2 create-load-balancer \
  --name my-load-balancer \
  --type application \
  --ip-address-type dualstack \
  --subnets subnet-1234567890abcdef0 subnet-0abcdef1234567890 \
  --security-groups sg-1111222233334444
```

Untuk menambahkan pendengar

Gunakan perintah [create-listener](#). Sebagai contoh, lihat [Membuat listener HTTP](#) dan [Buat listener HTTPS](#).

## CloudFormation

Untuk membuat Application Load Balancer

Tentukan sumber daya tipe [AWS::ElasticLoadBalancingV2::LoadBalancer](#).

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      IpAddressType: dualstack
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      Tags:
        - Key: "department"
          Value: "123"
```

Untuk menambahkan pendengar

Tentukan sumber daya tipe [AWS::ElasticLoadBalancingV2::Listener](#). Sebagai contoh, lihat [Membuat listener HTTP](#) dan [Buat listener HTTPS](#).

## Uji penyeimbang beban

Setelah membuat penyeimbang beban, Anda dapat memverifikasi bahwa instans EC2 Anda lulus pemeriksaan kesehatan awal. Anda kemudian dapat memeriksa apakah penyeimbang beban mengirimkan lalu lintas ke instans EC2 Anda. Untuk menghapus penyeimbang beban, lihat [Menghapus Application Load Balancer](#).

Untuk menguji penyeimbang beban

1. Setelah penyeimbang beban dibuat, pilih Tutup.
2. Di panel navigasi, pilih Target Groups.
3. Pilih grup target yang baru dibuat.
4. Pilih Target dan verifikasi bahwa instans Anda sudah siap. Jika status instance adalah `initial`, itu biasanya karena instance masih dalam proses didaftarkan. Status ini juga dapat menunjukkan bahwa instans belum lulus jumlah minimum pemeriksaan kesehatan untuk dianggap sehat. Setelah status setidaknya satu instans sehat, Anda dapat menguji penyeimbang beban Anda. Untuk informasi selengkapnya, lihat [Status kondisi target](#).
5. Di panel navigasi, pilih Load Balancers.
6. Pilih penyeimbang beban yang baru dibuat.
7. Pilih Deskripsi dan salin nama DNS yang menghadap ke internet atau penyeimbang beban internal (misalnya, `my-load-balancer-1234567890abcdef.elb.us-east-2.amazonaws.com`).
  - Untuk penyeimbang beban yang menghadap internet, tempelkan nama DNS ke bidang alamat browser web yang terhubung ke internet.
  - Untuk penyeimbang beban internal, tempelkan nama DNS ke bidang alamat browser web yang memiliki konektivitas pribadi ke VPC.

Jika semuanya dikonfigurasi dengan benar, browser menampilkan halaman default server Anda.

8. Jika halaman web tidak ditampilkan, lihat dokumen berikut untuk bantuan konfigurasi tambahan dan langkah pemecahan masalah.

- Untuk masalah terkait DNS, lihat [Merutekan lalu lintas ke penyeimbang beban ELB di Panduan Pengembang Amazon Route 53](#).
- Untuk masalah terkait Load Balancer, lihat [Memecahkan masalah Application Load Balancer](#)

## Langkah selanjutnya

Setelah membuat penyeimbang beban, Anda mungkin ingin melakukan hal berikut:

- Tambahkan [aturan pendengar](#).
- Konfigurasi [atribut penyeimbang beban](#).
- Konfigurasi [atribut grup target](#).
- [HTTPS listeners] Tambahkan sertifikat ke daftar [sertifikat opsional](#).
- Konfigurasi [fitur pemantauan](#).

## Perbarui Availability Zone untuk Application Load Balancer

Anda dapat mengaktifkan atau menonaktifkan Availability Zone untuk penyeimbang beban kapan saja. Setelah mengaktifkan Availability Zone, penyeimbang beban mulai merutekan permintaan ke target terdaftar di Availability Zone tersebut. Application Load Balancer memiliki penyeimbangan beban lintas zona secara default, sehingga permintaan dirutekan ke semua target terdaftar di semua Availability Zone. Saat penyeimbangan beban lintas zona tidak aktif, penyeimbang beban hanya merutekan permintaan ke target di Availability Zone yang sama. Untuk informasi selengkapnya, lihat [Penyeimbangan beban lintas zona](#). Penyeimbang beban Anda paling efektif jika Anda memastikan bahwa setiap Availability Zone yang diaktifkan memiliki setidaknya satu target terdaftar.

Setelah menonaktifkan Availability Zone, target di Availability Zone tersebut tetap terdaftar dengan penyeimbang beban, tetapi penyeimbang beban tidak akan merutekan permintaan ke mereka.

Untuk informasi selengkapnya, lihat [the section called “Subnet untuk penyeimbang beban Anda”](#).

### Console

Untuk memperbarui Availability Zone

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.

3. Pilih penyeimbang beban.
4. Pada tab Pemetaan jaringan, pilih Edit subnet.
5. Untuk mengaktifkan Availability Zone, pilih kotak centang dan pilih satu subnet. Jika hanya ada satu subnet yang tersedia, itu dipilih untuk Anda.
6. Untuk mengubah subnet untuk Availability Zone yang diaktifkan, pilih salah satu subnet lain dari daftar.
7. Untuk menonaktifkan Availability Zone, kosongkan kotak centang.
8. Pilih Simpan perubahan.

## AWS CLI

Untuk memperbarui Availability Zone

Gunakan perintah [set-subnet](#).

```
aws elbv2 set-subnets \  
  --load-balancer-arn load-balancer-arn \  
  --subnets subnet-8360a9e7EXAMPLE subnet-b7d581c0EXAMPLE
```

## CloudFormation

Untuk memperbarui Availability Zone

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:  
      Name: my-alb  
      Type: application  
      Scheme: internal  
      IpAddressType: dualstack  
      Subnets:  
        - !Ref subnet-AZ1  
        - !Ref new-subnet-AZ2  
      SecurityGroups:  
        - !Ref mySecurityGroup
```

## Grup keamanan untuk Application Load Balancer Anda

Grup keamanan untuk Application Load Balancer Anda mengontrol lalu lintas yang diizinkan untuk mencapai dan meninggalkan penyeimbang beban. Anda harus memastikan bahwa penyeimbang beban dapat berkomunikasi dengan target terdaftar pada port listener dan port pemeriksaan kondisi. Setiap kali menambahkan listener ke penyeimbang beban atau memperbarui port pemeriksaan kondisi untuk grup target yang digunakan oleh penyeimbang beban untuk merutekan permintaan, Anda harus memverifikasi bahwa grup keamanan yang terkait dengan penyeimbang beban mengizinkan lalu lintas pada port baru di kedua arah. Jika tidak, Anda dapat mengedit aturan untuk grup keamanan yang saat ini terkait atau mengaitkan grup keamanan yang berbeda dengan penyeimbang beban. Anda dapat memilih port dan protokol untuk memungkinkan. Misalnya, Anda dapat membuka koneksi Internet Control Message Protocol (ICMP) untuk penyeimbang beban untuk merespons permintaan ping (namun, permintaan ping tidak diteruskan ke instans apa pun).

### Pertimbangan-pertimbangan

- Untuk memastikan target Anda menerima lalu lintas secara eksklusif dari penyeimbang beban, batasi kelompok keamanan yang terkait dengan target Anda untuk menerima lalu lintas semata-mata dari penyeimbang beban. Hal ini dapat dicapai dengan menetapkan kelompok keamanan load balancer sebagai sumber dalam aturan ingress dari kelompok keamanan target.
- Jika Application Load Balancer Anda adalah target Network Load Balancer, grup keamanan untuk Application Load Balancer Anda menggunakan pelacakan koneksi untuk melacak informasi tentang lalu lintas yang berasal dari Network Load Balancer. Ini terjadi terlepas dari aturan grup keamanan yang ditetapkan untuk Application Load Balancer Anda. Untuk informasi selengkapnya, lihat [Pelacakan koneksi grup keamanan](#) di Panduan Pengguna Amazon EC2.
- Kami menyarankan Anda mengizinkan lalu lintas ICMP masuk untuk mendukung Path MTU Discovery. Untuk informasi selengkapnya, lihat [Path MTU Discovery](#) di Panduan Pengguna Amazon EC2.

### Aturan yang disarankan

Aturan berikut direkomendasikan untuk penyeimbang beban yang menghadap ke internet dengan instance sebagai target.

#### Inbound

Source	Port Range	Comment
0.0.0.0/0	<i>listener</i>	Izinkan semua lalu lintas masuk pada port listener penyeimbang beban
Outbound		
Destination	Port Range	Comment
<i>instance security group</i>	<i>instance listener</i>	Izinkan lalu lintas keluar ke instans pada port listener instans
<i>instance security group</i>	<i>health check</i>	Izinkan lalu lintas keluar ke instans pada port pemeriksaan kondisi

Aturan berikut direkomendasikan untuk penyeimbang beban internal dengan instance sebagai target.

Inbound		
Source	Port Range	Comment
<i>VPC CIDR</i>	<i>listener</i>	Izinkan lalu lintas masuk dari VPC CIDR pada port listener penyeimbang beban
Outbound		
Destination	Port Range	Comment
<i>instance security group</i>	<i>instance listener</i>	Izinkan lalu lintas keluar ke instans pada port listener instans
<i>instance security group</i>	<i>health check</i>	Izinkan lalu lintas keluar ke instans pada port pemeriksaan kondisi

Aturan berikut direkomendasikan untuk Application Load Balancer dengan instance sebagai target yang merupakan target Network Load Balancer.

### Inbound

Source	Port Range	Comment
<i>client IP addresses/ CIDR</i>	<i>alb listener</i>	Izinkan lalu lintas klien masuk pada port pendengar penyeimbang beban
<i>VPC CIDR</i>	<i>alb listener</i>	Izinkan lalu lintas klien masuk melalui AWS PrivateLink pada port pendengar penyeimbang beban
<i>VPC CIDR</i>	<i>alb listener</i>	Izinkan lalu lintas kesehatan masuk dari Network Load Balancer

### Outbound

Destination	Port Range	Comment
<i>instance security group</i>	<i>instance listener</i>	Izinkan lalu lintas keluar ke instans pada port listener instans
<i>instance security group</i>	<i>health check</i>	Izinkan lalu lintas keluar ke instans pada port pemeriksaan kondisi

## Memperbarui grup keamanan terkait

Anda dapat memperbarui grup keamanan yang terkait dengan penyeimbang beban kapan saja.

## Console

Untuk memperbarui grup keamanan

1. Buka konsol Amazon EC2 di. <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Keamanan, pilih Edit.
5. Untuk mengaitkan grup keamanan dengan penyeimbang beban Anda, pilih grup tersebut. Untuk menghapus asosiasi grup keamanan, pilih ikon X untuk grup keamanan.
6. Pilih Simpan perubahan.

## AWS CLI

Untuk memperbarui grup keamanan

Gunakan perintah [set-security-groups](#).

```
aws elbv2 set-security-groups \  
  --load-balancer-arn load-balancer-arn \  
  --security-groups sg-01dd3383691d02f42 sg-00f4e409629f1a42d
```

## CloudFormation

Untuk memperbarui grup keamanan

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:  
      Name: my-alb  
      Type: application  
      Scheme: internal  
      Subnets:  
        - !Ref subnet-AZ1  
        - !Ref subnet-AZ2  
      SecurityGroups:  
        - !Ref mySecurityGroup
```

- !Ref *myNewSecurityGroup*

## Perbarui jenis alamat IP untuk Application Load Balancer

Anda dapat mengonfigurasi Application Load Balancer Anda sehingga klien dapat berkomunikasi dengan penyeimbang beban hanya menggunakan IPv4 alamat, atau menggunakan keduanya IPv4 dan IPv6 alamat (dualstack). Load balancer berkomunikasi dengan target berdasarkan jenis alamat IP dari kelompok target. Untuk informasi selengkapnya, lihat [Jenis alamat IP](#).

### Persyaratan dualstack

- Anda dapat mengatur jenis alamat IP saat membuat penyeimbang beban dan memperbaruinya kapan saja.
- Virtual Private Cloud (VPC) dan subnet yang Anda tentukan untuk load balancer harus memiliki blok CIDR terkait. IPv6 Untuk informasi selengkapnya, lihat [IPv6alamat](#) di Panduan Pengguna Amazon EC2.
- Tabel rute untuk subnet penyeimbang beban harus merutekan IPv6 lalu lintas.
- Grup keamanan untuk penyeimbang beban harus mengizinkan IPv6 lalu lintas.
- Jaringan ACLs untuk subnet penyeimbang beban harus memungkinkan IPv6 lalu lintas.

### Console

Untuk memperbarui jenis alamat IP

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Pemetaan jaringan, pilih Edit jenis alamat IP.
5. Untuk jenis alamat IP, pilih IPv4 untuk mendukung IPv4 alamat saja, Dualstack untuk mendukung keduanya IPv4 dan IPv6 alamat, atau Dualstack tanpa publik IPv4 untuk mendukung alamat saja. IPv6
6. Pilih Simpan perubahan.

## AWS CLI

Untuk memperbarui jenis alamat IP

Gunakan perintah [set-ip-address-type](#).

```
aws elbv2 set-ip-address-type \  
  --load-balancer-arn load-balancer-arn \  
  --ip-address-type dualstack
```

## CloudFormation

Untuk memperbarui jenis alamat IP

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:  
      Name: my-alb  
      Type: application  
      Scheme: internal  
      IpAddressType: dualstack  
      Subnets:  
        - !Ref subnet-AZ1  
        - !Ref subnet-AZ2  
      SecurityGroups:  
        - !Ref mySecurityGroup
```

## Perbarui kumpulan alamat IP IPAM untuk Application Load Balancer Anda

Kumpulan alamat IP IPAM harus terlebih dahulu dibuat dalam IPAM sebelum dapat digunakan oleh Application Load Balancer Anda. Untuk informasi selengkapnya, lihat [Membawa alamat IP Anda ke IPAM](#).

## Console

Untuk memperbarui kumpulan alamat IP IPAM

1. Buka konsol Amazon EC2 di. <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Pemetaan jaringan, pilih Edit kolom IP.
5. Di bawah kolom IP, pilih Gunakan kolom IPAM untuk IPv4 alamat publik dan pilih kolom IPAM.
6. Pilih Simpan perubahan.

## AWS CLI

Untuk memperbarui kumpulan alamat IP IPAM

Gunakan perintah [modify-ip-pools](#).

```
aws elbv2 modify-ip-pools \  
  --load-balancer-arn load-balancer-arn \  
  --ipam-pools Ipv4IpamPoolId=ipam-pool-1234567890abcdef0
```

## CloudFormation

Untuk memperbarui kumpulan alamat IP IPAM

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:  
      Name: my-alb  
      Type: application  
      Scheme: internet-facing  
      IpAddressType: ipv4  
      Subnets:  
        - !Ref subnet-AZ1  
        - !Ref subnet-AZ2
```

```
SecurityGroups:  
  - !Ref mySecurityGroup  
Ipv4IpamPoolId: !Ref myIPAMPool
```

## Mengedit atribut untuk Application Load Balancer

Setelah Anda membuat Application Load Balancer, Anda dapat mengedit atributnya.

Atribut penyeimbang beban

- [Batas waktu idle koneksi](#)
- [Durasi keepalive klien HTTP](#)
- [Perlindungan penghapusan](#)
- [Mode mitigasi desync](#)
- [Pelestarian header host](#)

### Batas waktu idle koneksi

Batas waktu idle koneksi adalah periode waktu klien yang ada atau koneksi target dapat tetap tidak aktif, tanpa data dikirim atau diterima, sebelum penyeimbang beban menutup koneksi.

Untuk memastikan bahwa operasi yang panjang seperti unggahan file memiliki waktu untuk diselesaikan, kirim setidaknya 1 byte data sebelum setiap periode batas waktu idle berlalu dan tingkatkan panjang periode batas waktu idle sesuai kebutuhan. Sebaiknya konfigurasi juga batas waktu idle aplikasi Anda menjadi lebih besar daripada batas waktu idle yang dikonfigurasi untuk penyeimbang beban. Jika tidak, jika aplikasi menutup koneksi TCP ke penyeimbang beban secara tidak sengaja, penyeimbang beban mungkin mengirim permintaan ke aplikasi sebelum menerima paket yang menunjukkan bahwa koneksi ditutup. Jika ini masalahnya, maka penyeimbang beban mengirimkan kesalahan HTTP 502 Bad Gateway ke klien.

Aplikasi Load Balancer tidak mendukung bingkai HTTP/2 PING. Ini tidak mengatur ulang batas waktu idle koneksi.

Secara default, Elastic Load Balancing mengatur nilai batas waktu idle untuk penyeimbang beban Anda menjadi 60 detik.

## Console

Untuk memperbarui nilai batas waktu idle koneksi

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Atribut, pilih Edit.
5. Di bawah konfigurasi Traffic, masukkan nilai untuk batas waktu idle Connection. Rentang yang valid adalah 1 hingga 4000 detik.
6. Pilih Simpan perubahan.

## AWS CLI

Untuk memperbarui nilai batas waktu idle koneksi

Gunakan [modify-load-balancer-attributes](#) perintah dengan `idle_timeout.timeout_seconds` atribut. Rentang yang valid adalah 1 hingga 4000 detik.

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes "Key=idle_timeout.timeout_seconds,Value=120"
```

## CloudFormation

Untuk memperbarui nilai batas waktu idle koneksi

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya untuk menyertakan `idle_timeout.timeout_seconds` atribut. Rentang yang valid adalah 1 hingga 4000 detik.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:  
      Name: my-alb  
      Type: application  
      Scheme: internal  
      Subnets:
```

```
- !Ref subnet-AZ1
- !Ref subnet-AZ2
SecurityGroups:
- !Ref mySecurityGroup
LoadBalancerAttributes:
- Key: "idle_timeout.timeout_seconds"
  Value: "120"
```

## Durasi keepalive klien HTTP

Durasi keepalive klien HTTP adalah durasi maksimum waktu Application Load Balancer mempertahankan koneksi HTTP persisten ke klien. Setelah durasi keepalive klien HTTP yang dikonfigurasi berlalu, Application Load Balancer menerima satu permintaan lagi dan kemudian mengembalikan respons yang menutup koneksi dengan anggun.

Jenis respons yang dikirim oleh penyeimbang beban tergantung pada versi HTTP yang digunakan oleh koneksi klien.

- Untuk klien yang terhubung menggunakan HTTP 1.x, penyeimbang beban mengirimkan header HTTP yang berisi bidang. `Connection: close`
- Untuk klien yang terhubung menggunakan HTTP/2, penyeimbang beban mengirimkan bingkai. `GOAWAY`

Secara default, Application Load Balancer menetapkan nilai durasi keepalive klien HTTP untuk load balancer menjadi 3600 detik, atau 1 jam. Durasi keepalive klien HTTP tidak dapat dimatikan atau disetel di bawah minimum 60 detik, tetapi Anda dapat meningkatkan durasi keepalive klien HTTP, hingga maksimum 604800 detik, atau 7 hari. Application Load Balancer memulai periode durasi keepalive klien HTTP ketika koneksi HTTP ke klien awalnya dibuat. Periode durasi berlanjut ketika tidak ada lalu lintas, dan tidak diatur ulang sampai koneksi baru dibuat.

Ketika lalu lintas penyeimbang beban digeser dari Zona Ketersediaan yang terganggu menggunakan pergeseran zona atau pergeseran otomatis zona, klien dengan koneksi terbuka yang ada mungkin terus membuat permintaan terhadap lokasi yang rusak hingga klien terhubung kembali. Untuk mendukung pemulihan yang lebih cepat, pertimbangkan untuk menetapkan nilai durasi keepalive yang lebih rendah, untuk membatasi jumlah waktu klien tetap terhubung ke penyeimbang beban. Untuk informasi selengkapnya, lihat [Batasi waktu klien tetap terhubung ke titik akhir Anda](#) di Panduan Pengembang Amazon Application Recovery Controller (ARC).

**Note**

Ketika penyeimbang beban mengalihkan jenis alamat IP Application Load Balancer Anda `dualstack-without-public-ipv4` ke, penyeimbang beban menunggu semua koneksi aktif selesai. Untuk mengurangi jumlah waktu yang diperlukan untuk mengganti jenis alamat IP untuk Application Load Balancer Anda, pertimbangkan untuk menurunkan durasi keepalive klien HTTP.

Application Load Balancer menetapkan nilai durasi keepalive klien HTTP selama koneksi awal. Saat Anda memperbarui durasi keepalive klien HTTP, ini dapat menghasilkan koneksi simultan dengan nilai durasi keepalive klien HTTP yang berbeda. Koneksi yang ada mempertahankan nilai durasi keepalive klien HTTP yang diterapkan selama koneksi awal. Koneksi baru menerima nilai durasi keepalive klien HTTP yang diperbarui.

**Console**

Untuk memperbarui durasi keepalive klien

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Atribut, pilih Edit.
5. Di bawah konfigurasi Lalu lintas, masukkan nilai untuk durasi keepalive klien HTTP. Kisaran yang valid adalah 60 hingga 604800 detik.
6. Pilih Simpan perubahan.

**AWS CLI**

Untuk memperbarui durasi keepalive klien

Gunakan [modify-load-balancer-attributes](#) perintah dengan `client_keep_alive.seconds` atribut. Kisaran yang valid adalah 60 hingga 604800 detik.

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes "Key=client_keep_alive.seconds,Value=7200"
```

## CloudFormation

Untuk memperbarui durasi keepalive klien

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya untuk menyertakan `client_keep_alive.seconds` atribut. Kisaran yang valid adalah 60 hingga 604800 detik.

```
Resources:
  myLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "client_keep_alive.seconds"
          Value: "7200"
```

## Perlindungan penghapusan

Untuk mencegah penyeimbang beban terhapus secara tidak sengaja, Anda dapat mengaktifkan perlindungan penghapusan. Secara default, perlindungan penghapusan dinonaktifkan untuk penyeimbang beban Anda.

Jika Anda mengaktifkan perlindungan penghapusan untuk penyeimbang beban, Anda harus menonaktifkannya sebelum dapat menghapus penyeimbang beban.

### Console

Untuk mengaktifkan atau menonaktifkan perlindungan penghapusan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Atribut, pilih Edit.

5. Di bawah Perlindungan, aktifkan atau nonaktifkan perlindungan Penghapusan.
6. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaktifkan atau menonaktifkan perlindungan penghapusan

Gunakan [modify-load-balancer-attributes](#) perintah dengan `deletion_protection.enabled` atribut.

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes "Key=deletion_protection.enabled,Value=true"
```

## CloudFormation

Untuk mengaktifkan atau menonaktifkan perlindungan penghapusan

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya untuk menyertakan `deletion_protection.enabled` atribut.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:  
      Name: my-alb  
      Type: application  
      Scheme: internal  
      Subnets:  
        - !Ref subnet-AZ1  
        - !Ref subnet-AZ2  
      SecurityGroups:  
        - !Ref mySecurityGroup  
      LoadBalancerAttributes:  
        - Key: "deletion_protection.enabled"  
          Value: "true"
```

## Mode mitigasi desync

Mode mitigasi desync melindungi aplikasi Anda dari masalah karena desync HTTP. Penyeimbang beban mengklasifikasikan setiap permintaan berdasarkan tingkat ancamannya, memungkinkan permintaan yang aman, lalu mengurangi risiko seperti yang ditentukan oleh mode mitigasi yang Anda tentukan. Mode mitigasi desync adalah monitor, defensive, dan strictest. Default-nya adalah mode defensive yang memberikan mitigasi tahan lama terhadap HTTP desync sambil mempertahankan ketersediaan aplikasi Anda. Anda dapat beralih ke mode strictest untuk memastikan bahwa aplikasi Anda hanya menerima permintaan yang sesuai dengan [RFC 7230](#).

Pustaka [http\\_desync\\_guardian](#) menganalisis permintaan HTTP untuk mencegah serangan desync HTTP. Untuk informasi selengkapnya, lihat [HTTP Desync Guardian](#) di GitHub

### Klasifikasi

Klasifikasi adalah sebagai berikut:

- Patuh — Permintaan sesuai dengan RFC 7230 dan tidak menimbulkan ancaman keamanan yang diketahui.
- Dapat diterima — Permintaan tidak sesuai dengan RFC 7230, tetapi tidak menimbulkan ancaman keamanan yang diketahui.
- Ambigu — Permintaan tidak sesuai dengan RFC 7230, tetapi menimbulkan risiko karena berbagai server web dan proxy dapat menanganinya secara berbeda.
- Parah — Permintaan menimbulkan risiko keamanan yang tinggi. Penyeimbang beban memblokir permintaan, memberikan 400 respons ke klien, dan menutup koneksi klien.

Jika permintaan tidak sesuai dengan RFC 7230, penyeimbang beban akan menambah metrik `DesyncMitigationMode_NonCompliant_Request_Count`. Untuk informasi selengkapnya, lihat [Metrik Application Load Balancer](#).

Klasifikasi untuk setiap permintaan disertakan dalam log akses penyeimbang beban. Jika permintaan tidak sesuai, log akses menyertakan kode alasan klasifikasi. Untuk informasi selengkapnya, lihat [Alasan klasifikasi](#).

### Modus

Tabel berikut menjelaskan cara Application Load Balancer menangani permintaan berdasarkan mode dan klasifikasi.

Klasifikasi	Mode monitor	Mode defensive	Mode strictest
Patuh	Diizinkan	Diizinkan	Diizinkan
Dapat diterima	Diizinkan	Diizinkan	Diblokir
Ambigu	Diizinkan	Diizinkan <sup>1</sup>	Diblokir
Parah	Diizinkan	Diblokir	Diblokir

<sup>1</sup> Merutekan permintaan, tetapi menutup koneksi klien dan target. Anda mungkin dikenakan biaya tambahan jika penyeimbang beban menerima sejumlah besar permintaan Ambigu dalam mode Defensive. Hal ini karena peningkatan jumlah koneksi baru per detik berkontribusi terhadap Load Balancer Capacity Unit (LCU) yang digunakan per jam. Anda dapat menggunakan metrik `NewConnectionCount` untuk membandingkan cara penyeimbang beban membuat koneksi baru dalam mode Monitor dan mode Defensive.

## Console

Untuk memperbarui mode mitigasi desync

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Atribut, pilih Edit.
5. Di bawah konfigurasi Lalu Lintas, Penanganan paket, untuk mode mitigasi Desync, pilih Defensive, Strictest, atau Monitor.
6. Pilih Simpan perubahan.

## AWS CLI

Untuk memperbarui mode mitigasi desync

Gunakan `modify-load-balancer-attributes` perintah dengan `routing.http.desync_mitigation_mode` atribut. Nilai yang mungkin adalah `monitor`, `defensive`, atau `strictest`. Nilai default-nya `defensive`.

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes "Key=routing.http.desync_mitigation_mode,Value=monitor"
```

## CloudFormation

Untuk memperbarui mode mitigasi desync

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya untuk menyertakan `routing.http.desync_mitigation_mode` atribut. Nilai yang mungkin adalah `monitor`, `defensive`, atau `strictest`. Nilai default-nya `defensive`.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:  
      Name: my-alb  
      Type: application  
      Scheme: internal  
      Subnets:  
        - !Ref subnet-AZ1  
        - !Ref subnet-AZ2  
      SecurityGroups:  
        - !Ref mySecurityGroup  
      LoadBalancerAttributes:  
        - Key: "routing.http.desync_mitigation_mode"  
          Value: "monitor"
```

## Pelestarian header host

Saat Anda mengaktifkan atribut header `Preserve host`, Application Load Balancer mempertahankan Host header dalam permintaan HTTP, dan mengirimkan header ke target tanpa modifikasi apa pun. Jika Application Load Balancer menerima beberapa Host header, itu mempertahankan semuanya. Aturan pendengar hanya diterapkan pada Host header pertama yang diterima.

Secara default, ketika atribut header `Preserve host` tidak diaktifkan, Application Load Balancer memodifikasi Host header dengan cara berikut:

Ketika pelestarian header host tidak diaktifkan, dan port listener adalah port non-default: Saat tidak menggunakan port default (port 80 atau 443) kami menambahkan nomor port ke header host jika

belum ditambahkan oleh klien. Misalnya, Host header dalam permintaan HTTP dengan Host : `www.example.com` akan dimodifikasi menjadi Host : `www.example.com:8080`, jika port listener adalah port non-default seperti. `8080`

Ketika pelestarian header host tidak diaktifkan, dan port listener adalah port default (port 80 atau 443): Untuk port pendengar default (baik port 80 atau 443), kami tidak menambahkan nomor port ke header host keluar. Nomor port apa pun yang sudah ada di header host masuk, akan dihapus.

Tabel berikut menunjukkan lebih banyak contoh bagaimana Application Load Balancers memperlakukan header host dalam permintaan HTTP berdasarkan port listener.

Port pendengar	Contoh permintaan	Header host dalam permintaan	Pelestarian header host dinonaktifkan (perilaku default)	Pelestarian header host diaktifkan
Permintaan dikirim pada HTTP/HTTPS pendengar default.	GET / index.html HTTP/1.1 Host: example.com	example.com	example.com	example.com
Permintaan dikirim pada pendengar HTTP default dan header host memiliki port (misalnya, 80 atau 443).	GET / index.html HTTP/1.1 Host: example.com:80	example.com:80	example.com	example.com:80
Permintaan memiliki jalur absolut.	GET https:// dns_name/index.html HTTP/1.1 Host: example.com	example.com	dns_name	example.com

Port pendengar	Contoh permintaan	Header host dalam permintaan	Pelestarian header host dinonaktifkan (perilaku default)	Pelestarian header host diaktifkan
Permintaan dikirim pada port pendengar non-default (misalnya, 8080)	GET / index.html HTTP/1.1 Host: example.com	example.com	example.com:8080	example.com
Permintaan dikirim pada port pendengar non-default dan header host memiliki port (misalnya, 8080).	GET / index.html HTTP/1.1 Host: example.com:8080	example.com:8080	example.com:8080	example.com:8080

## Console

Untuk mengaktifkan pelestarian header host

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Di panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Atribut, pilih Edit.
5. Di bawah Penanganan paket, nyalakan header Preserve host.
6. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaktifkan pelestarian header host

Gunakan [modify-load-balancer-attributes](#) perintah dengan `routing.http.preserve_host_header.enabled` atribut yang disetel ke `true`.

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes "Key=routing.http.preserve_host_header.enabled,Value=true"
```

## CloudFormation

Untuk mengaktifkan pelestarian header host

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya untuk menyertakan `routing.http.preserve_host_header.enabled` atribut.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:  
      Name: my-alb  
      Type: application  
      Scheme: internal  
      Subnets:  
        - !Ref subnet-AZ1  
        - !Ref subnet-AZ2  
      SecurityGroups:  
        - !Ref mySecurityGroup  
      LoadBalancerAttributes:  
        - Key: "routing.http.preserve_host_header.enabled"  
          Value: "true"
```

## Menandai Application Load Balancer

Tag membantu Anda mengategorikan penyeimbang beban dengan cara yang berbeda, misalnya berdasarkan tujuan, pemilik, atau lingkungan.

Anda dapat menambahkan beberapa tag ke setiap penyeimbang beban. Jika Anda menambahkan tag dengan kunci yang sudah terkait dengan penyeimbang beban, kunci akan memperbarui nilai tag tersebut.

Setelah selesai dengan tag, Anda dapat menghapusnya dari penyeimbang beban Anda.

### Pembatasan

- Jumlah maksimum tanda per sumber daya—50

- Panjang kunci maksimum – 127 karakter Unicode
- Panjang nilai maksimum—255 karakter Unicode
- Kunci dan nilai tag peka huruf besar/kecil. Karakter yang diizinkan adalah huruf, spasi, dan angka yang dapat diwakili dalam UTF-8, ditambah karakter khusus berikut: + - = . \_:/@. Jangan gunakan spasi terkemuka atau paling belakang.
- Jangan gunakan `aws` : awalan dalam nama atau nilai tag Anda karena itu dicadangkan untuk AWS digunakan. Anda tidak dapat mengedit atau menghapus nama atau nilai tag dengan awalan ini. Tag dengan awalan ini tidak dihitung terhadap tag Anda per batas sumber daya.

## Console

Untuk memperbarui tag untuk penyeimbang beban

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Di bagian tab Tanda, pilih Kelola tanda.
5. Untuk menambahkan tag, pilih Tambahkan tag dan masukkan kunci tag dan nilai tag.
6. Untuk memperbarui tag, masukkan nilai baru di Kunci atau Nilai.
7. Untuk menghapus tanda, pilih Hapus di samping tanda.
8. Pilih Simpan perubahan.

## AWS CLI

Untuk menambahkan tag

Gunakan perintah [add-tag](#).

```
aws elbv2 add-tags \  
  --resource-arns load-balancer-arn \  
  --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

Untuk menghapus tag

Gunakan perintah [remove-tag](#).

```
aws elbv2 remove-tags \  
  --resource-arns load-balancer-arn \  
  --tags Key=project,Value=lima
```

```
--resource-arns load-balancer-arn \  
--tag-keys project department
```

## CloudFormation

Untuk menambahkan tag

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya untuk menyertakan Tags properti.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:  
      Name: my-alb  
      Type: application  
      Scheme: internal  
      Subnets:  
        - !Ref subnet-AZ1  
        - !Ref subnet-AZ2  
      SecurityGroups:  
        - !Ref mySecurityGroup  
      Tags:  
        - Key: 'project'  
          Value: 'lima'  
        - Key: 'department'  
          Value: 'digital-media'
```

## Menghapus Application Load Balancer

Segera setelah penyeimbang beban Anda tersedia, Anda akan ditagih untuk setiap jam atau sebagian jam yang tetap Anda jalankan. Saat tidak lagi membutuhkan penyeimbang beban, Anda dapat menghapusnya. Segera setelah penyeimbang beban dihapus, Anda berhenti dikenakan biaya untuk penyeimbang beban tersebut.

Anda tidak dapat menghapus penyeimbang beban jika perlindungan penghapusan diaktifkan. Untuk informasi selengkapnya, lihat [Perlindungan penghapusan](#).

Perhatikan bahwa menghapus penyeimbang beban tidak memengaruhi target terdaftarnya. Misalnya, instans EC2 Anda terus berjalan dan masih terdaftar ke grup target mereka. Untuk menghapus grup target Anda, lihat [Menghapus grup target Application Load Balancer](#).

## Catatan DNS

Jika Anda memiliki catatan DNS untuk domain Anda yang mengarah ke penyeimbang beban Anda, arahkan ke lokasi baru dan tunggu perubahan DNS diterapkan sebelum menghapus penyeimbang beban Anda.

- Jika rekaman adalah rekaman CNAME dengan Time To Live (TTL) 300 detik, tunggu setidaknya 300 detik sebelum melanjutkan ke langkah berikutnya.
- Jika catatan adalah catatan Route 53 Alias (A), tunggu setidaknya 60 detik.
- Jika menggunakan Route 53, perubahan catatan membutuhkan waktu 60 detik untuk menyebar ke semua server nama Route 53 global. Tambahkan waktu ini ke nilai TTL dari catatan yang sedang diperbarui.

## Console

Untuk menghapus penyeimbang beban

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih load balancer, lalu pilih Actions, Delete load balancer.
4. Saat diminta konfirmasi, masukkan **confirm**, lalu pilih Hapus.

## AWS CLI

Untuk menghapus penyeimbang beban

Gunakan perintah [delete-load-balancer](#).

```
aws elbv2 delete-load-balancer \  
  --load-balancer-arn load-balancer-arn
```

## Lihat peta sumber daya Application Load Balancer

Peta sumber daya Application Load Balancer menyediakan tampilan interaktif arsitektur penyeimbang beban Anda, termasuk pendengar terkait, aturan, grup target, dan target. Peta sumber daya juga menyoroti hubungan dan jalur perutean antara semua sumber daya, menghasilkan representasi visual dari konfigurasi penyeimbang beban Anda.

Untuk melihat peta sumber daya untuk Application Load Balancer

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pilih tab Resource map untuk menampilkan peta sumber daya penyeimbang beban.

## Komponen peta sumber daya

### Tampilan peta

Ada dua tampilan yang tersedia di peta sumber daya Application Load Balancer: Gambaran Umum, dan Peta Target Tidak Sehat. Ikhtisar dipilih secara default dan menampilkan semua sumber daya penyeimbang beban Anda. Memilih tampilan Peta Target Tidak Sehat hanya akan menampilkan target yang tidak sehat dan sumber daya yang terkait dengannya.

Tampilan Peta Target Tidak Sehat dapat digunakan untuk memecahkan masalah target yang gagal dalam pemeriksaan kesehatan. Untuk informasi selengkapnya, lihat [Memecahkan masalah target yang tidak sehat menggunakan peta sumber daya](#).

### Grup sumber daya

Peta sumber daya Application Load Balancer berisi empat kelompok sumber daya, satu untuk setiap jenis sumber daya. Grup sumber daya adalah Pendengar, Aturan, Grup target, dan Target.

### Ubin sumber daya

Setiap sumber daya dalam grup memiliki ubin sendiri, yang menampilkan detail tentang sumber daya tertentu.

- Melayang di atas ubin sumber daya menyoroti hubungan antara itu dan sumber daya lainnya.
- Memilih ubin sumber daya menyoroti hubungan antara itu dan sumber daya lainnya, dan menampilkan detail tambahan tentang sumber daya tersebut.
  - ketentuan aturan: Ketentuan untuk setiap aturan.
  - Ringkasan kesehatan kelompok sasaran: Jumlah target terdaftar untuk setiap status kesehatan.
  - Target status kesehatan Target status kesehatan saat ini dan deskripsi.

**Note**

Anda dapat menonaktifkan Tampilkan detail sumber daya untuk menyembunyikan detail tambahan dalam peta sumber daya.

- Setiap ubin sumber daya berisi tautan yang, ketika dipilih, menavigasi ke halaman detail sumber daya tersebut.
  - Listeners - Pilih protokol listeners: port. Sebagai contoh, HTTP:80.
  - Aturan - Pilih tindakan aturan. Sebagai contoh, Forward to target group.
  - Grup sasaran - Pilih nama grup target. Sebagai contoh, my-target-group.
  - Target - Pilih ID target. Sebagai contoh, i-1234567890abcdef0.

### Ekspor peta sumber daya

Memilih Ekspor memberi Anda opsi untuk mengekspor tampilan saat ini dari peta sumber daya Application Load Balancer Anda sebagai PDF.

## Pergeseran zona untuk Application Load Balancer Anda

Zonal shift dan zonal autoshift adalah fitur Amazon Application Recovery Controller (ARC). Dengan pergeseran zona, Anda dapat mengalihkan lalu lintas dari Availability Zone yang terganggu dengan satu tindakan. Dengan cara ini, Anda dapat terus beroperasi dari Availability Zone sehat lainnya di file AWS Region.

Dengan zonal autoshift, Anda mengizinkan AWS untuk mengalihkan lalu lintas sumber daya untuk aplikasi dari Availability Zone selama acara, atas nama Anda, untuk membantu mengurangi waktu pemulihan. AWS memulai pergeseran otomatis ketika pemantauan internal menunjukkan bahwa ada gangguan Availability Zone yang berpotensi berdampak pada pelanggan. Saat AWS memulai perpindahan otomatis, lalu lintas aplikasi ke sumber daya yang telah Anda konfigurasi untuk pergeseran otomatis zona mulai bergeser dari Availability Zone.

Saat Anda memulai pergeseran zona, penyeimbang beban Anda berhenti mengirim lalu lintas baru untuk sumber daya ke Availability Zone yang terpengaruh. ARC segera menciptakan pergeseran zona. Namun, diperlukan waktu singkat untuk menyelesaikan koneksi yang sedang berlangsung di Availability Zone, tergantung pada perilaku klien dan penggunaan kembali koneksi. Bergantung pada pengaturan DNS Anda dan faktor lainnya, koneksi yang ada dapat selesai hanya dalam beberapa

menit, atau mungkin memakan waktu lebih lama. Untuk informasi selengkapnya, lihat [Batasi waktu klien tetap terhubung ke titik akhir Anda](#) di Panduan Pengembang Amazon Application Recovery Controller (ARC).

## Daftar Isi

- [Sebelum Anda memulai pergeseran zona](#)
- [Penyeimbangan beban lintas zona](#)
- [Pengesampingan administratif pergeseran zona](#)
- [Aktifkan pergeseran zona untuk Application Load Balancer](#)
- [Mulai pergeseran zona untuk Application Load Balancer Anda](#)
- [Memperbarui pergeseran zona untuk Application Load Balancer](#)
- [Membatalkan pergeseran zona untuk Application Load Balancer](#)

## Sebelum Anda memulai pergeseran zona

- Pergeseran zona dinonaktifkan secara default dan harus diaktifkan pada setiap Application Load Balancer. Untuk informasi selengkapnya, lihat [Aktifkan pergeseran zona untuk Application Load Balancer](#).
- Anda dapat memulai pergeseran zona untuk menyeimbangkan beban tertentu hanya untuk satu Availability Zone. Anda tidak dapat memulai pergeseran zona untuk beberapa Availability Zone.
- AWS secara proaktif menghapus alamat IP penyeimbang beban zonal dari DNS ketika beberapa masalah infrastruktur berdampak pada layanan. Selalu periksa kapasitas Availability Zone saat ini sebelum Anda memulai pergeseran zona. Jika penyeimbang beban Anda mematikan penyeimbang beban lintas zona dan Anda menggunakan pergeseran zona untuk menghapus alamat IP penyeimbang beban zonal, Availability Zone yang terpengaruh oleh pergeseran zona juga kehilangan kapasitas target.

Untuk informasi selengkapnya, lihat [Praktik terbaik untuk pergeseran zona di ARC di Panduan Pengembang Amazon Application Recovery Controller \(ARC\)](#).

## Penyeimbangan beban lintas zona

Ketika pergeseran zona dimulai pada Application Load Balancer dengan penyeimbangan beban lintas zona diaktifkan, semua lalu lintas ke target diblokir di zona ketersediaan yang terpengaruh, dan alamat IP zona dihapus dari DNS.

## Manfaat:

- Pemulihan lebih cepat dari kegagalan zona ketersediaan.
- Kemampuan untuk memindahkan lalu lintas ke zona ketersediaan yang sehat jika kegagalan terdeteksi di zona ketersediaan.
- Anda dapat menguji integritas aplikasi dengan mensimulasikan dan mengidentifikasi kegagalan untuk mencegah downtime yang tidak direncanakan.

## Pengesampingan administratif pergeseran zona

Target yang termasuk dalam Application Load Balancer termasuk status baru `AdministrativeOverride`, yang independen dari negara. `TargetHealth`

Ketika pergeseran zona dimulai untuk Application Load Balancer, semua target dalam zona yang digeser dari dianggap diganti secara administratif. Application Load Balancer berhenti merutekan lalu lintas baru ke target yang diganti secara administratif. Koneksi yang ada tetap utuh sampai ditutup secara organik.

`AdministrativeOverride`Keadaan yang mungkin adalah:

tidak diketahui

Status tidak dapat disebar karena kesalahan internal

`no_override`

Tidak ada penggantian saat ini aktif pada target

`zonal_shift_active`

Pergeseran zona aktif di Zona Ketersediaan target

## Aktifkan pergeseran zona untuk Application Load Balancer

Pergeseran zona dinonaktifkan secara default dan harus diaktifkan pada setiap Application Load Balancer. Ini memastikan bahwa Anda dapat memulai pergeseran zona hanya menggunakan Application Load Balancers tertentu yang Anda inginkan. Untuk informasi selengkapnya, lihat [the section called “Peralihan zona”](#).

## Console

Untuk mengaktifkan pergeseran zona

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah PENYEIMBANGAN BEBAN, pilih Penyeimbang beban.
3. Pilih Application Load Balancer.
4. Pada tab Atribut, pilih Edit.
5. Di bawah konfigurasi perutean Availability Zone, untuk integrasi pergeseran zona ARC, pilih Aktifkan.
6. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaktifkan pergeseran zona

Gunakan [modify-load-balancer-attributes](#) perintah dengan `zonal_shift.config.enabled` atribut.

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes "Key=zonal_shift.config.enabled,Value=true"
```

## CloudFormation

Untuk mengaktifkan pergeseran zona

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya untuk menyertakan `zonal_shift.config.enabled` atribut.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:  
      Name: my-alb  
      Type: application  
      Scheme: internal  
      IpAddressType: dualstack  
      Subnets:  
        - !Ref subnet-AZ1
```

```
- !Ref subnet-AZ2
SecurityGroups:
- !Ref mySecurityGroup
LoadBalancerAttributes:
  -Key: "zonal_shift.config.enabled"
  Value: "true"
```

## Mulai pergeseran zona untuk Application Load Balancer Anda

Pergeseran zona di ARC memungkinkan Anda memindahkan lalu lintas sementara untuk sumber daya yang didukung dari Availability Zone sehingga aplikasi Anda dapat terus beroperasi secara normal dengan Availability Zone lainnya di suatu AWS Wilayah.

### Prasyarat

Sebelum memulai, verifikasi bahwa Anda [mengaktifkan pergeseran zona](#) untuk penyeimbang beban.

### Console

Prosedur ini menjelaskan cara memulai pergeseran zona menggunakan konsol Amazon EC2. Untuk langkah-langkah memulai pergeseran zona menggunakan konsol ARC, lihat [Memulai pergeseran zona di Panduan](#) Pengembang Amazon Application Recovery Controller (ARC).

Untuk memulai pergeseran zona

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah PENYEIMBANGAN BEBAN, pilih Penyeimbang beban.
3. Pilih Application Load Balancer.
4. Pada tab Integrasi, perluas Amazon Application Recovery Controller (ARC) dan pilih Mulai pergeseran zona.
5. Pilih Availability Zone yang ingin Anda pindahkan lalu lintas.
6. Pilih atau masukkan kedaluwarsa untuk pergeseran zona. Pergeseran zona awalnya dapat diatur dari 1 menit hingga tiga hari (72 jam).

Semua pergeseran zona bersifat sementara. Anda harus menetapkan kedaluwarsa, tetapi Anda dapat memperbarui shift aktif nanti untuk menetapkan kedaluwarsa baru.

7. Masukkan komentar. Anda dapat memperbarui pergeseran zona nanti untuk mengedit komentar.

8. Pilih kotak centang untuk mengetahui bahwa memulai pergeseran zona mengurangi kapasitas aplikasi Anda dengan mengalihkan lalu lintas dari Availability Zone.
9. Pilih Konfirmasi.

## AWS CLI

Untuk memulai pergeseran zona

Gunakan [start-zonal-shift](#) perintah Amazon Application Recovery Controller (ARC).

```
aws arc-zonal-shift start-zonal-shift \  
  --resource-identifier load-balancer-arn \  
  --away-from use2-az2 \  
  --expires-in 2h \  
  --comment "zonal shift due to scheduled maintenance"
```

## Memperbarui pergeseran zona untuk Application Load Balancer

Anda dapat memperbarui pergeseran zona untuk menetapkan kedaluwarsa baru, atau mengedit atau mengganti komentar untuk pergeseran zona.

### Console

Prosedur ini menjelaskan cara memperbarui pergeseran zona menggunakan konsol Amazon EC2. Untuk langkah-langkah memperbarui pergeseran zona menggunakan konsol Amazon Application Recovery Controller (ARC), lihat [Memperbarui pergeseran zona](#) di Panduan Pengembang Amazon Application Recovery Controller (ARC).

Untuk memperbarui pergeseran zona

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah PENYEIMBANGAN BEBAN, pilih Penyeimbang beban.
3. Pilih Application Load Balancer dengan pergeseran zona aktif.
4. Pada tab Integrasi, perluas Amazon Application Recovery Controller (ARC) dan pilih Update zonal shift.

Ini membuka konsol ARC untuk melanjutkan proses pembaruan.

5. (Opsional) Untuk Mengatur kedaluwarsa pergeseran zona, pilih atau masukkan kedaluwarsa.

6. (Opsional) Untuk Komentar, secara opsional edit komentar yang ada atau masukkan komentar baru.
7. Pilih Perbarui.

## AWS CLI

Untuk memperbarui pergeseran zona

Gunakan [update-zonal-shift](#) perintah Amazon Application Recovery Controller (ARC).

```
aws arc-zonal-shift update-zonal-shift \  
  --zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf57EXAMPLE \  
  --expires-in 1h \  
  --comment "extending zonal shift for scheduled maintenance"
```

## Membatalkan pergeseran zona untuk Application Load Balancer

Anda dapat membatalkan pergeseran zona kapan saja sebelum kedaluwarsa. Anda dapat membatalkan pergeseran zona yang Anda mulai, atau pergeseran zona yang AWS dimulai untuk sumber daya untuk latihan yang dijalankan untuk pergeseran otomatis zona.

### Console

Prosedur ini menjelaskan cara membatalkan pergeseran zona menggunakan konsol Amazon EC2. Untuk langkah-langkah membatalkan pergeseran zona menggunakan konsol Amazon Application Recovery Controller (ARC), lihat [Membatalkan pergeseran zona di Panduan Pengembang Amazon Application Recovery Controller \(ARC\)](#).

Untuk membatalkan pergeseran zona

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah PENYEIMBANGAN BEBAN, pilih Penyeimbang beban.
3. Pilih Application Load Balancer dengan pergeseran zona aktif.
4. Pada tab Integrasi, di bawah Amazon Application Recovery Controller (ARC), pilih Batalkan pergeseran zona.

Ini membuka konsol ARC untuk melanjutkan proses pembatalan.

5. Pilih Batalkan pergeseran zona.

6. Ketika diminta untuk mengonfirmasi, pilih Konfirmasi.

## AWS CLI

Untuk membatalkan pergeseran zona

Gunakan [cancel-zonal-shift](#) perintah Amazon Application Recovery Controller (ARC).

```
aws arc-zonal-shift cancel-zonal-shift \  
--zonal-shift-id 9ac9ec1e-1df1-0755-3dc5-8cf57EXAMPLE
```

## Pemesanan kapasitas untuk Application Load Balancer Anda

Pemesanan Load balancer Capacity Unit (LCU) memungkinkan Anda untuk memesan kapasitas minimum statis untuk penyeimbang beban Anda. Application Load Balancers secara otomatis menskalakan untuk mendukung beban kerja yang terdeteksi dan memenuhi kebutuhan kapasitas. Ketika kapasitas minimum dikonfigurasi, penyeimbang beban Anda terus meningkat atau turun berdasarkan lalu lintas yang diterima, tetapi juga mencegah kapasitas menjadi lebih rendah dari kapasitas minimum yang dikonfigurasi.

Pertimbangkan untuk menggunakan reservasi LCU dalam situasi berikut:

- Anda memiliki acara mendatang yang akan memiliki lalu lintas tinggi yang tiba-tiba dan tidak biasa dan ingin memastikan penyeimbang beban Anda dapat mendukung lonjakan lalu lintas yang tiba-tiba selama acara berlangsung.
- Anda memiliki lalu lintas runcing yang tidak dapat diprediksi karena sifat beban kerja Anda untuk waktu yang singkat.
- Anda menyiapkan penyeimbang beban ke on-board atau memigrasikan layanan Anda pada waktu mulai tertentu dan perlu memulai dengan kapasitas tinggi alih-alih menunggu auto-scaling diterapkan.
- Anda memigrasikan beban kerja antara penyeimbang beban dan ingin mengonfigurasi tujuan agar sesuai dengan skala sumber.

Perkirakan kapasitas yang Anda butuhkan

Saat menentukan jumlah kapasitas yang harus Anda pesan untuk penyeimbang beban, sebaiknya lakukan pengujian beban atau meninjau data beban kerja historis yang mewakili lalu lintas yang

akan datang yang Anda harapkan. Menggunakan konsol Elastic Load Balancing, Anda dapat memperkirakan berapa banyak kapasitas yang perlu Anda pesan berdasarkan lalu lintas yang ditinjau.

Atau, Anda dapat menggunakan CloudWatch metrik PeakLCUs untuk menentukan tingkat kapasitas yang dibutuhkan. PeakLCUsMetrik memperhitungkan puncak dalam pola lalu lintas Anda yang harus diskalakan oleh penyeimbang beban di semua dimensi penskalaan untuk mendukung beban kerja Anda. PeakLCUsMetrik berbeda dari ConsumedLCUs metrik, yang hanya menggabungkan dimensi penagihan lalu lintas Anda. Menggunakan PeakLCUs metrik disarankan untuk memastikan reservasi LCU Anda memadai selama penskalaan penyeimbang beban. Saat memperkirakan kapasitas, gunakan `per menitSum. PeakLCUs`

Jika Anda tidak memiliki data beban kerja historis untuk referensi dan tidak dapat melakukan pengujian beban, Anda dapat memperkirakan kapasitas yang dibutuhkan menggunakan kalkulator reservasi LCU. Kalkulator reservasi LCU menggunakan data berdasarkan AWS pengamatan beban kerja historis dan mungkin tidak mewakili beban kerja spesifik Anda. Untuk informasi selengkapnya, lihat Kalkulator [Reservasi Unit Kapasitas Load Balancer](#).

Nilai minimum dan maksimum untuk reservasi LCU

Total permintaan reservasi harus minimal 100 LCU. Nilai maksimum ditentukan oleh kuota untuk akun Anda. Untuk informasi selengkapnya, lihat [the section called “Unit Kapasitas Load Balancer”](#).

## Minta reservasi Load balancer Capacity Unit untuk Application Load Balancer

Sebelum Anda menggunakan reservasi LCU, tinjau hal-hal berikut:

- Kapasitas dicadangkan di tingkat regional dan didistribusikan secara merata di seluruh zona ketersediaan. Konfirmasikan bahwa Anda memiliki cukup target yang didistribusikan secara merata di setiap zona ketersediaan sebelum mengaktifkan reservasi LCU.
- Permintaan reservasi LCU dipenuhi berdasarkan first come first serve, dan tergantung pada kapasitas yang tersedia untuk suatu zona pada saat itu. Sebagian besar permintaan biasanya dipenuhi dalam beberapa menit, tetapi dapat memakan waktu hingga beberapa jam.
- Untuk memperbarui reservasi yang ada, permintaan sebelumnya harus disediakan atau gagal. Anda dapat meningkatkan kapasitas cadangan sebanyak yang Anda butuhkan, namun Anda hanya dapat mengurangi kapasitas cadangan dua kali per hari.

- Anda akan terus dikenakan biaya untuk kapasitas yang dipesan atau disediakan sampai mereka dihentikan atau dibatalkan.

## Console

Untuk meminta reservasi LCU

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih nama penyeimbang beban.
4. Pada tab Kapasitas, pilih Edit Reservasi LCU.
5. Pilih Estimasi berbasis referensi historis.
6. Pilih periode referensi untuk melihat tingkat LCU cadangan yang direkomendasikan.
7. Jika Anda tidak memiliki beban kerja referensi historis, Anda dapat memilih Perkiraan manual dan memasukkan jumlah yang akan LCUs dipesan.
8. Pilih Simpan.

## AWS CLI

Untuk meminta reservasi LCU

Gunakan perintah [modify-capacity-reservation](#).

```
aws elbv2 modify-capacity-reservation \  
  --load-balancer-arn load-balancer-arn \  
  --minimum-load-balancer-capacity CapacityUnits=100
```

## CloudFormation

Untuk meminta reservasi LCU

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:
```

```
Name: my-alb
Type: application
Scheme: internal
Subnets:
  - !Ref subnet-AZ1
  - !Ref subnet-AZ2
SecurityGroups:
  - !Ref mySecurityGroup
MinimumLoadBalancerCapacity:
  CapacityUnits: 100
```

## Perbarui atau batalkan pemesanan Unit Kapasitas Load Balancer untuk Application Load Balancer Anda

Jika pola lalu lintas untuk penyeimbang beban Anda berubah, Anda dapat memperbarui atau membatalkan reservasi LCU untuk penyeimbang beban Anda. Status reservasi LCU harus disediakan.

### Console

Untuk memperbarui atau membatalkan reservasi LCU

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih nama penyeimbang beban.
4. Pada tab Kapasitas, lakukan salah satu hal berikut:
  - a. Untuk memperbarui reservasi LCU pilih Edit Reservasi LCU.
  - b. Untuk membatalkan reservasi LCU, pilih Batalkan Kapasitas.

### AWS CLI

Untuk membatalkan reservasi LCU

Gunakan perintah [modify-capacity-reservation](#).

```
aws elbv2 modify-capacity-reservation \  
  --load-balancer-arn load-balancer-arn \  
  --
```

```
--reset-capacity-reservation
```

## Monitor reservasi Load Balancer Capacity Unit untuk Application Load Balancer Anda

### Status reservasi

Berikut ini adalah nilai status yang memungkinkan untuk reservasi LCU:

- `pending`- Menunjukkan reservasi itu sedang dalam proses penyediaan.
- `provisioned`- Menunjukkan kapasitas cadangan siap dan tersedia untuk digunakan.
- `failed`- Menunjukkan permintaan tidak dapat diselesaikan pada saat itu.
- `rebalancing`- Menunjukkan zona ketersediaan telah ditambahkan atau dihapus dan penyeimbang beban menyeimbangkan kembali kapasitas.

### Pemanfaatan LCU

`ReservedLCUs` Metrik dilaporkan per menit. Kapasitas dicadangkan setiap jam. Misalnya, jika Anda memiliki reservasi LCU 6.000, total satu jam untuk `ReservedLCUs` adalah 6.000, dan total satu menit adalah 100. Untuk menentukan penggunaan LCU cadangan Anda, lihat metrik `PeakLCUs`. Anda dapat mengatur CloudWatch alarm untuk membandingkan per menit `Sum` dengan nilai kapasitas cadangan Anda, atau per jam `SumReservedLCUs`, untuk menentukan apakah Anda telah memesan kapasitas yang cukup untuk memenuhi kebutuhan Anda. `PeakLCUs`

### Console

Untuk melihat status reservasi LCU

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih nama penyeimbang beban.
4. Pada tab Kapasitas, Anda dapat melihat Status Reservasi dan nilai LCU Cadangan.

### AWS CLI

Untuk memantau status reservasi LCU

Gunakan perintah [describe-capacity-reservation](#).

```
aws elbv2 describe-capacity-reservation \  
--load-balancer-arn load-balancer-arn
```

## Integrasi untuk Application Load Balancer

Anda dapat mengoptimalkan arsitektur Application Load Balancer Anda dengan mengintegrasikan dengan beberapa AWS layanan lain untuk meningkatkan kinerja, keamanan, dan ketersediaan aplikasi Anda.

Integrasi penyeimbang beban

- [Pengontrol Pemulihan Aplikasi Amazon \(ARC\)](#)
- [Amazon CloudFront + AWS WAF](#)
- [AWS Global Accelerator](#)
- [AWS Config](#)
- [AWS WAF](#)

## Pengontrol Pemulihan Aplikasi Amazon (ARC)

Amazon Application Recovery Controller (ARC) membantu Anda mengalihkan lalu lintas untuk penyeimbang beban dari Availability Zone yang terganggu ke Availability Zone yang sehat di Wilayah yang sama. Menggunakan pergeseran zona mengurangi durasi dan tingkat keparahan pemadaman listrik, masalah perangkat keras, atau masalah perangkat lunak di Availability Zone pada aplikasi Anda.

Untuk informasi selengkapnya, lihat [Pergeseran zona untuk Application Load Balancer Anda](#).

## Amazon CloudFront + AWS WAF

Amazon CloudFront adalah layanan web yang membantu meningkatkan kinerja, ketersediaan, dan keamanan aplikasi Anda yang digunakan AWS. CloudFront bertindak sebagai titik masuk tunggal terdistribusi untuk aplikasi web Anda yang menggunakan Application Load Balancers. Ini memperluas jangkauan Application Load Balancer Anda secara global, memungkinkannya melayani pengguna secara efisien dari lokasi tepi terdekat, mengoptimalkan pengiriman konten dan mengurangi latensi bagi pengguna di seluruh dunia. Caching konten otomatis di lokasi tepi ini

secara signifikan mengurangi beban pada Application Load Balancer Anda, meningkatkan kinerja dan skalabilitasnya.

Integrasi satu klik yang tersedia di konsol Elastic Load Balancing membuat distribusi dengan perlindungan keamanan CloudFront yang AWS WAF direkomendasikan, dan mengaitkannya ke Application Load Balancer Anda. AWS WAF Perlindungan memblokir terhadap eksploitasi web umum sebelum mencapai penyeimbang beban Anda. Anda dapat mengakses CloudFront distribusi dan dasbor keamanan yang sesuai dari tab Integrasi penyeimbang beban di konsol. Untuk informasi selengkapnya, lihat [Mengelola perlindungan AWS WAF keamanan di dasbor CloudFront keamanan](#) di Panduan CloudFront Pengembang Amazon dan [Memperkenalkan Dasbor CloudFront Keamanan, CDN Terpadu, dan Pengalaman Keamanan](#) di [aws.amazon.com/blogs](https://aws.amazon.com/blogs).

Sebagai praktik keamanan terbaik, konfigurasi grup keamanan Application Load Balancer yang menghadap ke internet untuk mengizinkan lalu lintas masuk hanya dari daftar awalan yang AWS dikelola CloudFront, dan hapus aturan masuk lainnya. Untuk informasi selengkapnya, lihat [Menggunakan daftar awalan CloudFront terkelola](#), [Mengkonfigurasi CloudFront untuk menambahkan header HTTP kustom ke permintaan](#), dan [Mengonfigurasi Application Load Balancer untuk hanya meneruskan permintaan yang berisi header tertentu](#) di Panduan Pengembang CloudFront Amazon >.

#### Note

CloudFront hanya mendukung sertifikat ACM di wilayah us-east-1 AS Timur (Virginia Utara). Jika Application Load Balancer Anda memiliki pendengar HTTPS yang dikonfigurasi dengan sertifikat ACM di wilayah selain us-east-1, Anda harus mengubah koneksi CloudFront asal dari HTTPS ke HTTP, atau memberikan sertifikat ACM di wilayah AS Timur (Virginia N.) dan melampirkannya ke distribusi Anda. CloudFront

## AWS Global Accelerator

Untuk mengoptimalkan ketersediaan, kinerja, dan keamanan aplikasi, buat akselerator untuk penyeimbang beban Anda. Akselerator mengarahkan lalu lintas melalui jaringan AWS global ke alamat IP statis yang berfungsi sebagai titik akhir tetap di Wilayah terdekat dengan klien. AWS Global Accelerator dilindungi oleh Shield Standard, yang meminimalkan waktu henti dan latensi aplikasi dari DDo serangan S.

Untuk informasi selengkapnya, lihat [Menambahkan akselerator saat Anda membuat penyeimbang beban](#) di Panduan AWS Global Accelerator Pengembang.

## AWS Config

Untuk mengoptimalkan pemantauan dan kepatuhan penyeimbang beban Anda, siapkan AWS Config. AWS Config memberikan tampilan terperinci tentang konfigurasi AWS sumber daya di AWS akun Anda. Ini termasuk bagaimana sumber daya terkait satu sama lain dan bagaimana mereka dikonfigurasi di masa lalu sehingga Anda dapat melihat bagaimana konfigurasi dan hubungan berubah dari waktu ke waktu. AWS Config merampingkan audit, kepatuhan, dan pemecahan masalah.

Lihat informasi selengkapnya di [Panduan Developer AWS Config](#).

## AWS WAF

Anda dapat menggunakan AWS WAF Application Load Balancer untuk mengizinkan atau memblokir permintaan berdasarkan aturan dalam daftar kontrol akses web (web ACL).

Secara default, jika penyeimbang beban tidak bisa mendapatkan respons dari AWS WAF, ia mengembalikan kesalahan HTTP 500 dan tidak meneruskan permintaan. Jika Anda membutuhkan penyeimbang beban untuk meneruskan permintaan ke target meskipun tidak dapat dihubungi AWS WAF, Anda dapat mengaktifkan AWS WAF gagal terbuka.

Web yang telah ditentukan sebelumnya ACLs

Saat mengaktifkan AWS WAF integrasi, Anda dapat memilih untuk secara otomatis membuat ACL web baru dengan aturan yang telah ditentukan sebelumnya. ACL web yang telah ditentukan sebelumnya mencakup tiga aturan AWS terkelola yang menawarkan perlindungan terhadap ancaman keamanan yang paling umum.

- `AWSManagedRulesAmazonIpReputationList`- Grup aturan daftar reputasi IP Amazon memblokir alamat IP yang biasanya terkait dengan bot atau ancaman lainnya. Untuk informasi selengkapnya, lihat [grup aturan terkelola daftar reputasi IP Amazon](#) di Panduan AWS WAF Pengembang.
- `AWSManagedRulesCommonRuleSet` [Kelompok aturan set inti \(CRS\) memberikan perlindungan terhadap eksploitasi berbagai kerentanan, termasuk beberapa risiko tinggi dan kerentanan yang umum terjadi yang dijelaskan dalam publikasi OWASP seperti OWASP Top 10.](#) Untuk informasi selengkapnya, lihat Grup [aturan terkelola kumpulan aturan inti \(CRS\)](#) di Panduan AWS WAF Pengembang.
- `AWSManagedRulesKnownBadInputsRuleSet`- Kelompok aturan masukan buruk yang diketahui memblokir pola permintaan yang diketahui tidak valid dan terkait dengan eksploitasi atau

penemuan kerentanan. Untuk informasi selengkapnya, lihat [Grup aturan terkelola masukan buruk yang diketahui](#) di Panduan AWS WAF Pengembang.

Untuk informasi selengkapnya, lihat [Menggunakan web ACLs AWS WAF di](#) Panduan AWS WAF Pengembang.

# Listener untuk Application Load Balancer Anda

Listener adalah proses yang memeriksa permintaan koneksi, menggunakan protokol dan port yang Anda konfigurasi. Sebelum Anda mulai menggunakan Application Load Balancer, Anda harus menambahkan setidaknya satu pendengar. Jika penyeimbang beban Anda tidak memiliki pendengar, ia tidak dapat menerima lalu lintas dari klien. Aturan yang Anda tetapkan untuk pendengar menentukan cara penyeimbang beban merutekan permintaan ke target yang Anda daftarkan, seperti instans EC2.

## Daftar Isi

- [Konfigurasi listener](#)
- [Atribut pendengar](#)
- [Tindakan default](#)
- [Membuat listener HTTP untuk Application Load Balancer Anda](#)
- [Sertifikat SSL untuk Application Load Balancer](#)
- [Kebijakan keamanan untuk Application Load Balancer](#)
- [Buat listener HTTPS untuk Application Load Balancer Anda](#)
- [Perbarui listener HTTPS untuk Application Load Balancer Anda](#)
- [Aturan listener untuk Application Load Balancer Anda](#)
- [Otentikasi timbal balik dengan TLS di Application Load Balancer](#)
- [Mengautentikasi pengguna menggunakan Application Load Balancer](#)
- [Verifikasi JWTs menggunakan Application Load Balancer](#)
- [Header HTTP dan Application Load Balancer](#)
- [Modifikasi header HTTP untuk Application Load Balancer](#)
- [Menghapus listener untuk Application Load Balancer Anda](#)

## Konfigurasi listener

Listener mendukung protokol dan port berikut ini:

- Protokol: HTTP, HTTPS
- Port: 1-65535

Anda dapat menggunakan listener HTTPS untuk memindahkan pekerjaan enkripsi dan dekripsi ke penyeimbang beban Anda sehingga aplikasi Anda dapat fokus pada logika bisnisnya. Jika protokol listener adalah HTTPS, Anda harus men-deploy setidaknya satu sertifikat server SSL pada listener. Untuk informasi selengkapnya, lihat [Buat listener HTTPS untuk Application Load Balancer Anda](#).

Jika Anda harus memastikan bahwa target mendekripsi lalu lintas HTTPS alih-alih penyeimbang beban, Anda dapat membuat Network Load Balancer dengan pendengar TCP di port 443. Dengan pendengar TCP, penyeimbang beban meneruskan lalu lintas terenkripsi ke target tanpa mendekripsi. Untuk informasi selengkapnya, lihat [Panduan Pengguna untuk Network Load Balancer](#).

## WebSockets

Application Load Balancers memberikan dukungan asli untuk WebSockets. Anda dapat meng-upgrade koneksi HTTP/1.1 yang ada ke koneksi WebSocket (wsatau wss) dengan menggunakan upgrade koneksi HTTP. Saat Anda memutakhirkan, koneksi TCP yang digunakan untuk permintaan (ke penyeimbang beban dan juga target) menjadi WebSocket koneksi persisten antara klien dan target melalui penyeimbang beban. Anda dapat menggunakan WebSockets dengan pendengar HTTP dan HTTPS. Opsi yang Anda pilih untuk pendengar Anda berlaku untuk WebSocket koneksi serta lalu lintas HTTP. Websockets tidak didukung untuk permintaan yang dirutekan ke grup target yang telah mengaktifkan pengoptimal target. Untuk informasi selengkapnya, lihat [Cara Kerja WebSocket Protokol](#) di Panduan CloudFront Pengembang Amazon.

## HTTP/2

Application Load Balancer memberikan dukungan asli untuk HTTP/2 dengan listener HTTPS. Anda dapat mengirim hingga 128 permintaan secara paralel menggunakan satu koneksi HTTP/2. Anda dapat menggunakan versi protokol untuk mengirim permintaan ke target menggunakan HTTP/2. Untuk informasi selengkapnya, lihat [Versi protokol](#). Karena HTTP/2 menggunakan koneksi front-end secara lebih efisien, Anda mungkin melihat lebih sedikit koneksi antara klien dan penyeimbang beban. Anda tidak dapat menggunakan fitur server-push HTTP/2.

Otentikasi TLS bersama untuk Application Load Balancers mendukung HTTP/2 dalam mode passthrough dan verifikasi. Untuk informasi selengkapnya, lihat [Otentikasi timbal balik dengan TLS di Application Load Balancer](#).

Untuk informasi lebih lanjut, lihat [Perutean permintaan](#) di Panduan Pengguna Elastic Load Balancing.

## Atribut pendengar

Berikut ini adalah atribut listener untuk Application Load Balancers:

`routing.http.request.x_amzn_mtls_clientcert_serial_number.header_name`

Memungkinkan Anda untuk memodifikasi nama header header permintaan HTTP X-Amzn-Mtls-Clientcert-Serial-Number.

`routing.http.request.x_amzn_mtls_clientcert_issuer.header_name`

Memungkinkan Anda untuk memodifikasi nama header header permintaan HTTP X-Amzn-Mtls-Clientcert-Issuer.

`routing.http.request.x_amzn_mtls_clientcert_subject.header_name`

Memungkinkan Anda untuk memodifikasi nama header header permintaan HTTP X-Amzn-Mtls-Clientcert-Subject.

`routing.http.request.x_amzn_mtls_clientcert_validity.header_name`

Memungkinkan Anda untuk memodifikasi nama header header permintaan HTTP X-Amzn-Mtls-Clientcert-Validity.

`routing.http.request.x_amzn_mtls_clientcert_leaf.header_name`

Memungkinkan Anda untuk memodifikasi nama header header permintaan HTTP X-Amzn-Mtls-Clientcert-Leaf.

`routing.http.request.x_amzn_mtls_clientcert.header_name`

Memungkinkan Anda untuk memodifikasi nama header header permintaan HTTP X-Amzn-Mtls-Clientcert.

`routing.http.request.x_amzn_tls_version.header_name`

Memungkinkan Anda untuk memodifikasi nama header header permintaan HTTP X-Amzn-Tls-Version.

`routing.http.request.x_amzn_tls_cipher_suite.header_name`

Memungkinkan Anda untuk memodifikasi nama header header permintaan HTTP X-Amzn-Tls-Cipher-Suite.

`routing.http.response.server.enabled`

Memungkinkan Anda untuk mengizinkan atau menghapus header server respon HTTP.

`routing.http.response.strict_transport_security.header_value`

Menginformasikan browser bahwa situs hanya boleh diakses menggunakan HTTPS, dan bahwa setiap upaya future untuk mengaksesnya menggunakan HTTP harus secara otomatis dikonversi ke HTTPS.

`routing.http.response.access_control_allow_origin.header_value`

Menentukan asal mana yang diizinkan untuk mengakses server.

`routing.http.response.access_control_allow_methods.header_value`

Mengembalikan metode HTTP yang diizinkan saat mengakses server dari asal yang berbeda.

`routing.http.response.access_control_allow_headers.header_value`

Menentukan header mana yang dapat digunakan selama permintaan.

`routing.http.response.access_control_allow_credentials.header_value`

Menunjukkan apakah browser harus menyertakan kredensyal seperti cookie atau otentikasi saat membuat permintaan.

`routing.http.response.access_control_expose_headers.header_value`

Mengembalikan header yang browser dapat mengekspos ke klien yang meminta.

`routing.http.response.access_control_max_age.header_value`

Menentukan berapa lama hasil permintaan preflight dapat di-cache, dalam hitungan detik.

`routing.http.response.content_security_policy.header_value`

Menentukan pembatasan yang diberlakukan oleh browser untuk membantu meminimalkan risiko jenis ancaman keamanan tertentu.

`routing.http.response.x_content_type_options.header_value`

Menunjukkan apakah tipe MIME yang diiklankan di header Content-Type harus diikuti dan tidak diubah.

`routing.http.response.x_frame_options.header_value`

Menunjukkan apakah browser diizinkan untuk merender halaman dalam bingkai, iframe, embed atau objek.

## Tindakan default

Setiap pendengar memiliki tindakan default, juga dikenal sebagai aturan default. Aturan default tidak dapat dihapus dan selalu dilakukan terakhir. Anda dapat membuat aturan tambahan. Aturan-aturan ini terdiri dari prioritas, satu atau lebih tindakan, dan satu atau lebih kondisi. Anda dapat menambahkan atau mengedit peraturan kapan saja. Untuk informasi selengkapnya, lihat [Aturan pendengar](#).

## Membuat listener HTTP untuk Application Load Balancer Anda

Pendengar memeriksa permintaan koneksi. Anda menentukan listener saat membuat penyeimbang beban, dan Anda dapat menambahkan listener ke penyeimbang beban kapan Anda saja.

Informasi di halaman ini membantu Anda membuat listener HTTP untuk penyeimbang beban Anda. Untuk menambahkan listener HTTPS ke penyeimbang beban Anda, lihat [Buat listener HTTPS untuk Application Load Balancer Anda](#).

## Prasyarat

- Untuk menambahkan tindakan maju ke peraturan listener default, Anda harus menentukan grup target yang tersedia. Untuk informasi selengkapnya, lihat [Buat grup target untuk Application Load Balancer Anda](#).
- Anda dapat menentukan grup target yang sama di beberapa pendengar, tetapi pendengar ini harus termasuk dalam penyeimbang beban yang sama. Untuk menggunakan grup target dengan penyeimbang beban, Anda harus memverifikasi bahwa grup tersebut tidak digunakan oleh pendengar untuk penyeimbang beban lainnya.

## Menambahkan listener HTTP

Anda mengonfigurasi listener dengan protokol dan port untuk koneksi dari klien ke load balancer, dan grup target untuk aturan listener default. Untuk informasi selengkapnya, lihat [Konfigurasi listener](#).

Untuk menambahkan aturan pendengar lain, lihat [Aturan pendengar](#).

## Console

Untuk menambahkan pendengar HTTP

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>

2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Listeners and rules, pilih Add listener.
5. Untuk Protokol, pilih HTTP. Simpan port default atau masukkan port yang berbeda.
6. Untuk tindakan Default, pilih salah satu tindakan routing berikut dan berikan informasi yang diperlukan:
  - Teruskan ke grup sasaran - Pilih grup sasaran. Untuk menambahkan grup target lain, pilih Tambahkan grup target, pilih grup target, tinjau bobot relatif, dan perbarui bobot sesuai kebutuhan. Anda harus mengaktifkan kelengkapan tingkat grup jika Anda mengaktifkan kekakuan pada salah satu grup target.

Jika Anda tidak memiliki grup target yang memenuhi kebutuhan Anda, pilih Buat grup target untuk membuatnya sekarang. Untuk informasi selengkapnya, lihat [Buat grup target](#).
  - Redirect ke URL - Masukkan URL dengan memasukkan setiap bagian secara terpisah pada tab bagian URI, atau dengan memasukkan alamat lengkap pada tab URL Lengkap. Untuk kode Status, pilih sementara (HTTP 302) atau permanen (HTTP 301) berdasarkan kebutuhan Anda.
  - Kembalikan respons tetap - Masukkan kode Respons untuk mengembalikan permintaan klien yang dijatuhkan. Secara opsional, Anda dapat menentukan jenis Konten dan badan Respons.
7. (Opsional) Untuk menambahkan tag, perluas tag Listener. Pilih Tambahkan tag baru dan masukkan kunci tag dan nilai tag.
8. Pilih Tambahkan pendengar.

## AWS CLI

Untuk membuat grup target

Jika Anda tidak memiliki grup target yang dapat Anda gunakan untuk tindakan default, gunakan [create-target-group](#) perintah untuk membuatnya sekarang. Sebagai contoh, lihat [Buat grup target](#).

Untuk membuat pendengar HTTP

Gunakan perintah [create-listener](#). Contoh berikut membuat pendengar HTTP dengan aturan default yang meneruskan lalu lintas ke grup target yang ditentukan.

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

Untuk membuat tindakan maju yang mendistribusikan lalu lintas antara dua kelompok target, gunakan `--default-actions` opsi berikut sebagai gantinya. Saat menentukan beberapa kelompok sasaran, Anda harus memberikan bobot untuk setiap kelompok sasaran.

```
--default-actions '[{  
  "Type":"forward",  
  "ForwardConfig":{  
    "TargetGroups":[  
      {"TargetGroupArn": "target-group-1-arn", "Weight":50},  
      {"TargetGroupArn": "target-group-2-arn", "Weight":50}  
    ]  
  }  
}]'
```

## CloudFormation

Untuk membuat pendengar HTTP

Tentukan sumber daya tipe [AWS::ElasticLoadBalancingV2::Listener](#). Contoh berikut membuat pendengar HTTP dengan aturan default yang meneruskan lalu lintas ke grup target yang ditentukan.

```
Resources:  
  myHTTPListener:  
    Type: 'AWS::ElasticLoadBalancingV2::Listener'  
    Properties:  
      LoadBalancerArn: !Ref myLoadBalancer  
      Protocol: HTTP  
      Port: 80  
      DefaultActions:  
        - Type: "forward"  
          TargetGroupArn: !Ref myTargetGroup
```

Untuk membuat tindakan maju yang mendistribusikan lalu lintas di antara beberapa grup target, gunakan `ForwardConfig` properti. Saat menentukan beberapa kelompok sasaran, Anda harus memberikan bobot untuk setiap kelompok sasaran.

```
Resources:
  myHTTPListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTP
      Port: 80
      DefaultActions:
        - Type: "forward"
          ForwardConfig:
            TargetGroups:
              - TargetGroupArn: !Ref TargetGroup1
                Weight: 50
              - TargetGroupArn: !Ref TargetGroup2
                Weight: 50
```

## Sertifikat SSL untuk Application Load Balancer

Saat Anda membuat pendengar yang aman untuk Application Load Balancer, Anda harus menerapkan setidaknya satu sertifikat pada penyeimbang beban. Penyeimbang beban memerlukan sertifikat X.509 (sertifikat server SSL/TLS). Sertifikat adalah bentuk digital identifikasi yang dikeluarkan oleh otoritas sertifikat (CA). Sertifikat berisi informasi identifikasi, masa berlaku, kunci publik, nomor seri, dan tanda tangan digital penerbit.

Ketika Anda membuat sertifikat untuk digunakan dengan penyeimbang beban Anda, Anda harus menentukan nama domain. Nama domain pada sertifikat harus cocok dengan catatan nama domain khusus sehingga kami dapat memverifikasi koneksi TLS. Jika mereka tidak cocok, lalu lintas tidak dienkripsi.

Anda harus menentukan nama domain yang sepenuhnya memenuhi syarat (FQDN) untuk sertifikat Anda, seperti `www.example.com` atau nama domain apex seperti `example.com`. Anda juga dapat menggunakan tanda bintang (\*) sebagai kartu liar untuk melindungi beberapa nama situs di domain yang sama. Saat Anda meminta sertifikat kartu liar, tanda bintang (\*) harus berada di posisi paling kiri dari nama domain dan hanya dapat melindungi satu tingkat subdomain. Misalnya, `*.example.com` melindungi `corp.example.com`, dan `images.example.com`, tetapi tidak dapat

melindungi `test.login.example.com`. Perhatikan juga bahwa `*.example.com` melindungi hanya subdomain dari `example.com`, itu tidak melindungi domain telanjang atau apex `example.com`. Nama kartu liar muncul di bidang Subjek dan di ekstensi Nama Alternatif Subjek sertifikat. Untuk informasi selengkapnya tentang sertifikat publik, lihat [Meminta sertifikat publik](#) di Panduan AWS Certificate Manager Pengguna.

Kami menyarankan Anda membuat sertifikat untuk penyeimbang beban menggunakan [AWS Certificate Manager \(ACM\)](#). ACM mendukung sertifikat RSA dengan panjang kunci 2048, 3072, dan 4096-bit, dan semua sertifikat ECDSA. ACM terintegrasi dengan Elastic Load Balancing sehingga Anda dapat men-deploy sertifikat pada penyeimbang beban Anda. Untuk informasi selengkapnya, lihat [AWS Certificate Manager Panduan Pengguna](#).

Atau, Anda dapat menggunakan SSL/TLS alat untuk membuat permintaan penandatanganan sertifikat (CSR), lalu mendapatkan CSR yang ditandatangani oleh CA untuk menghasilkan sertifikat, lalu mengimpor sertifikat ke ACM atau mengunggah sertifikat ke AWS Identity and Access Management (IAM). Untuk informasi lebih lanjut tentang mengimpor sertifikat ke ACM, lihat [Mengimpor sertifikat](#) ke AWS Certificate Manager Panduan Pengguna. Untuk informasi selengkapnya tentang mengunggah sertifikat ke IAM, lihat [Bekerja dengan sertifikat server](#) di Panduan Pengguna IAM.

## Sertifikat default

Ketika Anda membuat listener HTTPS, Anda harus menentukan tepat satu sertifikat. Sertifikat ini dikenal sebagai sertifikat default. Anda dapat mengganti sertifikat default setelah membuat listener HTTPS. Untuk informasi selengkapnya, lihat [Mengganti sertifikat default](#).

Jika Anda menentukan sertifikat tambahan di [daftar sertifikat](#), sertifikat default hanya digunakan jika klien tersambung tanpa menggunakan protokol Indikasi Nama Server (SNI) untuk menentukan nama host atau jika tidak ada sertifikat yang cocok dalam daftar sertifikat.

Jika Anda tidak menentukan sertifikat tambahan tetapi perlu menghosting beberapa aplikasi aman melalui penyeimbang beban tunggal, Anda dapat menggunakan sertifikat wildcard atau menambahkan Nama Alternatif Subjek (SAN) untuk setiap domain tambahan ke sertifikat Anda.

## Daftar sertifikat

Setelah membuat pendengar HTTPS, Anda dapat menambahkan sertifikat ke daftar sertifikat. Jika Anda membuat listener menggunakan Konsol Manajemen AWS, kami menambahkan sertifikat

default ke daftar sertifikat untuk Anda. Jika tidak, daftar sertifikat kosong. Menggunakan daftar sertifikat memungkinkan penyeimbang beban untuk mendukung beberapa domain pada port yang sama dan memberikan sertifikat yang berbeda untuk setiap domain. Untuk informasi selengkapnya, lihat [Menambahkan sertifikat ke daftar sertifikat](#).

Penyeimbang beban menggunakan algoritme pemilihan sertifikat cerdas dengan dukungan SNI. Jika nama host yang disediakan oleh klien cocok dengan satu sertifikat dalam daftar sertifikat, penyeimbang beban akan memilih sertifikat ini. Jika nama host yang disediakan oleh klien cocok dengan beberapa sertifikat dalam daftar sertifikat, penyeimbang beban memilih sertifikat terbaik yang dapat didukung klien. Pemilihan sertifikat didasarkan pada kriteria dalam urutan sebagai berikut:

- Algoritme kunci publik (lebih suka ECDSA daripada RSA)
- Kedaluwarsa (lebih suka tidak kedaluwarsa)
- Algoritma hashing (lebih suka SHA daripada MD5). Jika ada beberapa sertifikat SHA, pilih nomor SHA tertinggi.
- Panjang kunci (lebih memilih yang terbesar)
- Masa berlaku

Entri log akses penyeimbang beban menunjukkan nama host yang ditentukan oleh klien dan sertifikat yang diberikan kepada klien. Untuk informasi selengkapnya, lihat [Entri log akses](#).

## Perpanjangan sertifikat

Setiap sertifikat memiliki masa berlaku. Anda harus memastikan bahwa Anda memperpanjang atau mengganti setiap sertifikat untuk penyeimbang beban Anda sebelum masa berlakunya berakhir. Ini termasuk sertifikat default dan sertifikat dalam daftar sertifikat. Memperpanjang atau mengganti sertifikat tidak memengaruhi permintaan dalam penerbangan yang diterima oleh node penyeimbang beban dan sedang menunggu perutean ke target yang sehat. Setelah sertifikat diperpanjang, permintaan baru menggunakan akan menggunakan sertifikat yang telah diperpanjang. Setelah sertifikat diganti, permintaan baru akan menggunakan sertifikat baru.

Anda dapat mengelola perpanjangan sertifikat dan penggantian sebagai berikut:

- Sertifikat yang disediakan oleh AWS Certificate Manager dan digunakan pada penyeimbang beban Anda dapat diperbarui secara otomatis. ACM mencoba untuk memperpanjang sertifikat sebelum masa berlakunya habis. Untuk informasi lebih lanjut, lihat [Perpanjangan Terkelola](#) dalam AWS Certificate Manager Panduan Pengguna.

- Jika Anda mengimpor sertifikat ke ACM, Anda harus memantau tanggal kedaluwarsa sertifikat dan memperpanjang masa berlakunya sebelum kedaluwarsa. Untuk informasi lebih lanjut, lihat [Mengimpor sertifikat](#) di AWS Certificate Manager Panduan Pengguna.
- Jika Anda mengimpor sertifikat ke IAM, Anda harus membuat sertifikat baru, mengimpor sertifikat baru ke ACM atau IAM, menambahkan sertifikat baru ke penyeimbang beban Anda, dan menghapus sertifikat yang kedaluwarsa dari penyeimbang beban Anda.

## Kebijakan keamanan untuk Application Load Balancer

Elastic Load Balancing menggunakan konfigurasi negosiasi Lapisan Soket Aman (SSL), yang dikenal sebagai kebijakan keamanan, untuk menegosiasikan koneksi SSL antara klien dan penyeimbang beban. Kebijakan keamanan adalah kombinasi dari protokol dan sandi. Protokol membuat koneksi aman antara klien dan server dan memastikan bahwa semua data yang diteruskan antara klien dan penyeimbang beban Anda bersifat pribadi. Sandi adalah algoritme enkripsi yang menggunakan kunci enkripsi untuk membuat pesan kode. Protokol menggunakan beberapa sandi untuk mengenkripsi data melalui internet. Selama proses negosiasi koneksi, klien dan penyeimbang beban menyajikan daftar sandi dan protokol yang masing-masing mendukung, dalam urutan preferensi. Secara default, sandi pertama pada daftar server yang cocok salah satu sandi klien dipilih untuk sambungan aman.

### Pertimbangan-pertimbangan

- Pendengar HTTPS memerlukan kebijakan keamanan. Jika Anda tidak menentukan kebijakan keamanan saat membuat listener, kami menggunakan kebijakan keamanan default. Kebijakan keamanan default bergantung pada cara Anda membuat listener HTTPS:
  - Konsol — Kebijakan keamanan default adalah `ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09`.
  - Metode lain (misalnya, AWS CLI, AWS CloudFormation, dan AWS CDK) — Kebijakan keamanan default adalah `ELBSecurityPolicy-2016-08`.
- Untuk melihat versi protokol TLS (posisi bidang log 5) dan pertukaran kunci (posisi bidang log 13) untuk permintaan koneksi ke penyeimbang beban Anda, aktifkan pencatatan koneksi dan periksa entri log yang sesuai. Untuk informasi selengkapnya, lihat [Log koneksi](#).
- Kebijakan keamanan dengan PQ dalam nama mereka menawarkan pertukaran kunci pasca-kuantum hibrida. Untuk kompatibilitas, mereka mendukung algoritma pertukaran kunci ML-KEM klasik dan pasca-kuantum. Klien harus mendukung pertukaran kunci ML-KEM untuk menggunakan TLS pasca-kuantum hibrida untuk pertukaran kunci. Kebijakan pasca-kuantum

hibrida mendukung algoritma Secp256R1, Secp384r1 dan MLKEM768 X25519. MLKEM1024 MLKEM768 Untuk informasi lebih lanjut, lihat [Post-Quantum Cryptography](#).

- AWS merekomendasikan penerapan kebijakan keamanan berbasis TLS pasca-kuantum (PQ-TLS) baru atau. `ELBSecurityPolicy-TLS13-1-2-Res-PQ-2025-09` `ELBSecurityPolicy-TLS13-1-2-FIPS-PQ-2025-09` Kebijakan ini memastikan kompatibilitas mundur dengan mendukung klien yang mampu menegosiasikan hybrid PQ-TLS, TLS 1.3 saja, atau TLS 1.2 saja, sehingga meminimalkan gangguan layanan selama transisi ke kriptografi pasca-kuantum. Anda dapat bermigrasi secara progresif ke kebijakan keamanan yang lebih ketat saat aplikasi klien Anda mengembangkan kemampuan untuk menegosiasikan PQ-TLS untuk operasi pertukaran kunci.
- Untuk memenuhi standar kepatuhan dan keamanan yang mengharuskan menonaktifkan versi protokol TLS tertentu, atau untuk mendukung klien lama yang membutuhkan cipher usang, Anda dapat menggunakan salah satu kebijakan keamanan. `ELBSecurityPolicy-TLS-` Untuk melihat versi protokol TLS untuk permintaan ke Application Load Balancer Anda, aktifkan pencatatan akses untuk penyeimbang beban Anda dan periksa entri log akses yang sesuai. Untuk informasi selengkapnya, lihat [Log akses](#).
- Anda dapat membatasi kebijakan keamanan yang tersedia untuk pengguna di seluruh Akun AWS dan AWS Organizations dengan menggunakan kunci [kondisi Elastic Load Balancing](#) di IAM dan kebijakan kontrol layanan SCPs (), masing-masing. Untuk informasi selengkapnya, lihat [Kebijakan kontrol layanan \(SCPs\)](#) di Panduan AWS Organizations Pengguna.
- Kebijakan yang hanya mendukung TLS 1.3 mendukung Forward Secrecy (FS). Kebijakan yang mendukung TLS 1.3 dan TLS 1.2 yang hanya memiliki cipher dari bentuk `TLS_*` dan `ECDHE_*` juga menyediakan FS.
- Aplikasi Load Balancer mendukung dimulainya kembali TLS menggunakan PSK (TLS 1.3) dan IDs/session Tiket sesi (TLS 1.2 dan yang lebih lama). Resume hanya didukung dalam koneksi ke alamat IP Application Load Balancer yang sama. Fitur Data 0-RTT dan ekstensi `early_data` tidak diimplementasikan.
- Application Load Balancer tidak mendukung kebijakan keamanan kustom.
- Application Load Balancers mendukung renegotiasi SSL hanya untuk koneksi target.

## Kompatibilitas

- Semua pendengar aman yang terpasang pada penyeimbang beban yang sama harus menggunakan kebijakan keamanan yang kompatibel. Untuk memigrasikan semua pendengar aman untuk penyeimbang beban ke kebijakan keamanan yang tidak kompatibel dengan kebijakan

yang saat ini digunakan, hapus semua kecuali satu pendengar aman, ubah kebijakan keamanan pendengar aman, lalu buat pendengar aman tambahan.

- Kebijakan TLS pasca-kuantum FIPS dan kebijakan FIPS - Kompatibel
- Kebijakan TLS pasca-kuantum dan kebijakan TLS pasca-kuantum FIPS atau FIPS - Kompatibel
- Kebijakan TLS (non-FIPS, non-post-quantum) dan kebijakan TLS pasca-kuantum FIPS atau FIPS - Tidak Kompatibel
- Kebijakan TLS (non-FIPS, non-post-quantum) dan kebijakan TLS pasca-kuantum - Tidak Kompatibel

### Koneksi backend

- Anda dapat memilih kebijakan keamanan yang digunakan untuk koneksi front-end, tetapi tidak koneksi backend. Kebijakan keamanan untuk koneksi backend bergantung pada kebijakan keamanan pendengar. Jika ada pendengar Anda yang menggunakan:
  - Kebijakan TLS pasca-kuantum FIPS - Koneksi backend digunakan `ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09`
  - Kebijakan FIPS - Koneksi backend digunakan `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04`
  - Kebijakan TLS pasca-kuantum - Koneksi backend digunakan `ELBSecurityPolicy-TLS13-1-0-PQ-2025-09`
  - Kebijakan TLS 1.3 - Koneksi backend digunakan `ELBSecurityPolicy-TLS13-1-0-2021-06`
  - Kebijakan TLS lainnya - Koneksi backend digunakan `ELBSecurityPolicy-2016-08`

### Kebijakan Keamanan

- [Contoh describe-ssl-policies perintah](#)
- [Kebijakan keamanan TLS](#)
  - [Protokol berdasarkan kebijakan](#)
  - [Cipher berdasarkan kebijakan](#)
  - [Kebijakan oleh cipher](#)
- [Kebijakan keamanan FIPS](#)
  - [Protokol berdasarkan kebijakan](#)
  - [Cipher berdasarkan kebijakan](#)

- [Kebijakan oleh cipher](#)
- [Kebijakan yang didukung FS](#)
- [Protokol berdasarkan kebijakan](#)
- [Cipher berdasarkan kebijakan](#)
- [Kebijakan oleh cipher](#)

## Contoh describe-ssl-policies perintah

Anda dapat menjelaskan protokol dan cipher untuk kebijakan keamanan, atau menemukan kebijakan yang memenuhi kebutuhan Anda, menggunakan perintah. [describe-ssl-policies](#) AWS CLI

Contoh berikut menjelaskan kebijakan yang ditentukan.

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

Contoh berikut mencantumkan kebijakan dengan string yang ditentukan dalam nama kebijakan.

```
aws elbv2 describe-ssl-policies \  
  --query "SslPolicies[?contains(Name, 'FIPS')].Name"
```

Contoh berikut mencantumkan kebijakan yang mendukung protokol yang ditentukan.

```
aws elbv2 describe-ssl-policies \  
  --query "SslPolicies[?contains(SslProtocols, 'TLSv1.3')].Name"
```

Contoh berikut mencantumkan kebijakan yang mendukung cipher yang ditentukan.

```
aws elbv2 describe-ssl-policies \  
  --query "SslPolicies[?Ciphers[?contains(Name, 'TLS_AES_128_GCM_SHA256')]].Name"
```

Contoh berikut mencantumkan kebijakan yang tidak mendukung cipher yang ditentukan.

```
aws elbv2 describe-ssl-policies \  
  --query 'SslPolicies[?length(Ciphers[?starts_with(Name, `AES128-GCM-SHA256`))] ==  
  `0`].Name'
```

## Kebijakan keamanan TLS

Anda dapat menggunakan kebijakan keamanan TLS untuk memenuhi standar kepatuhan dan keamanan yang mengharuskan menonaktifkan versi protokol TLS tertentu, atau untuk mendukung klien lama yang memerlukan cipher usang.

Kebijakan yang hanya mendukung TLS 1.3 mendukung Forward Secrecy (FS). Kebijakan yang mendukung TLS 1.3 dan TLS 1.2 yang hanya memiliki cipher dari bentuk TLS\_\* dan ECDHE\_\* juga menyediakan FS.

### Daftar Isi

- [Protokol berdasarkan kebijakan](#)
- [Cipher berdasarkan kebijakan](#)
- [Kebijakan oleh cipher](#)

### Protokol berdasarkan kebijakan

Tabel berikut menjelaskan protokol yang didukung oleh setiap kebijakan keamanan TLS.

Kebijakan Keamanan	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityKebijakan- TLS13 -1-3-2021-06	Ya	Tidak	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-3-PQ-2025-09	Ya	Tidak	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-2021-06	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-Re-2021-06	Ya	Ya	Tidak	Tidak

Kebijakan Keamanan	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityKebijakan- TLS13 -1-2-RES-PQ-2025-09	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-1-2021-06	Ya	Ya	Ya	Tidak
ELBSecurityKebijakan- TLS13 -1-0-2021-06	Ya	Ya	Ya	Ya
ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09	Ya	Ya	Ya	Ya
ELBSecurityKebijakan-TLS-1-2-EXT-2018-06	Tidak	Ya	Tidak	Tidak
ELBSecurityKebijakan-TLS-1-2-2017-01	Tidak	Ya	Tidak	Tidak
ELBSecurityKebijakan-TLS-1-1-2017-01	Tidak	Ya	Ya	Tidak
ELBSecurityKebijakan-2016-08	Tidak	Ya	Ya	Ya

## Cipher berdasarkan kebijakan

Tabel berikut menjelaskan cipher yang didukung oleh setiap kebijakan keamanan TLS.

Kebijakan keamanan	Cipher
ELBSecurityKebijakan- TLS13 -1-3-2021-06	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> <li>• TLS_ _ _ CHACHA20 POLY1305 SHA256</li> </ul>
ELBSecurityKebijakan- TLS13 -1-3-PQ-2 025-09	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> <li>• TLS_ _ _ CHACHA20 POLY1305 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-2-2021-06	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> <li>• TLS_ _ _ CHACHA20 POLY1305 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-2-PQ-2 025-09	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> <li>• TLS_ _ _ CHACHA20 POLY1305 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-2-Re-2021-06	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> <li>• TLS_ _ _ CHACHA20 POLY1305 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-2-RES- PQ-2025-09	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> <li>• TLS_ _ _ CHACHA20 POLY1305 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-2-Ext2 -2021-06	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> <li>• TLS_ _ _ CHACHA20 POLY1305 SHA256</li> </ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09	<ul style="list-style-type: none"> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -SHA AES128</li> <li>• ECDHE-RSA- -SHA AES128</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> <li>• ECDHE-ECDSA- -SHA AES256</li> <li>• ECDHE-RSA- -SHA AES256</li> <li>• AES128-GCM- SHA256</li> <li>• AES128-SHA256</li> <li>• AES128-SHA</li> <li>• AES256-GCM- SHA384</li> <li>• AES256-SHA256</li> <li>• AES256-SHA</li> </ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan- TLS13 -1-2-Ext1 -2021-06	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09	<ul style="list-style-type: none"> <li>• TLS_ _ _ CHACHA20 POLY1305 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> <li>• AES128-GCM- SHA256</li> <li>• AES128-SHA256</li> <li>• AES256-GCM- SHA384</li> <li>• AES256-SHA256</li> </ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan- TLS13 -1-1-2021-06	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> <li>• TLS_ _ _ CHACHA20 POLY1305 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -SHA AES128</li> <li>• ECDHE-RSA- -SHA AES128</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> <li>• ECDHE-ECDSA- -SHA AES256</li> <li>• ECDHE-RSA- -SHA AES256</li> <li>• AES128-GCM- SHA256</li> <li>• AES128-SHA256</li> <li>• AES128-SHA</li> <li>• AES256-GCM- SHA384</li> <li>• AES256-SHA256</li> <li>• AES256-SHA</li> </ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan- TLS13 -1-0-2021-06	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> </ul>
ELBSecurityKebijakan- TLS13 -1-0-PQ-2 025-09	<ul style="list-style-type: none"> <li>• TLS_AES_256_GCM_SHA384</li> <li>• TLS_..._CHACHA20_POLY1305_SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES128_SHA256</li> <li>• ECDHE-RSA- -GCM- AES128_SHA256</li> <li>• ECDHE-ECDSA- - AES128_SHA256</li> <li>• ECDHE-RSA- - AES128_SHA256</li> <li>• ECDHE-ECDSA- -SHA AES128</li> <li>• ECDHE-RSA- -SHA AES128</li> <li>• ECDHE-ECDSA- -GCM- AES256_SHA384</li> <li>• ECDHE-RSA- -GCM- AES256_SHA384</li> <li>• ECDHE-ECDSA- - AES256_SHA384</li> <li>• ECDHE-RSA- - AES256_SHA384</li> <li>• ECDHE-ECDSA- -SHA AES256</li> <li>• ECDHE-RSA- -SHA AES256</li> <li>• AES128-GCM- SHA256</li> <li>• AES128-SHA256</li> <li>• AES128-SHA</li> <li>• AES256-GCM- SHA384</li> <li>• AES256-SHA256</li> <li>• AES256-SHA</li> </ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan-TLS-1-2-EXT-2018-06	<ul style="list-style-type: none"><li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li><li>• ECDHE-RSA- -GCM- AES128 SHA256</li><li>• ECDHE-ECDSA- - AES128 SHA256</li><li>• ECDHE-RSA- - AES128 SHA256</li><li>• ECDHE-ECDSA- -SHA AES128</li><li>• ECDHE-RSA- -SHA AES128</li><li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li><li>• ECDHE-RSA- -GCM- AES256 SHA384</li><li>• ECDHE-ECDSA- - AES256 SHA384</li><li>• ECDHE-RSA- - AES256 SHA384</li><li>• ECDHE-ECDSA- -SHA AES256</li><li>• ECDHE-RSA- -SHA AES256</li><li>• AES128-GCM- SHA256</li><li>• AES128-SHA256</li><li>• AES128-SHA</li><li>• AES256-GCM- SHA384</li><li>• AES256-SHA256</li><li>• AES256-SHA</li></ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan-TLS-1-2-2017-01	<ul style="list-style-type: none"><li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li><li>• ECDHE-RSA- -GCM- AES128 SHA256</li><li>• ECDHE-ECDSA- - AES128 SHA256</li><li>• ECDHE-RSA- - AES128 SHA256</li><li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li><li>• ECDHE-RSA- -GCM- AES256 SHA384</li><li>• ECDHE-ECDSA- - AES256 SHA384</li><li>• ECDHE-RSA- - AES256 SHA384</li><li>• AES128-GCM- SHA256</li><li>• AES128-SHA256</li><li>• AES256-GCM- SHA384</li><li>• AES256-SHA256</li></ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan-TLS-1-1-2017-01	<ul style="list-style-type: none"><li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li><li>• ECDHE-RSA- -GCM- AES128 SHA256</li><li>• ECDHE-ECDSA- - AES128 SHA256</li><li>• ECDHE-RSA- - AES128 SHA256</li><li>• ECDHE-ECDSA- -SHA AES128</li><li>• ECDHE-RSA- -SHA AES128</li><li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li><li>• ECDHE-RSA- -GCM- AES256 SHA384</li><li>• ECDHE-ECDSA- - AES256 SHA384</li><li>• ECDHE-RSA- - AES256 SHA384</li><li>• ECDHE-ECDSA- -SHA AES256</li><li>• ECDHE-RSA- -SHA AES256</li><li>• AES128-GCM- SHA256</li><li>• AES128-SHA256</li><li>• AES128-SHA</li><li>• AES256-GCM- SHA384</li><li>• AES256-SHA256</li><li>• AES256-SHA</li></ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan-2016-08	<ul style="list-style-type: none"> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -SHA AES128</li> <li>• ECDHE-RSA- -SHA AES128</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> <li>• ECDHE-ECDSA- -SHA AES256</li> <li>• ECDHE-RSA- -SHA AES256</li> <li>• AES128-GCM- SHA256</li> <li>• AES128-SHA256</li> <li>• AES128-SHA</li> <li>• AES256-GCM- SHA384</li> <li>• AES256-SHA256</li> <li>• AES256-SHA</li> </ul>

## Kebijakan oleh cipher

Tabel berikut menjelaskan kebijakan keamanan TLS yang mendukung setiap cipher.

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — TLS_AES_128_GCM_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-3-2021-06</li> </ul>	1301
IANA — TLS_AES_128_GCM_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-3-PQ-2025-09</li> </ul>	

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Re-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> </ul>	

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — TLS_AES_256_GCM_SHA384  IANA — TLS_AES_256_GCM_SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-3-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-3-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Re-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> </ul>	1302

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL - TLS___CHACHA20 POLY1305 SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-3-2021-06</li> </ul>	1303
IANA - TLS___CHACHA20 POLY1305 SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-3-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Re-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> </ul>	

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
<p>ECDHE-ECDSA-AESOpenSSL - 128-GCM- SHA256</p> <p>IANA — TLS_ECDHE_ECDSA_DE NGAN_AES_128_GCM_ SHA256</p>	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Re-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	c02b

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-RSA-AESOpenSSL - 128-GCM- SHA256  IANA — TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Re-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	c02f

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-ECDSA-AESOpenSSL - 128-SHA256  IANA — TLS_ECDHE_ECDSA_DENGAN_AES_128_CBC_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	c023

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-RSA-AESOpenSSL - 128-SHA256  IANA — TLS_ECDHE_RSA_DENG AN_AES_128_CBC_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	c027

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — ECDHE-ECDSA-AES 128-SHA  IANA — TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	c009

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — ECDHE-RSA-AES 128-SHA  IANA — TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	c013

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-ECDSA-AESOpenSSL — 256-GCM- SHA384  IANA — TLS_ECDHE_ECDSA_DE NGAN_AES_256_GCM_ SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Re-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	c02c

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-RSA-AESOpenSSL — 256-GCM- SHA384  IANA — TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Re-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	c030

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-ECDSA-AESOpenSSL — 256-SHA384  IANA — TLS_ECDHE_ECDSA_DENGAN_AES_256_CBC_SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	c024

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-RSA-AESOpenSSL — 256-SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-2021-06</li> </ul>	c028
IANA — TLS_ECDHE_RSA_DENG AN_AES_256_CBC_SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — ECDHE-ECDSA-AES 256-SHA  IANA — TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	c00a

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — ECDHE-RSA-AES 256-SHA  IANA — TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	c014

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
AES128OpenSSL — -GCM- SHA256  IANA — TLS_RSA_WITH_AES_128_GCM_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	9c

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
AES128OpenSSL — - SHA256  IANA — TLS_RSA_DENGAN_AES_128_CBC_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	3c

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — AES128 -SHA  IANA — TLS_RSA_WITH_AES_128_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	2f

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
AES256OpenSSL — -GCM- SHA384  IANA — TLS_RSA_WITH_AES_256_GCM_SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	9d

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
AES256OpenSSL — - SHA256  IANA — TLS_RSA_DENGAN_AES_256_CBC_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-2-2017-01</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	3d

Nama sandi	Kebijakan Keamanan	Rangkaian Penyediaan
OpenSSL — AES256 -SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-2021-06</li> </ul>	35
IANA — TLS_RSA_WITH_AES_256_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-Ext2-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-2021-06</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-PQ-2025-09</li> <li>• ELBSecurityKebijakan-TLS-1-2-EXT-2018-06</li> <li>• ELBSecurityKebijakan-TLS-1-1-2017-01</li> <li>• ELBSecurityKebijakan-2016-08</li> </ul>	

## Kebijakan keamanan FIPS

Federal Information Processing Standard (FIPS) adalah standar pemerintah AS dan Kanada yang menetapkan persyaratan keamanan untuk modul kriptografi yang melindungi informasi sensitif. Untuk mempelajari lebih lanjut, lihat [Federal Information Processing Standard \(FIPS\) 140](#) di halaman Kepatuhan Keamanan AWS Cloud.

Semua kebijakan FIPS memanfaatkan modul kriptografi yang divalidasi AWS-LC FIPS. Untuk mempelajari lebih lanjut, lihat halaman [Modul Kriptografi AWS-LC di situs Program Validasi Modul Kriptografi NIST](#).

### Important

Kebijakan ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04 dan ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04 disediakan hanya untuk kompatibilitas

lama. Meskipun mereka menggunakan kriptografi FIPS menggunakan FIPS140 modul, mereka mungkin tidak sesuai dengan panduan NIST terbaru untuk konfigurasi TLS.

## Daftar Isi

- [Protokol berdasarkan kebijakan](#)
- [Cipher berdasarkan kebijakan](#)
- [Kebijakan oleh cipher](#)

## Protokol berdasarkan kebijakan

Tabel berikut menjelaskan protokol yang didukung oleh setiap kebijakan keamanan FIPS.

Kebijakan Keamanan	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityKebijakan- TLS13 -1-3-FIPS-2023-04	Ya	Tidak	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-3-FIPS-PQ-2025-09	Ya	Tidak	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-2023-04	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-PQ-2025-09	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04	Ya	Ya	Tidak	Tidak

Kebijakan Keamanan	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09	Ya	Ya	Tidak	Tidak
ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04	Ya	Ya	Ya	Tidak
ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04	Ya	Ya	Ya	Ya
ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09	Ya	Ya	Ya	Ya

## Cipher berdasarkan kebijakan

Tabel berikut menjelaskan cipher yang didukung oleh setiap kebijakan keamanan FIPS.

Kebijakan keamanan	Cipher
ELBSecurityKebijakan- TLS13 -1-3-FIPS-2023-04	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-3-FIPS-PQ-2025-09	

Kebijakan keamanan	Cipher
ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04  ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-2023-04  ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-PQ-2025-09	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> </ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09	<ul style="list-style-type: none"> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -SHA AES128</li> <li>• ECDHE-RSA- -SHA AES128</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> <li>• ECDHE-RSA- -SHA AES256</li> <li>• ECDHE-ECDSA- -SHA AES256</li> <li>• AES128-GCM- SHA256</li> <li>• AES128-SHA256</li> <li>• AES128-SHA</li> <li>• AES256-GCM- SHA384</li> <li>• AES256-SHA256</li> <li>• AES256-SHA</li> </ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09	<ul style="list-style-type: none"> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> <li>• AES128-GCM- SHA256</li> <li>• AES128-SHA256</li> <li>• AES256-GCM- SHA384</li> <li>• AES256-SHA256</li> </ul>
ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09	<ul style="list-style-type: none"> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -SHA AES128</li> <li>• ECDHE-RSA- -SHA AES128</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> <li>• ECDHE-RSA- -SHA AES256</li> <li>• ECDHE-ECDSA- -SHA AES256</li> </ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan- TLS13 -1-1-FIPS -2023-04	<ul style="list-style-type: none"><li>• TLS_AES_128_GCM_SHA256</li><li>• TLS_AES_256_GCM_SHA384</li><li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li><li>• ECDHE-RSA- -GCM- AES128 SHA256</li><li>• ECDHE-ECDSA- - AES128 SHA256</li><li>• ECDHE-RSA- - AES128 SHA256</li><li>• ECDHE-ECDSA- -SHA AES128</li><li>• ECDHE-RSA- -SHA AES128</li><li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li><li>• ECDHE-RSA- -GCM- AES256 SHA384</li><li>• ECDHE-ECDSA- - AES256 SHA384</li><li>• ECDHE-RSA- - AES256 SHA384</li><li>• ECDHE-RSA- -SHA AES256</li><li>• ECDHE-ECDSA- -SHA AES256</li><li>• AES128-GCM- SHA256</li><li>• AES128-SHA256</li><li>• AES128-SHA</li><li>• AES256-GCM- SHA384</li><li>• AES256-SHA256</li><li>• AES256-SHA</li></ul>

Kebijakan keamanan	Cipher
ELBSecurityKebijakan- TLS13 -1-0-FIPS -2023-04	<ul style="list-style-type: none"> <li>• TLS_AES_128_GCM_SHA256</li> <li>• TLS_AES_256_GCM_SHA384</li> </ul>
ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09	<ul style="list-style-type: none"> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -SHA AES128</li> <li>• ECDHE-RSA- -SHA AES128</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> <li>• ECDHE-RSA- -SHA AES256</li> <li>• ECDHE-ECDSA- -SHA AES256</li> <li>• AES128-GCM- SHA256</li> <li>• AES128-SHA256</li> <li>• AES128-SHA</li> <li>• AES256-GCM- SHA384</li> <li>• AES256-SHA256</li> <li>• AES256-SHA</li> </ul>

## Kebijakan oleh cipher

Tabel berikut menjelaskan kebijakan keamanan FIPS yang mendukung setiap cipher.

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — TLS_AES_128_GCM_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-3-FIPS-2023-04</li> </ul>	1301
IANA — TLS_AES_128_GCM_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-3-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> </ul>	

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
	<ul style="list-style-type: none"><li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li></ul>	

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — TLS_AES_256_GCM_SHA384  IANA — TLS_AES_256_GCM_SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-3-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-3-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> </ul>	1302

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	
ECDHE-ECDSA-AESOpenSSL - 128-GCM- SHA256  IANA — TLS_ECDHE_ECDSA_DE NGAN_AES_128_GCM_ SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	c02b

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-RSA-AESOpenSSL - 128-GCM- SHA256  IANA — TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	c02f

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-ECDSA-AESOpenSSL - 128-SHA256  IANA — TLS_ECDHE_ECDSA_DE NGAN_AES_128_CBC_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	c023

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
<p>ECDHE-RSA-AESOpenSSL - 128-SHA256</p> <p>IANA — TLS_ECDHE_RSA_DESIGN_AES_128_CBC_SHA256</p>	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	c027

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — ECDHE-ECDSA-AES 128-SHA  IANA — TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	c009
OpenSSL — ECDHE-RSA-AES 128-SHA  IANA — TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	c013

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-ECDSA-AESOpenSSL — 256-GCM- SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-2023-04</li> </ul>	c02c
IANA — TLS_ECDHE_ECDSA_DE NGAN_AES_256_GCM_ SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-RSA-AESOpenSSL — 256-GCM- SHA384  IANA — TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-RES-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	c030

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-ECDSA-AESOpenSSL — 256-SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04</li> </ul>	c024
IANA — TLS_ECDHE_ECDSA_DE NGAN_AES_256_CBC_SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-RSA-AESOpenSSL — 256-SHA384  IANA — TLS_ECDHE_RSA_DESIGN_AES_256_CBC_SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	c028

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — ECDHE-ECDSA-AES 256-SHA  IANA — TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	c00a
OpenSSL — ECDHE-RSA-AES 256-SHA  IANA — TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT0-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	c014

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
AES128OpenSSL — -GCM- SHA256  IANA — TLS_RSA_WITH_AES_128_GCM_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	9c
AES128OpenSSL — - SHA256  IANA — TLS_RSA_DENGAN_AES_128_CBC_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	3c

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — AES128 -SHA  IANA — TLS_RSA_WITH_AES_128_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	2f
AES256OpenSSL — -GCM- SHA384  IANA — TLS_RSA_WITH_AES_256_GCM_SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	9d

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
AES256OpenSSL — - SHA256  IANA — TLS_RSA_DENGAN_AES_256_CBC_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT1-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	3d
OpenSSL — AES256 -SHA  IANA — TLS_RSA_WITH_AES_256_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-2-EXT2-FIPS-PQ-2025-09</li> <li>• ELBSecurityKebijakan- TLS13 -1-1-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-2023-04</li> <li>• ELBSecurityKebijakan- TLS13 -1-0-FIPS-PQ-2025-09</li> </ul>	35

## Kebijakan yang didukung FS

Kebijakan keamanan yang didukung FS (Forward Secrecy) memberikan perlindungan tambahan terhadap penyadapan data terenkripsi, melalui penggunaan kunci sesi acak yang unik. Ini mencegah decoding data yang diambil, bahkan jika kunci rahasia jangka panjang dikompromikan.

Kebijakan di bagian ini mendukung FS, dan “FS” disertakan dalam nama mereka. Namun, ini bukan satu-satunya kebijakan yang mendukung FS. Kebijakan yang hanya mendukung TLS 1.3 mendukung FS. Kebijakan yang mendukung TLS 1.3 dan TLS 1.2 yang hanya memiliki cipher dari bentuk TLS\_\* dan ECDHE\_\* juga menyediakan FS.

### Daftar Isi

- [Protokol berdasarkan kebijakan](#)
- [Cipher berdasarkan kebijakan](#)
- [Kebijakan oleh cipher](#)

### Protokol berdasarkan kebijakan

Tabel berikut menjelaskan protokol yang didukung oleh setiap kebijakan keamanan FS yang didukung.

Kebijakan Keamanan	TLS 1.3	TLS 1.2	TLS 1.1	TLS 1.0
ELBSecurityKebijakan-FS-1-2-RES-2020-10	Tidak	Ya	Tidak	Tidak
ELBSecurityKebijakan-FS-1-2-RES-2019-08	Tidak	Ya	Tidak	Tidak
ELBSecurityKebijakan-FS-1-2-2019-08	Tidak	Ya	Tidak	Tidak
ELBSecurityKebijakan-FS-1-1-2019-08	Tidak	Ya	Ya	Tidak
ELBSecurityKebijakan-FS-2018-06	Tidak	Ya	Ya	Ya

## Cipher berdasarkan kebijakan

Tabel berikut menjelaskan sandi yang didukung oleh setiap kebijakan keamanan yang didukung FS.

Kebijakan keamanan	Cipher
ELBSecurityKebijakan-FS-1-2-RES-2020-10	<ul style="list-style-type: none"> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> </ul>
ELBSecurityKebijakan-FS-1-2-RES-2019-08	<ul style="list-style-type: none"> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> </ul>
ELBSecurityKebijakan-FS-1-2-2019-08	<ul style="list-style-type: none"> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -SHA AES128</li> <li>• ECDHE-RSA- -SHA AES128</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> <li>• ECDHE-RSA- -SHA AES256</li> <li>• ECDHE-ECDSA- -SHA AES256</li> </ul>
ELBSecurityKebijakan-FS-1-1-2019-08	<ul style="list-style-type: none"> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> </ul>

Kebijakan keamanan	Cipher
	<ul style="list-style-type: none"> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -SHA AES128</li> <li>• ECDHE-RSA- -SHA AES128</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> <li>• ECDHE-RSA- -SHA AES256</li> <li>• ECDHE-ECDSA- -SHA AES256</li> </ul>
ELBSecurityKebijakan-FS-2018-06	<ul style="list-style-type: none"> <li>• ECDHE-ECDSA- -GCM- AES128 SHA256</li> <li>• ECDHE-RSA- -GCM- AES128 SHA256</li> <li>• ECDHE-ECDSA- - AES128 SHA256</li> <li>• ECDHE-RSA- - AES128 SHA256</li> <li>• ECDHE-ECDSA- -SHA AES128</li> <li>• ECDHE-RSA- -SHA AES128</li> <li>• ECDHE-ECDSA- -GCM- AES256 SHA384</li> <li>• ECDHE-RSA- -GCM- AES256 SHA384</li> <li>• ECDHE-ECDSA- - AES256 SHA384</li> <li>• ECDHE-RSA- - AES256 SHA384</li> <li>• ECDHE-RSA- -SHA AES256</li> <li>• ECDHE-ECDSA- -SHA AES256</li> </ul>

## Kebijakan oleh cipher

Tabel berikut menjelaskan kebijakan keamanan yang didukung FS yang mendukung setiap cipher.

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
ECDHE-ECDSA-AESOpenSSL - 128-GCM- SHA256  IANA — TLS_ECDHE_ECDSA_DE NGAN_AES_128_GCM_ SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-RES-2020-10</li> <li>• ELBSecurityKebijakan-FS-1-2-RES-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-2-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>• ELBSecurityKebijakan-FS-2018-06</li> </ul>	c02b
ECDHE-RSA-AESOpenSSL - 128-GCM- SHA256  IANA — TLS_ECDHE_RSA_WITH_AES_128_GCM_ SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-RES-2020-10</li> <li>• ELBSecurityKebijakan-FS-1-2-RES-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-2-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>• ELBSecurityKebijakan-FS-2018-06</li> </ul>	c02f
ECDHE-ECDSA-AESOpenSSL - 128-SHA256  IANA — TLS_ECDHE_ECDSA_DE NGAN_AES_128_CBC_ SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-RES-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-2-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>• ELBSecurityKebijakan-FS-2018-06</li> </ul>	c023
ECDHE-RSA-AESOpenSSL - 128-SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-RES-2019-08</li> </ul>	c027

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
IANA — TLS_ECDHE_RSA_DENG AN_AES_128_CBC_SHA256	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>• ELBSecurityKebijakan-FS-2018-06</li> </ul>	
OpenSSL — ECDHE-ECDSA-AES 128-SHA  IANA — TLS_ECDHE_ECDSA_WI TH_AES_128_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>• ELBSecurityKebijakan-FS-2018-06</li> </ul>	c009
OpenSSL — ECDHE-RSA-AES 128-SHA  IANA — TLS_ECDHE_RSA_WITH _AES_128_CBC_SHA	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>• ELBSecurityKebijakan-FS-2018-06</li> </ul>	c013
ECDHE-ECDSA-AESOpenSSL — 256-GCM- SHA384  IANA — TLS_ECDHE_ECDSA_DE NGAN_AES_256_GCM_SHA384	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-RES-2020-10</li> <li>• ELBSecurityKebijakan-FS-1-2-RES-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-2-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>• ELBSecurityKebijakan-FS-2018-06</li> </ul>	c02c

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
<p>ECDHE-RSA-AESOpenSSL — 256-GCM- SHA384</p> <p>IANA — TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384</p>	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-RES-2020-10</li> <li>• ELBSecurityKebijakan-FS-1-2-RES-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-2-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>• ELBSecurityKebijakan-FS-2018-06</li> </ul>	c030
<p>ECDHE-ECDSA-AESOpenSSL — 256-SHA384</p> <p>IANA — TLS_ECDHE_ECDSA_DENGAN_AES_256_CBC_SHA384</p>	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-RES-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-2-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>• ELBSecurityKebijakan-FS-2018-06</li> </ul>	c024
<p>ECDHE-RSA-AESOpenSSL — 256-SHA384</p> <p>IANA — TLS_ECDHE_RSA_DENGAN_AES_256_CBC_SHA384</p>	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-RES-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-2-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>• ELBSecurityKebijakan-FS-2018-06</li> </ul>	c028
<p>OpenSSL — ECDHE-ECDSA-AES 256-SHA</p> <p>IANA — TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA</p>	<ul style="list-style-type: none"> <li>• ELBSecurityKebijakan-FS-1-2-2019-08</li> <li>• ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>• ELBSecurityKebijakan-FS-2018-06</li> </ul>	c00a

Nama sandi	Kebijakan Keamanan	Rangkaian Penyandian
OpenSSL — ECDHE-RSA-AES 256-SHA	<ul style="list-style-type: none"> <li>ELBSecurityKebijakan-FS-1-2-2019-08</li> </ul>	c014
IANA — TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	<ul style="list-style-type: none"> <li>ELBSecurityKebijakan-FS-1-1-2019-08</li> <li>ELBSecurityKebijakan-FS-2018-06</li> </ul>	

## Buat listener HTTPS untuk Application Load Balancer Anda

Pendengar memeriksa permintaan koneksi. Anda menentukan listener saat membuat penyeimbang beban, dan Anda dapat menambahkan listener ke penyeimbang beban kapan Anda saja.

Untuk membuat pendengar HTTPS, Anda harus menerapkan setidaknya satu [sertifikat server SSL](#) pada penyeimbang beban Anda. Penyeimbang beban menggunakan sertifikat server untuk mengakhiri koneksi front-end dan kemudian mendekripsi permintaan dari klien sebelum mengirimkannya ke target. Anda juga harus menentukan [kebijakan keamanan](#), yang digunakan untuk menegosiasikan koneksi aman antara klien dan penyeimbang beban.

Jika Anda perlu meneruskan lalu lintas terenkripsi ke target tanpa penyeimbang beban mendekripsi, Anda dapat membuat Network Load Balancer atau Classic Load Balancer dengan pendengar TCP di port 443. Dengan pendengar TCP, penyeimbang beban meneruskan lalu lintas terenkripsi ke target tanpa mendekripsi.

Informasi di halaman ini membantu Anda membuat listener HTTPS untuk penyeimbang beban Anda. Untuk menambahkan listener HTTP ke penyeimbang beban Anda, lihat [Membuat listener HTTP untuk Application Load Balancer Anda](#).

### Prasyarat

- Untuk menambahkan tindakan maju ke peraturan listener default, Anda harus menentukan grup target yang tersedia. Untuk informasi selengkapnya, lihat [Buat grup target untuk Application Load Balancer Anda](#).
- Anda dapat menentukan grup target yang sama di beberapa pendengar, tetapi pendengar ini harus termasuk dalam penyeimbang beban yang sama. Untuk menggunakan grup target dengan

penyeimbang beban, Anda harus memverifikasi bahwa grup tersebut tidak digunakan oleh pendengar untuk menyeimbang beban lainnya.

- Application Load Balancers tidak mendukung ED25519 kunci.

## Menambahkan pendengar HTTPS

Anda mengonfigurasi pendengar dengan protokol dan port untuk koneksi dari klien ke penyeimbang beban. Untuk informasi selengkapnya, lihat [Konfigurasi listener](#).

Saat membuat pendengar yang aman, Anda harus menentukan kebijakan keamanan dan sertifikat. Untuk menambahkan sertifikat ke daftar sertifikat, lihat [the section called “Menambahkan sertifikat ke daftar sertifikat”](#).

Anda harus mengonfigurasi aturan default untuk pendengar. Anda dapat menambahkan aturan listener lainnya setelah Anda membuat listener. Untuk informasi selengkapnya, lihat [Aturan pendengar](#).

### Console

Untuk menambahkan pendengar HTTPS

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Listeners and rules, pilih Add listener.
5. Untuk Protokol, pilih HTTPS. Simpan port default atau masukkan port yang berbeda.
6. (Opsional) Untuk tindakan Pra-routing, pilih salah satu tindakan berikut:
  - Otentikasi pengguna — Pilih penyedia identitas dan berikan informasi yang diperlukan. Untuk informasi selengkapnya, lihat [Mengautentikasi pengguna menggunakan Application Load Balancer](#).
  - Validasi token — Masukkan titik akhir JWKS, masalah, dan klaim tambahan apa pun. Untuk informasi selengkapnya, lihat [Verifikasi JWTs menggunakan Application Load Balancer](#).
7. Untuk tindakan Routing, pilih salah satu tindakan berikut:

- Teruskan ke grup sasaran - Pilih grup sasaran. Untuk menambahkan grup target lain, pilih Tambahkan grup target, pilih grup target, tinjau bobot relatif, dan perbarui bobot sesuai kebutuhan. Anda harus mengaktifkan kelengketan tingkat grup jika Anda mengaktifkan kekakuan pada salah satu grup target.

Jika Anda tidak memiliki grup target yang memenuhi kebutuhan Anda, pilih Buat grup target untuk membuatnya sekarang. Untuk informasi selengkapnya, lihat [Buat grup target](#).

- Redirect ke URL - Masukkan URL dengan memasukkan setiap bagian secara terpisah pada tab bagian URI, atau dengan memasukkan alamat lengkap pada tab URL Lengkap. Untuk kode Status, pilih sementara (HTTP 302) atau permanen (HTTP 301) berdasarkan kebutuhan Anda.
  - Kembalikan respons tetap - Masukkan kode Respons untuk mengembalikan permintaan klien yang dijatuhkan. Secara opsional, Anda dapat menentukan jenis Konten dan badan Respons.
8. Untuk kebijakan Keamanan, kami memilih kebijakan keamanan yang disarankan. Anda dapat memilih kebijakan keamanan yang berbeda sesuai kebutuhan.
  9. Untuk SSL/TLS sertifikat Default, pilih sertifikat default. Kami juga menambahkan sertifikat default ke daftar SNI. Anda dapat memilih sertifikat menggunakan salah satu opsi berikut:
    - Dari ACM — Pilih sertifikat dari Sertifikat (dari ACM), yang menampilkan sertifikat yang tersedia dari AWS Certificate Manager
    - Dari IAM — Pilih sertifikat dari Sertifikat (dari IAM), yang menampilkan sertifikat yang Anda impor. AWS Identity and Access Management
    - Impor sertifikat - Pilih tujuan untuk sertifikat Anda; Impor ke ACM atau Impor ke IAM. Untuk kunci pribadi Sertifikat, salin dan tempel isi file kunci pribadi (dikodekan PEM). Untuk badan Sertifikat, salin dan tempel isi file sertifikat kunci publik (dikodekan PEM). Untuk Rantai Sertifikat, salin dan tempel konten file rantai sertifikat (dikodekan PEM), kecuali jika Anda menggunakan sertifikat yang ditandatangani sendiri dan tidak penting bahwa browser secara implisit menerima sertifikat.
  10. (Opsional) Untuk mengaktifkan otentikasi timbal balik, di bawah penanganan sertifikat Klien, aktifkan Mutual Authentication (mTLS).

Mode default adalah passthrough. Jika Anda memilih Verifikasi dengan toko kepercayaan:

- Secara default, koneksi dengan sertifikat klien yang kedaluwarsa ditolak. Untuk mengubah perilaku ini, perluas pengaturan mTL lanjutan, lalu di bawah kedaluwarsa sertifikat Klien pilih Izinkan sertifikat klien yang kedaluwarsa.
  - Untuk toko Trust, pilih toko kepercayaan yang ada, atau pilih Toko kepercayaan baru dan berikan informasi yang diperlukan.
11. (Opsional) Untuk menambahkan tag, perluas tag Listener. Pilih Tambahkan tag baru dan masukkan kunci tag dan nilai tag.
  12. Pilih Tambahkan pendengar.

## AWS CLI

Untuk membuat pendengar HTTPS

Gunakan perintah [create-listener](#). Contoh berikut membuat pendengar HTTPS dengan aturan default yang meneruskan lalu lintas ke grup target yang ditentukan.

```
aws elbv2 create-listener \  
  --load-balancer-arn load-balancer-arn \  
  --protocol HTTPS \  
  --port 443 \  
  --default-actions Type=forward,TargetGroupArn=target-group-arn \  
  --ssl-policy ELBSecurityPolicy-TLS13-1-2-2021-06 \  
  --certificates certificate-arn
```

## CloudFormation

Untuk membuat pendengar HTTPS

Tentukan sumber daya tipe [AWS::ElasticLoadBalancingV2::Listener](#). Contoh berikut membuat pendengar HTTPS dengan aturan default yang meneruskan lalu lintas ke grup target yang ditentukan.

```
Resources:  
  myHTTPSListener:  
    Type: 'AWS::ElasticLoadBalancingV2::Listener'  
    Properties:  
      LoadBalancerArn: !Ref myLoadBalancer  
      Protocol: HTTPS  
      Port: 443
```

```
DefaultActions:
  - Type: "forward"
    TargetGroupArn: !Ref myTargetGroup
SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06
Certificates:
  - CertificateArn: certificate-arn
```

## Perbarui listener HTTPS untuk Application Load Balancer Anda

Setelah Anda membuat listener HTTPS, Anda dapat mengganti sertifikat default, memperbarui daftar sertifikat, atau mengganti kebijakan keamanan.

### Tugas

- [Mengganti sertifikat default](#)
- [Menambahkan sertifikat ke daftar sertifikat](#)
- [Menghapus sertifikat dari daftar sertifikat](#)
- [Memperbarui kebijakan keamanan](#)
- [Modifikasi header HTTP](#)

## Mengganti sertifikat default

Anda dapat mengganti sertifikat default untuk listener Anda menggunakan prosedur berikut. Untuk informasi selengkapnya, lihat [Sertifikat default](#).

### Console

Untuk mengganti sertifikat default

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Listeners and rules, pilih teks di kolom Protocol:Port untuk membuka halaman detail bagi listener.
5. Pada tab Sertifikat, pilih Ubah default.
6. Dalam tabel sertifikat ACM dan IAM, pilih sertifikat default baru.

7. (Opsional) Secara default, kami memilih Tambahkan sertifikat default sebelumnya ke daftar sertifikat pendengar. Kami menyarankan agar Anda tetap memilih opsi ini, kecuali saat ini Anda tidak memiliki sertifikat pendengar untuk SNI dan mengandalkan dimulainya kembali sesi TLS.
8. Pilih Simpan sebagai default.

## AWS CLI

Untuk mengganti sertifikat default

Gunakan perintah [modifikasi-listener](#).

```
aws elbv2 modify-listener \  
  --listener-arn listener-arn \  
  --certificates CertificateArn=new-default-certificate-arn
```

## CloudFormation

Untuk mengganti sertifikat default

Perbarui [AWS::ElasticLoadBalancingV2::Listener](#).

```
Resources:  
  myHTTPSListener:  
    Type: 'AWS::ElasticLoadBalancingV2::Listener'  
    Properties:  
      LoadBalancerArn: !Ref myLoadBalancer  
      Protocol: HTTPS  
      Port: 443  
      DefaultActions:  
        - Type: "forward"  
          TargetGroupArn: !Ref myTargetGroup  
      SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06  
      Certificates:  
        - CertificateArn: new-default-certificate-arn
```

## Menambahkan sertifikat ke daftar sertifikat

Anda dapat menambahkan sertifikat ke daftar sertifikat untuk listener Anda menggunakan prosedur berikut. Jika Anda membuat listener menggunakan Konsol Manajemen AWS, kami menambahkan

sertifikat default ke daftar sertifikat untuk Anda. Jika tidak, daftar sertifikat kosong. Menambahkan sertifikat default ke daftar sertifikat memastikan bahwa sertifikat ini digunakan dengan protokol SNI meskipun diganti sebagai sertifikat default. Untuk informasi selengkapnya, lihat [Sertifikat SSL untuk Application Load Balancer](#).

## Console

Untuk menambahkan sertifikat ke daftar sertifikat

1. Buka konsol Amazon EC2 di. <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Listeners and rules, pilih teks di kolom Protocol:Port untuk membuka halaman detail bagi listener.
5. Pilih tab Sertifikat.
6. Untuk menambahkan sertifikat default ke daftar, pilih Tambahkan default ke daftar.
7. Untuk menambahkan sertifikat nondefault ke daftar, lakukan hal berikut:
  - a. Pilih Tambahkan sertifikat.
  - b. Untuk menambahkan sertifikat yang sudah dikelola oleh ACM atau IAM, pilih kotak centang untuk sertifikat dan pilih Sertakan sebagai tertunda di bawah ini.
  - c. Untuk menambahkan sertifikat yang tidak dikelola oleh ACM atau IAM, pilih Impor sertifikat, lengkapi formulir, dan pilih Impor.
  - d. Pilih Tambahkan sertifikat yang tertunda.

## AWS CLI

Untuk menambahkan sertifikat ke daftar sertifikat

Gunakan perintah [add-listener-certificates](#).

```
aws elbv2 add-listener-certificates \  
  --listener-arn listener-arn \  
  --certificates \  
    CertificateArn=certificate-arn-1 \  
    CertificateArn=certificate-arn-2 \  
    CertificateArn=certificate-arn-3
```

## CloudFormation

Untuk menambahkan sertifikat ke daftar sertifikat

Tentukan sumber daya tipe [AWS::ElasticLoadBalancingV2::ListenerCertificate](#).

```
Resources:
  myCertificateList:
    Type: 'AWS::ElasticLoadBalancingV2::ListenerCertificate'
    Properties:
      ListenerArn: !Ref myTLSEListener
      Certificates:
        - CertificateArn: "certificate-arn-1"
        - CertificateArn: "certificate-arn-2"
        - CertificateArn: "certificate-arn-3"
```

## Menghapus sertifikat dari daftar sertifikat

Anda dapat menghapus sertifikat dari daftar sertifikat untuk HTTPS listener menggunakan prosedur berikut. Setelah Anda menghapus sertifikat, pendengar tidak dapat lagi membuat koneksi menggunakan sertifikat tersebut. Untuk memastikan bahwa klien tidak terpengaruh, tambahkan sertifikat baru ke daftar dan konfirmasi bahwa koneksi berfungsi sebelum Anda menghapus sertifikat dari daftar.

Untuk menghapus sertifikat default untuk pendengar TLS, lihat [Mengganti sertifikat default](#).

## Console

Untuk menghapus sertifikat dari daftar sertifikat

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Listeners and rules, pilih teks di kolom Protocol:Port untuk membuka halaman detail bagi listener.
5. Pada tab Sertifikat, pilih kotak centang untuk sertifikat dan pilih Hapus.
6. Saat diminta konfirmasi, masukkan **confirm** dan pilih Hapus.

## AWS CLI

Untuk menghapus sertifikat dari daftar sertifikat

Gunakan perintah [remove-listener-certificates](#).

```
aws elbv2 remove-listener-certificates \  
  --listener-arn listener-arn \  
  --certificates CertificateArn=certificate-arn
```

## Memperbarui kebijakan keamanan

Ketika Anda membuat HTTPS listener, Anda dapat memilih kebijakan keamanan yang sesuai kebutuhan Anda. Ketika kebijakan keamanan baru ditambahkan, Anda dapat memperbarui listener HTTPS Anda untuk menggunakan kebijakan keamanan baru. Application Load Balancer tidak mendukung kebijakan keamanan kustom. Untuk informasi selengkapnya, lihat [Kebijakan keamanan untuk Application Load Balancer](#).

Memperbarui kebijakan keamanan dapat mengakibatkan gangguan jika penyeimbang beban menangani volume lalu lintas yang tinggi. Untuk mengurangi kemungkinan gangguan ketika penyeimbang beban Anda menangani volume lalu lintas yang tinggi, buat penyeimbang beban tambahan untuk membantu menangani lalu lintas atau meminta reservasi LCU.

### Kompatibilitas

- Semua pendengar aman yang terpasang pada penyeimbang beban yang sama harus menggunakan kebijakan keamanan yang kompatibel. Untuk memigrasikan semua pendengar aman untuk penyeimbang beban ke kebijakan keamanan yang tidak kompatibel dengan kebijakan yang saat ini digunakan, hapus semua kecuali satu pendengar aman, ubah kebijakan keamanan pendengar aman, lalu buat pendengar aman tambahan.
  - Kebijakan TLS pasca-kuantum FIPS dan kebijakan FIPS - Kompatibel
  - Kebijakan TLS pasca-kuantum dan kebijakan TLS pasca-kuantum FIPS atau FIPS - Kompatibel
  - Kebijakan TLS (non-FIPS, non-post-quantum) dan kebijakan TLS pasca-kuantum FIPS atau FIPS - Tidak Kompatibel
  - Kebijakan TLS (non-FIPS, non-post-quantum) dan kebijakan TLS pasca-kuantum - Tidak Kompatibel

## Console

Untuk memperbarui kebijakan keamanan

1. Buka konsol Amazon EC2 di. <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Listeners and rules, pilih teks di kolom Protocol:Port untuk membuka halaman detail bagi listener.
5. Pada tab Keamanan, pilih Edit pengaturan pendengar aman.
6. Di bagian Pengaturan pendengar aman, di bawah Kebijakan keamanan, pilih kebijakan keamanan baru.
7. Pilih Simpan perubahan.

## AWS CLI

Untuk memperbarui kebijakan keamanan

Gunakan perintah [modifikasi-listener](#).

```
aws elbv2 modify-listener \  
  --listener-arn listener-arn \  
  --ssl-policy ELBSecurityPolicy-TLS13-1-2-Res-2021-06
```

## CloudFormation

Untuk memperbarui kebijakan keamanan

Perbarui [AWS::ElasticLoadBalancingV2::Listener](#) sumber daya dengan kebijakan keamanan baru.

```
Resources:  
  myHTTPSListener:  
    Type: 'AWS::ElasticLoadBalancingV2::Listener'  
    Properties:  
      LoadBalancerArn: !Ref myLoadBalancer  
      Protocol: HTTPS  
      Port: 443  
      DefaultActions:  
        - Type: "forward"  
          TargetGroupArn: !Ref myTargetGroup
```

```
SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06
Certificates:
  - CertificateArn: certificate-arn
```

## Modifikasi header HTTP

Modifikasi header HTTP memungkinkan Anda untuk mengganti nama header yang dihasilkan penyeimbang beban tertentu, menyisipkan header respons tertentu, dan menonaktifkan header respons server. Application Load Balancers mendukung modifikasi header untuk header permintaan dan respons.

Untuk informasi selengkapnya, lihat [Aktifkan modifikasi header HTTP untuk Application Load Balancer](#).

## Aturan listener untuk Application Load Balancer Anda

Aturan listener untuk Application Load Balancer menentukan cara rute permintaan ke target. Ketika pendengar menerima permintaan, ia mengevaluasi permintaan terhadap setiap aturan dalam urutan prioritas, dimulai dengan aturan bernomor terendah. Setiap aturan mencakup kondisi yang harus dipenuhi dan tindakan yang harus dilakukan ketika kondisi untuk aturan terpenuhi. Mekanisme perutean yang fleksibel ini memungkinkan Anda menerapkan pola distribusi lalu lintas yang canggih, mendukung beberapa aplikasi atau layanan mikro di belakang penyeimbang beban tunggal, dan menyesuaikan penanganan permintaan berdasarkan persyaratan spesifik aplikasi Anda.

### Dasar-dasar aturan

- Setiap aturan terdiri dari komponen-komponen berikut: prioritas, tindakan, kondisi, dan transformasi opsional.
- Setiap tindakan aturan memiliki jenis dan informasi yang diperlukan untuk melakukan tindakan.
- Setiap kondisi aturan memiliki jenis dan informasi yang diperlukan untuk mengevaluasi kondisi tersebut.
- Setiap transformasi aturan memiliki ekspresi reguler untuk mencocokkan dan string pengganti.
- Bila Anda membuat listener, Anda menentukan tindakan untuk peraturan default. Aturan default tidak dapat memiliki kondisi atau transformasi. Jika tidak ada kondisi untuk aturan lain yang terpenuhi, maka tindakan untuk aturan default dilakukan.
- Peraturan dievaluasi dalam urutan prioritas, dari nilai terendah ke nilai tertinggi. Peraturan default dievaluasi terakhir. Anda tidak dapat mengubah prioritas aturan default.

- Setiap peraturan harus mencakup salah satu tindakan berikut: `forward`, `redirect`, atau `fixed-response`, dan harus menjadi tindakan terakhir yang harus dilakukan.
- Setiap aturan selain aturan default secara opsional dapat menyertakan salah satu kondisi berikut: `host-header`, `http-request-methodpath-pattern`, dan `source-ip`. Ini juga dapat secara opsional mencakup satu atau kedua kondisi berikut: `http-header` dan `query-string`.
- Setiap aturan selain aturan default secara opsional dapat menyertakan satu transformasi penulisan ulang header host dan satu transformasi penulisan ulang URL.
- Anda dapat menentukan hingga tiga string perbandingan per syarat dan hingga lima per peraturan.

## Daftar Isi

- [Jenis tindakan untuk aturan pendengar](#)
- [Jenis kondisi untuk aturan pendengar](#)
- [Transformasi untuk aturan pendengar](#)
- [Menambahkan aturan listener untuk Application Load Balancer](#)
- [Mengedit aturan listener untuk Application Load Balancer](#)
- [Menghapus aturan listener untuk Application Load Balancer](#)

## Jenis tindakan untuk aturan pendengar

Tindakan menentukan cara menyeimbangkan beban menangani permintaan saat kondisi untuk aturan pendengar terpenuhi. Setiap aturan harus memiliki setidaknya satu tindakan yang menentukan cara menangani permintaan yang cocok. Setiap tindakan aturan memiliki jenis dan informasi konfigurasi. Application Load Balancers mendukung jenis tindakan berikut untuk aturan listener.

### Jenis tindakan

#### `authenticate-cognito`

[Listener HTTPS] Gunakan Amazon Cognito untuk mengautentikasi pengguna. Untuk informasi selengkapnya, lihat [Otentikasi pengguna](#).

#### `authenticate-oidc`

[Listener HTTPS] Gunakan penyedia identitas yang sesuai dengan OpenID Connect (OIDC) untuk mengautentikasi pengguna. Untuk informasi selengkapnya, lihat [Otentikasi pengguna](#).

## fixed-response

Kembalikan respons HTTP khusus. Untuk informasi selengkapnya, lihat [Tindakan respons tetap](#).

## forward

Meneruskan permintaan ke kelompok target yang ditentukan. Untuk informasi selengkapnya, lihat [Tindakan ke depan](#).

## jwt-validation

Validasi token akses JWT dalam permintaan klien. Untuk informasi selengkapnya, lihat [Verifikasi JWT](#).

## redirect

Mengalihkan permintaan dari satu URL ke URL lainnya. Untuk informasi selengkapnya, lihat [Tindakan pengalihan](#).

## Dasar-dasar tindakan

- Setiap aturan harus menyertakan salah satu tindakan routing berikut: `forward`, `redirect`, atau `fixed-response`, dan itu harus menjadi tindakan terakhir yang harus dilakukan.
- Pendengar HTTPS dapat memiliki aturan dengan tindakan otentikasi pengguna dan tindakan perutean.
- Ketika ada beberapa tindakan, tindakan dengan prioritas terendah dilakukan terlebih dahulu.
- Jika versi protokol adalah gRPC atau HTTP/2, satu-satunya tindakan yang didukung adalah tindakan `forward`.

## Tindakan respons tetap

`fixed-response` Tindakan menjatuhkan permintaan klien dan mengembalikan respons HTTP kustom. Anda dapat menggunakan tindakan ini untuk mengembalikan kode respons 2XX, 4XX, atau 5XX dan pesan opsional.

Saat tindakan `fixed-response` diambil, tindakan dan URL dari target pengalihan dicatat dalam log akses. Untuk informasi selengkapnya, lihat [Entri log akses](#). Hitungan tindakan `fixed-response` yang berhasil dilaporkan dalam metrik `HTTP_Fixed_Response_Count`. Untuk informasi selengkapnya, lihat [Metrik Application Load Balancer](#).

## Example Contoh tindakan respons tetap

Anda dapat menentukan tindakan ketika Anda membuat atau memodifikasi peraturan. Untuk informasi lebih lanjut, lihat perintah [buat-peraturan](#) dan [modifikasi-peraturan](#). Tindakan berikut mengirimkan respons tetap dengan kode status dan tubuh pesan yang ditentukan.

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

## Tindakan ke depan

Tindakan `forward` mengarahkan permintaan ke grup targetnya. Sebelum Anda menambahkan tindakan `forward`, buat kelompok target dan tambahkan target untuk kelompok itu. Untuk informasi selengkapnya, lihat [Buat grup target](#).

### Mendistribusikan lalu lintas ke beberapa kelompok sasaran

Jika Anda menentukan beberapa kelompok target untuk tindakan `forward`, Anda harus menentukan bobot untuk setiap grup target. Bobot setiap grup target adalah nilai dari 0 hingga 999. Permintaan yang sesuai dengan peraturan listener dengan kelompok target tertimbang didistribusikan ke grup target ini berdasarkan bobot mereka. Misalnya, jika Anda menentukan dua grup target, masing-masing dengan bobot 10, setiap grup target menerima setengah dari permintaan. Jika Anda menentukan dua grup target, satu dengan bobot 10 dan lainnya dengan bobot 20, grup target dengan bobot 20 menerima permintaan dua kali lebih banyak dari grup target lainnya.

Jika Anda mengonfigurasi aturan untuk mendistribusikan lalu lintas antara grup target tertimbang dan salah satu grup target kosong atau hanya memiliki target yang tidak sehat, penyeimbang beban tidak secara otomatis gagal ke grup target dengan target yang sehat.

### Sesi lengket dan kelompok sasaran tertimbang

Secara default, mengonfigurasi aturan untuk mendistribusikan lalu lintas di antara grup target berbobot tidak menjamin bahwa sesi lekat akan dipenuhi. Untuk memastikan bahwa sesi lekat

dipatuhi, aktifkan kelekatan grup target untuk peraturan. Saat penyeimbang beban pertama kali merutekan permintaan ke grup target tertimbang, ia menghasilkan cookie bernama AWSALBTG yang mengkodekan informasi tentang grup target yang dipilih, mengenkripsi cookie, dan menyertakan cookie dalam respons terhadap klien. Klien harus menyertakan cookie yang diterimanya dalam permintaan berikutnya ke penyeimbang beban. Saat penyeimbang beban menerima permintaan yang cocok dengan peraturan dengan kelekatan grup target yang diaktifkan dan berisi cookie, permintaan akan diarahkan ke grup target yang ditentukan dalam cookie.

Application Load Balancer tidak mendukung nilai cookie yang diencode URL.

Dengan permintaan CORS (cross-origin resource sharing), beberapa peramban memerlukan `SameSite=None; Secure` untuk mengaktifkan kelekatan. Dalam hal ini, Elastic Load Balancing menghasilkan cookie kedua `AWSALBTGCORS`, yang mencakup informasi yang sama dengan cookie lengket asli ditambah atribut ini. `SameSite` Klien menerima kedua cookie.

Contoh tindakan maju dengan satu grup target

Anda dapat menentukan tindakan ketika Anda membuat atau memodifikasi peraturan. Untuk informasi lebih lanjut, lihat perintah [buat-peraturan](#) dan [modifikasi-peraturan](#). Tindakan berikut meneruskan permintaan ke grup target yang ditentukan.

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

Contoh aksi maju dengan kelompok sasaran tertimbang

Tindakan berikut meneruskan permintaan ke dua grup target yang ditentukan, berdasarkan berat masing-masing grup target.

```
[
```

```

{
  "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
      }
    ]
  }
}
]

```

### Contoh tindakan maju dengan kelengketan diaktifkan

Jika Anda memiliki tindakan maju dengan beberapa grup target dan satu grup target atau lebih memiliki [sesi lekat](#) yang diaktifkan, Anda harus mengaktifkan kelekatan grup target.

Tindakan berikut meneruskan permintaan ke dua grup target yang ditentukan, dengan kelengketan grup target diaktifkan. Permintaan yang tidak berisi cookie kelengketan dirutekan berdasarkan berat setiap grup target.

```

[
  {
    "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
    }
  }
}
```

## Tindakan pengalihan

`redirect` Tindakan mengalihkan permintaan klien dari satu URL ke URL lainnya. Anda dapat mengonfigurasi pengalihan sebagai sementara (HTTP 302) atau permanen (HTTP 301) berdasarkan kebutuhan Anda.

URI terdiri dari komponen-komponen berikut:

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

Anda harus memodifikasi setidaknya satu dari komponen berikut untuk menghindari loop pengalihan: protokol, nama host, port, atau jalur. Setiap komponen yang tidak Anda ubah mempertahankan nilai aslinya.

### protokol

Protokol (HTTP atau HTTPS). Anda dapat mengalihkan HTTP ke HTTP, HTTP ke HTTPS, dan HTTPS ke HTTPS. Anda tidak dapat mengalihkan HTTPS ke HTTP.

### nama host

Nama host. Nama host tidak peka huruf besar/kecil, panjangnya dapat mencapai 128 karakter, dan terdiri dari karakter alfanumerik, karakter pengganti (\* dan ?), dan tanda hubung (-).

### port

Port (1 untuk 65535).

### jalur

Jalur absolut, dimulai dengan awalan "/". Jalur peka huruf besar-kecil, panjangnya dapat mencapai 128 karakter, dan terdiri dari karakter alfanumerik, karakter pengganti (\* dan ?), & (menggunakan &), dan karakter khusus berikut: `_.$/~'"@:+`.

## kueri

Parameter kueri. Panjang maksimum adalah 128 karakter.

Anda dapat menggunakan kembali komponen URI dari URL asli di URL target menggunakan kata kunci cadangan berikut:

- `{protocol}` - Mempertahankan protokol. Gunakan dalam komponen protokol dan kueri.
- `{host}` - Mempertahankan domain. Gunakan di nama host, jalur, dan komponen kueri.
- `{port}` - Mempertahankan port. Gunakan di komponen port, jalur, dan kueri.
- `{path}` - Mempertahankan jalur. Gunakan di jalur dan komponen kueri.
- `{query}` - Mempertahankan parameter kueri. Gunakan dalam komponen kueri.

Saat tindakan `redirect` diambil, tindakan tersebut dicatat dalam log akses. Untuk informasi selengkapnya, lihat [Entri log akses](#). Hitungan tindakan `redirect` yang berhasil dilaporkan dalam metrik `HTTP_Redirect_Count`. Untuk informasi selengkapnya, lihat [Metrik Application Load Balancer](#).

Contoh tindakan pengalihan menggunakan konsol

Redirect menggunakan HTTPS dan port 40443

Aturan berikut menyiapkan pengalihan permanen ke URL yang menggunakan protokol HTTPS dan port tertentu (40443), tetapi mempertahankan nama host, jalur, dan parameter kueri asli. Layar ini setara dengan `https://{host}:40443/{path}?{query}`.

**Routing action** 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**

Used for connections from clients to the load balancer.

HTTPS ▼

**Port**

The port on which the load balancer is listening for connections.

40443

1-65535 or to retain the original port enter #{port}

 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 ▼

Arahkan ulang menggunakan jalur yang dimodifikasi

Aturan berikut menyiapkan pengalihan permanen ke URL yang mempertahankan protokol asli, port, nama host, dan parameter kueri, dan menggunakan kata kunci `#{path}` untuk membuat jalur yang dimodifikasi. Layar ini setara dengan `"#{protocol}://#{host}:#{port}/new/#{path}?#{query}"`.

**Routing action** 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**

Used for connections from clients to the load balancer.

**Port**

The port on which the load balancer is listening for connections.

1-65535 or to retain the original port enter #{port}

 **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.

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.

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

**Query - optional**

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

Maximum 128 characters.

**Status code**

Contoh tindakan pengalihan menggunakan AWS CLI

Redirect menggunakan HTTPS dan port 40443

Anda dapat menentukan tindakan ketika Anda membuat atau memodifikasi peraturan. Untuk informasi selengkapnya, lihat perintah [buat-peraturan](#) dan [modifikasi-peraturan](#). Tindakan berikut mengalihkan permintaan HTTP ke permintaan HTTPS pada port 443, dengan nama host, jalur, dan string kueri yang sama dengan permintaan HTTP.

```
--actions '[{
```

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

## Jenis kondisi untuk aturan pendengar

Ketentuan menentukan kriteria yang harus dipenuhi oleh permintaan masuk agar aturan pendengar berlaku. Jika permintaan cocok dengan kondisi untuk aturan, permintaan akan ditangani seperti yang ditentukan oleh tindakan aturan. Setiap syarat peraturan memiliki jenis dan konfigurasi informasi. Application Load Balancers mendukung jenis kondisi berikut untuk aturan pendengar.

Jenis kondisi

host-header

Rutekan berdasarkan nama host dari setiap permintaan. Untuk informasi selengkapnya, lihat [Syarat host](#).

http-header

Rutekan berdasarkan header HTTP untuk setiap permintaan. Untuk informasi selengkapnya, lihat [Syarat header HTTP](#).

http-request-method

Rutekan berdasarkan metode permintaan HTTP dari setiap permintaan. Untuk informasi selengkapnya, lihat [Syarat metode permintaan HTTP](#).

path-pattern

Rute berdasarkan pola jalur dalam permintaan URLs. Untuk informasi selengkapnya, lihat [Syarat jalur](#).

query-string

Rute berdasarkan key/value pasangan atau nilai dalam string kueri. Untuk informasi selengkapnya, lihat [Syarat string kueri](#).

## source-ip

Rutekan berdasarkan alamat IP sumber dari setiap permintaan. Untuk informasi selengkapnya, lihat [Syarat alamat IP sumber](#).

### Dasar-dasar kondisi

- Setiap aturan secara opsional dapat mencakup nol atau salah satu dari masing-masing kondisi berikut: `host-header`, `http-request-method-path-pattern`, dan `source-ip`. Setiap aturan juga dapat mencakup nol atau lebih dari masing-masing kondisi berikut: `http-header` dan `query-string`.
- Dengan `host-header`, dan `path-pattern` kondisi `http-header`, Anda dapat menggunakan pencocokan nilai atau pencocokan ekspresi reguler (regex).
- Anda dapat menentukan hingga tiga evaluasi kecocokan per syarat. Misalnya, untuk masing-masing syarat `http-header`, Anda dapat menentukan hingga tiga string untuk dibandingkan dengan nilai header HTTP dalam permintaan. Syarat terpenuhi jika salah satu string cocok dengan nilai header HTTP. Untuk mengharuskan semua string cocok, buat satu syarat per evaluasi kecocokan.
- Anda dapat menentukan hingga lima evaluasi kecocokan per peraturan. Misalnya, Anda dapat membuat peraturan dengan lima ketentuan di mana setiap syarat memiliki satu evaluasi kecocokan.
- Anda dapat memasukkan karakter wildcard dalam evaluasi kecocokan untuk syarat `http-header`, `host-header`, `path-pattern`, dan `query-string`. Ada batas lima karakter wildcard per peraturan.
- Peraturan hanya diterapkan pada karakter ASCII yang terlihat; karakter kontrol (0x00 hingga 0x1f dan 0x7f) dikecualikan.

### Demo

Untuk demo, lihat [Perutean permintaan lanjutan](#).

## Syarat host

Anda dapat menggunakan syarat host untuk menentukan peraturan yang merutekan permintaan berdasarkan nama host di header host (juga dikenal sebagai perutean berbasis host). Ini memungkinkan Anda untuk mendukung beberapa subdomain dan domain tingkat atas yang berbeda menggunakan penyeimbang beban tunggal.

Nama host tidak peka huruf besar/kecil, dapat memiliki panjang hingga 128 karakter, dan dapat berisi salah satu dari karakter berikut:

- A–Z, a–z, 0–9
- - .
- \* (cocok dengan 0 karakter atau lebih)
- ? (cocok tepat dengan 1 karakter)

Anda harus menyertakan setidaknya satu karakter ".". Anda hanya dapat memasukkan karakter alfabet setelah akhir karakter ".".

Contoh nama host

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

Peraturan \*.example.com cocok dengan test.example.com tetapi tidak cocok dengan example.com.

Example Contoh kondisi header host

Anda dapat menentukan syarat ketika membuat atau memodifikasi peraturan. Untuk informasi lebih lanjut, lihat perintah [buat-peraturan](#) dan [modifikasi-peraturan](#).

Value matching

```
[
  {
    "Field": "host-header",
    "HostHeaderConfig": {
      "Values": ["*.example.com"]
    }
  }
]
```

Regex matching

```
[
  {
```

```

    "Field": "host-header",
    "HostHeaderConfig": {
      "RegexValues": ["^(.*)\\.example\\.com$"]
    }
  }
]

```

## Syarat header HTTP

Anda dapat menggunakan syarat header HTTP untuk mengonfigurasi aturan yang merutekan permintaan berdasarkan header HTTP untuk permintaan tersebut. Anda dapat menentukan nama-nama bidang header HTTP standar atau kustom. Nama header dan evaluasi kecocokan tidak peka huruf besar/kecil. Karakter wildcard berikut didukung dalam string perbandingan: \* (cocok dengan 0 karakter atau lebih) dan ? (cocok persis dengan 1 karakter). Karakter wildcard tidak didukung dalam nama header.

Ketika atribut Application Load Balancer `routing.http.drop_invalid_header_fields` diaktifkan, itu akan menjatuhkan nama header yang tidak sesuai dengan ekspresi reguler (). A-Z, a-z, 0-9 Nama header yang tidak sesuai dengan ekspresi reguler juga dapat ditambahkan.

### Example Contoh kondisi header HTTP

Anda dapat menentukan syarat ketika membuat atau memodifikasi peraturan. Untuk informasi lebih lanjut, lihat perintah [buat-peraturan](#) dan [modifikasi-peraturan](#). Syarat berikut dipenuhi oleh permintaan dengan header User-Agent yang cocok dengan salah satu string yang ditentukan.

### Value matching

```

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

```

### Regex matching

```

[

```

```
{
  "Field": "http-header",
  "HTTPHeaderConfig": {
    "HttpHeaderName": "User-Agent",
    "RegexValues": [".+"]
  }
}
```

## Syarat metode permintaan HTTP

Anda dapat menggunakan syarat metode permintaan HTTP untuk mengonfigurasi aturan yang merutekan permintaan berdasarkan metode permintaan HTTP dari permintaan tersebut. Anda dapat menentukan metode HTTP standar atau kustom. Evaluasi kecocokan peka terhadap huruf besar-kecil. Karakter wildcard tidak didukung; oleh karena itu, nama metode harus sama persis.

Kami menyarankan Anda merutekan permintaan GET dan HEAD dengan cara yang sama, karena respons terhadap permintaan HEAD mungkin di-cache.

### Example Contoh kondisi metode HTTP

Anda dapat menentukan syarat ketika membuat atau memodifikasi peraturan. Untuk informasi lebih lanjut, lihat perintah [buat-peraturan](#) dan [modifikasi-peraturan](#). Syarat berikut dipenuhi oleh permintaan yang menggunakan metode yang ditentukan.

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

## Syarat jalur

Anda dapat menggunakan syarat jalur untuk menentukan peraturan yang merutekan permintaan berdasarkan URL dalam permintaan (juga dikenal sebagai perutean berbasis jalur).

Pola jalur hanya diterapkan ke jalur URL, bukan ke parameter kuerinya. Ini hanya diterapkan pada karakter ASCII yang terlihat; karakter kontrol (0x00 hingga 0x1f dan 0x7f) dikecualikan.

Evaluasi aturan dilakukan hanya setelah normalisasi URI terjadi.

Pola jalur peka huruf besar-kecil, panjangnya bisa hingga 128 karakter, dan bisa berisi salah satu karakter berikut.

- A–Z, a–z, 0–9
- \_ - . \$ / ~ " ' @ : +
- & (menggunakan &)
- \* (cocok dengan 0 karakter atau lebih)
- ? (cocok tepat dengan 1 karakter)

Jika versi protokol adalah gRPC, syaratnya bisa spesifik untuk paket, layanan, atau metode.

Contoh pola jalur HTTP

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

Contoh pola jalur gRPC

- /package
- /package.service
- /package.service/method

Pola jalur digunakan untuk merutekan permintaan tetapi tidak mengubahnya. Misalnya, jika sebuah peraturan memiliki pola jalur /img/\*, aturan meneruskan permintaan untuk /img/picture.jpg ke grup target yang ditentukan sebagai permintaan untuk /img/picture.jpg.

Example Contoh kondisi pola jalur

Anda dapat menentukan syarat ketika membuat atau memodifikasi peraturan. Untuk informasi lebih lanjut, lihat perintah [buat-peraturan](#) dan [modifikasi-peraturan](#). Syarat berikut dipenuhi oleh permintaan dengan URL yang berisi string yang ditentukan.

Value matching

[

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

## Regex matching

```
[
  {
    "Field": "path-pattern",
    "PathPatternConfig": {
      "RegexValues": ["^\\/api\\/(.*)$"]
    }
  }
]
```

## Syarat string kueri

Anda dapat menggunakan kondisi string kueri untuk mengonfigurasi aturan yang merutekan permintaan berdasarkan key/value pasangan atau nilai dalam string kueri. Evaluasi kecocokan tidak peka huruf besar-kecil. Karakter wildcard berikut didukung: \* (cocok dengan 0 karakter atau lebih) dan ? (cocok persis dengan 1 karakter).

### Example Contoh kondisi string kueri

Anda dapat menentukan syarat ketika membuat atau memodifikasi peraturan. Untuk informasi lebih lanjut, lihat perintah [buat-peraturan](#) dan [modifikasi-peraturan](#). Kondisi berikut dipenuhi oleh permintaan dengan string kueri yang mencakup key/value sepasang "version=v1" atau kunci apa pun yang disetel ke "contoh".

```
[
  {
    "Field": "query-string",
    "QueryStringConfig": {
      "Values": [
        {
          "Key": "version",
          "Value": "v1"
        }
      ]
    }
  }
]
```

```

    },
    {
      "Value": "*example*"
    }
  ]
}
]

```

## Syarat alamat IP sumber

Anda dapat menggunakan syarat alamat IP sumber untuk mengonfigurasi aturan yang merutekan permintaan berdasarkan alamat IP sumber permintaan. Alamat IP harus ditentukan dalam format CIDR. Anda dapat menggunakan keduanya IPv4 dan IPv6 alamat. Karakter wildcard tidak didukung. Anda tidak dapat menentukan 255.255.255.255/32 CIDR untuk kondisi aturan IP sumber.

Jika klien berada di belakang proxy, ini adalah alamat IP proxy, bukan alamat IP klien.

Kondisi ini tidak dipenuhi oleh alamat di X-Forwarded-For header. Untuk mencari alamat di X-Forwarded-For header, gunakan `http-header` kondisi.

### Example Contoh kondisi IP sumber

Anda dapat menentukan syarat ketika membuat atau memodifikasi peraturan. Untuk informasi lebih lanjut, lihat perintah [buat-peraturan](#) dan [modifikasi-peraturan](#). Syarat berikut dipenuhi oleh permintaan dengan alamat IP sumber di salah satu blok CIDR yang ditentukan.

```

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

```

## Transformasi untuk aturan pendengar

Transformasi aturan menulis ulang permintaan masuk sebelum dialihkan ke target. Menulis ulang permintaan tidak mengubah keputusan perutean yang dibuat saat mengevaluasi kondisi aturan. Ini berguna ketika klien mengirim URL atau header host yang berbeda dari yang diharapkan target.

Menggunakan aturan mengubah tanggung jawab untuk memodifikasi jalur, string kueri, dan header host ke penyeimbang beban. Ini menghilangkan kebutuhan untuk menambahkan logika modifikasi kustom ke kode aplikasi Anda atau mengandalkan proxy pihak ketiga untuk melakukan modifikasi.

Application Load Balancers mendukung transformasi berikut untuk aturan pendengar.

## Mengubah

### host-header-rewrite

Menulis ulang header host dalam permintaan. Transformasi menggunakan ekspresi reguler untuk mencocokkan pola di header host dan kemudian menggantinya dengan string pengganti.

### url-rewrite

Menulis ulang URL permintaan. Transformasi menggunakan ekspresi reguler untuk mencocokkan pola di URL permintaan dan kemudian menggantinya dengan string pengganti.

## Ubah dasar-dasar

- Anda dapat menambahkan satu transformasi penulisan ulang header host dan satu transformasi penulisan ulang URL per aturan.
- Anda tidak dapat menambahkan transformasi ke aturan default.
- Jika tidak ada kecocokan pola, permintaan asli dikirim ke target.
- Jika ada kecocokan pola tetapi transformasi gagal, kami mengembalikan kesalahan HTTP 500.

## Transformasi penulisan ulang header host

Anda dapat memodifikasi nama domain yang ditentukan di header host.

### Example Contoh transformasi header host

Anda dapat menentukan transformasi saat membuat atau memodifikasi aturan. Untuk informasi lebih lanjut, lihat perintah [buat-peraturan](#) dan [modifikasi-peraturan](#). Berikut ini adalah contoh transformasi header host. Ini mengubah header host ke titik akhir internal.

```
[
  {
    "Type": "host-header-rewrite",
```

```
    "HostHeaderRewriteConfig": {
      "Rewrites": [
        {
          "Regex": "^mywebsite-(.+).com$",
          "Replace": "internal.dev.$1.myweb.com"
        }
      ]
    }
  ]
}
```

Misalnya, transformasi ini menulis ulang header host `https://mywebsite-example.com/project-a` sebagai `https://internal.dev.example.myweb.com/project-a`.

## Transformasi penulisan ulang URL

Anda dapat memodifikasi jalur atau string kueri URL. Dengan menulis ulang URL pada tingkat penyeimbang beban, frontend Anda URLs dapat tetap konsisten untuk pengguna dan mesin pencari bahkan jika layanan backend Anda berubah. Anda juga dapat menyederhanakan string kueri URL yang kompleks untuk membuatnya lebih mudah bagi pelanggan untuk mengetik.

Perhatikan bahwa Anda tidak dapat mengubah protokol atau port URL, hanya jalur dan string kueri.

### Example Contoh transformasi penulisan ulang URL

Anda dapat menentukan transformasi saat membuat atau memodifikasi aturan. Untuk informasi lebih lanjut, lihat perintah [buat-peraturan](#) dan [modifikasi-peraturan](#). Berikut ini adalah contoh URL rewrite transform. Ini mengubah struktur direktori ke string query.

```
[
  {
    "Type": "url-rewrite",
    "UrlRewriteConfig": {
      "Rewrites": [
        {
          "Regex": "^/dp/([A-Za-z0-9]+)/?$",
          "Replace": "/product.php?id=$1"
        }
      ]
    }
  }
]
```

Misalnya, transformasi ini menulis ulang URL permintaan `https://www.example.com/dp/B09G3HRMW` sebagai `https://www.example.com/product.php?id=B09G3HRMW`.

Bagaimana penulisan ulang URL berbeda dari pengalihan URL

Karakteristik	Pengalihan URL	Penulisan ulang URL
Tampilan URL	Perubahan pada bilah alamat browser	Tidak ada perubahan di bilah alamat browser
Kode status	Menggunakan 301 (permanen) atau 302 (sementara)	Tidak ada perubahan kode status
Pemrosesan	Sisi peramban	Sisi server
Penggunaan umum	Perubahan domain, konsolidasi situs web, memperbaiki tautan yang rusak	Bersihkan URLs untuk SEO, sembunyikan struktur kompleks, berikan pemetaan URL lama

## Menambahkan aturan listener untuk Application Load Balancer

Anda menentukan aturan default saat membuat listener. Anda dapat menentukan aturan tambahan kapan saja. Setiap aturan harus menentukan tindakan dan kondisi, dan secara opsional dapat menentukan transformasi. Untuk informasi selengkapnya, lihat berikut ini:

- [Jenis tindakan](#)
- [Jenis kondisi](#)
- [Mengubah](#)

### Console

Untuk menambahkan aturan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.

4. Pada tab Listeners and rules, pilih teks di kolom Protocol:Port untuk membuka halaman detail bagi listener.
5. Pada tab Aturan, pilih Tambahkan aturan.
6. (Opsional) Untuk menentukan nama untuk aturan Anda, perluas Nama dan tag dan masukkan nama. Untuk menambahkan tag tambahan, pilih Tambahkan tag tambahan dan masukkan kunci tag dan nilai tag.
7. Untuk setiap kondisi, pilih Tambahkan kondisi, pilih jenis kondisi, dan berikan nilai kondisi yang diperlukan:

- Header Host - Pilih jenis pola kecocokan dan masukkan header host.

Pencocokan nilai - Maksimum 128 karakter. Tidak peka terhadap huruf besar-kecil. Karakter yang diizinkan adalah a-z, A-Z, 0-9; karakter khusus berikut: -\_.; dan wildcard (\* dan?). Anda harus menyertakan setidaknya satu karakter ".". Anda hanya dapat memasukkan karakter alfabet setelah akhir karakter ".".

Pencocokan Regex - Maksimum 128 karakter.

- Path - Pilih jenis pola kecocokan dan masukkan jalur.

Pencocokan nilai - Maksimum 128 karakter. Peka huruf besar/kecil. Karakter yang diizinkan adalah a-z, A-Z, 0-9; karakter khusus berikut: \_-.\$/~"@: +; &; dan wildcard (\* dan?).

Pencocokan Regex - Maksimum 128 karakter.

- String kueri - Masukkan pasangan kunci:nilai, atau nilai tanpa kunci.

Maksimal 128 karakter. Tidak peka terhadap huruf besar-kecil. Karakter yang diizinkan adalah a-z, A-Z, 0-9; karakter khusus berikut: \_-.\$/~"@: +& (!, ;, =; dan wildcard (\* dan?).

- Metode permintaan HTTP - Masukkan metode permintaan HTTP.

Maksimal 40 karakter. Peka huruf besar/kecil. Karakter yang diizinkan adalah A-Z, dan karakter khusus berikut: \_-. Wildcard tidak didukung.

- Header HTTP - Pilih jenis pola kecocokan dan masukkan nama header dan string perbandingan.

- Nama header HTTP - Aturan akan menilai permintaan yang berisi header ini untuk mengonfirmasi nilai yang cocok.

Pencocokan nilai - Maksimum 40 karakter. Tidak peka terhadap huruf besar-kecil. Karakter yang diizinkan adalah a-z, A-Z, 0-9, dan karakter khusus berikut: \*? -! # \$ % & ' + . ^ \_ ` | ~. Wildcard tidak didukung.

Pencocokan Regex - Maksimum 128 karakter.

- Nilai header HTTP - Masukkan string untuk dibandingkan dengan nilai header HTTP.

Pencocokan nilai Maksimum 128 karakter. Tidak peka terhadap huruf besar-kecil. Karakter yang diizinkan adalah a-z, A-Z, 0-9; spasi; karakter khusus berikut: ! # \$ % & ' ( ) + , . / : ; < = > @ [ ] ^ \_ ` { } ~ - ; dan wildcard (\* dan?).

Pencocokan Regex - Maksimum 128 karakter.

- Sumber IP — Tentukan alamat IP sumber dalam format CIDR. Keduanya IPv4 dan IPv6 CIDRs diizinkan. Wildcard tidak didukung.
8. (Opsional) Untuk menambahkan transformasi, pilih Tambahkan transformasi, pilih jenis transformasi, dan masukkan ekspresi reguler untuk mencocokkan dan string pengganti.
  9. (Opsional, hanya pendengar HTTPS) Untuk tindakan Pra-perutean, pilih salah satu tindakan berikut:
    - Otentikasi pengguna — Pilih penyedia identitas dan berikan informasi yang diperlukan. Untuk informasi selengkapnya, lihat [Mengautentikasi pengguna menggunakan Application Load Balancer](#).
    - Validasi token — Masukkan titik akhir JWKS, masalah, dan klaim tambahan apa pun. Untuk informasi selengkapnya, lihat [Verifikasi JWTs menggunakan Application Load Balancer](#).
  10. Untuk tindakan Routing, pilih salah satu tindakan berikut:
    - Teruskan ke grup sasaran - Pilih grup sasaran. Untuk menambahkan grup target lain, pilih Tambahkan grup target, pilih grup target, tinjau bobot relatif, dan perbarui bobot sesuai kebutuhan. Anda harus mengaktifkan kelengkapan tingkat grup jika Anda mengaktifkan kekakuan pada salah satu grup target.
    - Redirect ke URL - Masukkan URL dengan memasukkan setiap bagian secara terpisah pada tab bagian URI, atau dengan memasukkan alamat lengkap pada tab URL Lengkap. Untuk kode Status, pilih sementara (HTTP 302) atau permanen (HTTP 301) berdasarkan kebutuhan Anda.

- Kembalikan respons tetap - Masukkan kode Respons untuk mengembalikan permintaan klien yang dijatuhkan. Secara opsional, Anda dapat menentukan jenis Konten dan badan Respons.
11. Pilih Berikutnya.
  12. Untuk Prioritas, masukkan nilai dari 1-50.000. Aturan dievaluasi dalam urutan prioritas dari nilai terendah ke nilai tertinggi.
  13. Pilih Berikutnya.
  14. Pada halaman Tinjau dan buat, pilih Buat.

## AWS CLI

Untuk menambahkan aturan

Gunakan perintah [create-rule](#).

Contoh berikut membuat aturan dengan forward tindakan dan host-header kondisi.

```
aws elbv2 create-rule \  
  --listener-arn listener-arn \  
  --priority 10 \  
  --conditions "Field=host-header,Values=example.com,www.example.com" \  
  --actions "Type=forward,TargetGroupArn=target-group-arn"
```

Untuk membuat tindakan maju yang mendistribusikan lalu lintas antara dua kelompok target, gunakan --actions opsi berikut sebagai gantinya.

```
--actions '[{  
  "Type":"forward",  
  "ForwardConfig":{  
    "TargetGroups":[  
      {"TargetGroupArn":"target-group-1-arn","Weight":50},  
      {"TargetGroupArn":"target-group-2-arn","Weight":50}  
    ]  
  }  
}]'
```

Contoh berikut membuat aturan dengan fixed-response tindakan dan source-ip kondisi.

```
aws elbv2 create-rule \  
  --listener-arn listener-arn \  
  --priority 10 \  
  --conditions "Field=source-ip,Values=10.0.0.0/24" \  
  --actions "Type=fixed-response,ResponseCode=200,ResponseContent=Fixed response"
```

```
--listener-arn listener-arn \
--priority 20 \
--conditions '[{"Field":"source-ip","SourceIpConfig":{"Values":
["192.168.1.0/24","10.0.0.0/16"]}}]' \
--actions "Type=fixed-
response,FixedResponseConfig={StatusCode=403,ContentType=text/
plain,MessageBody='Access denied'}"
```

Contoh berikut membuat aturan dengan redirect tindakan dan http-header kondisi.

```
aws elbv2 create-rule \
--listener-arn listener-arn \
--priority 30 \
--conditions '[{"Field":"http-header","HttpHeaderConfig":
{"HttpHeaderName":"User-Agent","Values":["*Mobile*","*Android*","*iPhone*"]}]' \
--actions
"Type=redirect,RedirectConfig={Host=m.example.com,StatusCode=HTTP_302}"
```

## CloudFormation

Untuk menambahkan aturan

Tentukan sumber daya tipe [AWS::ElasticLoadBalancingV2::ListenerRule](#).

Contoh berikut membuat aturan dengan forward tindakan dan host -header kondisi. Aturan mengirimkan lalu lintas ke grup target yang ditentukan ketika kondisi terpenuhi.

```
Resources:
  myForwardListenerRule:
    Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
    Properties:
      ListenerArn: !Ref myListener
      Priority: 10
      Conditions:
        - Field: host-header
          Values:
            - example.com
            - www.example.com
      Actions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
```

Atau, untuk membuat tindakan maju yang mendistribusikan lalu lintas antara dua kelompok target ketika kondisi terpenuhi, tentukan Actions sebagai berikut.

```

Actions:
- Type: forward
  ForwardConfig:
    TargetGroups:
      - TargetGroupArn: !Ref TargetGroup1
        Weight: 50
      - TargetGroupArn: !Ref TargetGroup2
        Weight: 50

```

Contoh berikut membuat aturan dengan fixed-response tindakan dan source-ip kondisi.

```

Resources:
  myFixedResponseListenerRule:
    Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
    Properties:
      ListenerArn: !Ref myListener
      Priority: 20
      Conditions:
        - Field: source-ip
          SourceIpConfig:
            Values:
              - 192.168.1.0/24
              - 10.0.0.0/16
      Actions:
        - Type: fixed-response
          FixedResponseConfig:
            StatusCode: 403
            ContentType: text/plain
            MessageBody: "Access denied"

```

Contoh berikut membuat aturan dengan redirect tindakan dan http-header kondisi.

```

Resources:
  myRedirectListenerRule:
    Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
    Properties:
      ListenerArn: !Ref myListener
      Priority: 30
      Conditions:

```

```
- Field: http-header
  HttpHeadersConfig:
    HttpHeadersName: User-Agent
    Values:
      - "*Mobile*"
      - "*Android*"
      - "*iPhone*"
Actions:
- Type: redirect
  RedirectConfig:
    Host: m.example.com
    StatusCode: HTTP_302
```

## Mengedit aturan listener untuk Application Load Balancer

Anda dapat mengedit tindakan dan ketentuan untuk aturan pendengar kapan saja. Pembaruan peraturan tidak berlaku segera, sehingga permintaan dapat diarahkan menggunakan konfigurasi peraturan sebelumnya untuk waktu yang singkat setelah Anda memperbarui peraturan. Semua permintaan yang sedang berjalan diselesaikan.

### Tugas

- [Ubah tindakan default](#)
- [Perbarui prioritas aturan](#)
- [Perbarui tindakan, kondisi, dan transformasi](#)
- [Mengelola tag aturan](#)

### Ubah tindakan default

Tindakan default ditetapkan ke aturan bernama Default. Anda dapat menyimpan jenis aturan saat ini dan mengubah informasi yang diperlukan, atau Anda dapat mengubah jenis aturan dan memberikan informasi baru yang diperlukan.

### Console

Untuk memodifikasi tindakan default

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.

3. Pilih penyeimbang beban.
4. Pada tab Listeners and rules, pilih teks di kolom Protocol:Port untuk membuka halaman detail bagi listener.
5. Pada tab Aturan, di bagian Aturan pendengar, pilih aturan default. Pilih Tindakan, Edit aturan.
6. Di bawah Tindakan default, perbarui tindakan sesuai kebutuhan.

## AWS CLI

Untuk memodifikasi tindakan default

Gunakan perintah [modifikasi-listener](#). Contoh berikut memperbarui grup target untuk forward tindakan tersebut.

```
aws elbv2 modify-listener \
  --listener-arn listener-arn \
  --default-actions Type=forward,TargetGroupArn=new-target-group-arn
```

Contoh berikut memperbarui tindakan default untuk mendistribusikan lalu lintas secara merata antara dua kelompok target.

```
aws elbv2 modify-listener \
  --listener-arn listener-arn \
  --default-actions '[{
    "Type":"forward",
    "ForwardConfig":{
      "TargetGroups":[
        {"TargetGroupArn":"target-group-1-arn", "Weight":50},
        {"TargetGroupArn":"target-group-2-arn", "Weight":50}
      ]
    }
  }]
```

## CloudFormation

Untuk memodifikasi tindakan default

Perbarui [AWS::ElasticLoadBalancingV2::Listener](#) sumber daya.

Resources :

```
myHTTPListener:
  Type: 'AWS::ElasticLoadBalancingV2::Listener'
  Properties:
    LoadBalancerArn: !Ref myLoadBalancer
    Protocol: HTTP
    Port: 80
    DefaultActions:
      - Type: "forward"
        TargetGroupArn: !Ref myNewTargetGroup
```

## Perbarui prioritas aturan

Peraturan dievaluasi dalam urutan prioritas, dari nilai terendah ke nilai tertinggi. Peraturan default dievaluasi terakhir. Anda dapat mengubah prioritas peraturan nondefault kapan saja. Anda tidak dapat mengubah prioritas aturan default.

### Console

Untuk memperbarui prioritas aturan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Di panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Listeners and rules, pilih teks di kolom Protocol:Port untuk membuka halaman detail bagi listener.
5. Pada tab Aturan, pilih aturan listener dan kemudian pilih Tindakan, Prioritaskan ulang aturan.
6. Di bagian Aturan Listener, kolom Prioritas menampilkan prioritas aturan saat ini. Untuk memperbarui prioritas aturan, masukkan nilai dari 1-50.000.
7. Pilih Simpan perubahan.

### AWS CLI

Untuk memperbarui prioritas aturan

Gunakan perintah [set-rule-priorities](#).

```
aws elbv2 set-rule-priorities \  
  --rule-priorities "RuleArn=listener-rule-arn,Priority=5"
```

## CloudFormation

Untuk memperbarui prioritas aturan

Perbarui [AWS::ElasticLoadBalancingV2::ListenerRule](#) sumber daya.

```
Resources:
  myListenerRule:
    Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'
    Properties:
      ListenerArn: !Ref myListener
      Priority: 5
      Conditions:
        - Field: host-header
          Values:
            - example.com
            - www.example.com
      Actions:
        - Type: forward
          TargetGroupArn: !Ref myTargetGroup
```

## Perbarui tindakan, kondisi, dan transformasi

Anda dapat memperbarui tindakan, kondisi, dan transformasi untuk suatu aturan.

### Console

Untuk memperbarui tindakan aturan, kondisi, dan transformasi

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Di panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Listeners and rules, pilih teks di kolom Protocol:Port untuk membuka halaman detail bagi listener.
5. Pada tab Aturan, pilih aturan listener dan kemudian pilih Tindakan, Edit aturan.
6. Perbarui tindakan, kondisi, dan transformasi sesuai kebutuhan. Untuk langkah mendetail, lihat [Tambahkan peraturan](#).
7. Pilih Berikutnya.
8. (Opsional) Perbarui prioritas.

9. Pilih Berikutnya.
10. Pilih Simpan perubahan.

## AWS CLI

Untuk memperbarui tindakan aturan, kondisi, dan transformasi

Gunakan perintah [modifikasi-peraturan](#). Sertakan setidaknya satu dari opsi berikut: `--actions`, `--conditions`, dan `--transforms`.

Untuk contoh opsi ini, lihat [Tambahkan peraturan](#).

## CloudFormation

Untuk memperbarui tindakan aturan, kondisi, dan transformasi

Perbarui [AWS::ElasticLoadBalancingV2::ListenerRules](#) sumber daya.

Misalnya aturan, lihat [Tambahkan peraturan](#).

## Mengelola tag aturan

Tag membantu Anda mengkategorikan pendengar dan aturan Anda dengan cara yang berbeda. Misalnya, Anda dapat menandai sumber daya berdasarkan tujuan, pemilik, atau lingkungan. Kunci tag harus unik untuk setiap aturan. Jika Anda menambahkan tag dengan kunci yang sudah terkait dengan aturan, maka nilai tag tersebut akan diperbarui.

Setelah selesai dengan tag, Anda dapat menghapusnya.

## Console

Untuk mengelola tag untuk aturan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Di panel navigasi, pilih Load Balancers.
3. Pilih nama penyeimbang beban untuk membuka halaman detailnya.
4. Pada tab Listeners and rules, pilih teks di kolom Protocol:Port untuk membuka halaman detail bagi listener.
5. Pada tab Aturan, pilih teks di kolom Tag nama untuk membuka halaman detail aturan.
6. Pada halaman detail aturan, pilih Kelola tag.

7. Pada halaman Kelola tag, lakukan satu atau beberapa hal berikut:
  - a. Untuk menambahkan tag, pilih Tambahkan tag baru dan masukkan nilai untuk Kunci dan Nilai.
  - b. Untuk menghapus tanda, pilih Hapus di samping tanda.
  - c. Untuk memperbarui tag, masukkan nilai baru untuk Kunci atau Nilai.
8. Pilih Simpan perubahan.

## AWS CLI

Untuk menambahkan tag ke aturan

Gunakan perintah [add-tag](#).

```
aws elbv2 add-tags \  
  --resource-arns listener-rule-arn \  
  --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

Untuk menghapus tag dari aturan

Gunakan perintah [remove-tag](#).

```
aws elbv2 remove-tags \  
  --resource-arns listener-rule-arn \  
  --tag-keys project department
```

## CloudFormation

Untuk menambahkan tag ke aturan

Perbarui [AWS::ElasticLoadBalancingV2::ListenerRule](#) sumber daya.

```
Resources:  
  myListenerRule:  
    Type: 'AWS::ElasticLoadBalancingV2::ListenerRule'  
    Properties:  
      ListenerArn: !Ref myListener  
      Priority: 10  
      Conditions:  
        - Field: host-header  
          Values:
```

```
- example.com
- www.example.com
Actions:
- Type: forward
  TargetGroupArn: !Ref myTargetGroup
Tags:
- Key: 'project'
  Value: 'Lima'
- Key: 'department'
  Value: 'digital-media'
```

## Menghapus aturan listener untuk Application Load Balancer

Anda dapat menghapus peraturan nondefault untuk listener kapan saja. Anda tidak dapat menghapus aturan default untuk pendengar. Bila Anda menghapus listener, semua peraturan akan dihapus.

### Console

Untuk menghapus aturan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Listeners and rules, pilih teks di kolom Protocol:Port untuk membuka halaman detail bagi listener.
5. Pilih aturannya.
6. Pilih Tindakan, Hapus aturan.
7. Saat diminta konfirmasi, masukkan **confirm**, lalu pilih Hapus.

### AWS CLI

Untuk menghapus aturan

Gunakan perintah [hapus-peraturan](#).

```
aws elbv2 delete-rule \  
  --rule-arn listener-rule-arn
```

## Otentikasi timbal balik dengan TLS di Application Load Balancer

Mutual TLS authentication adalah variasi dari transport layer security (TLS). TLS tradisional membangun komunikasi yang aman antara server dan klien, di mana server perlu memberikan identitasnya kepada kliennya. Dengan TLS timbal balik, penyeimbang beban menegosiasikan otentikasi timbal balik antara klien dan server saat menegosiasikan TLS. Ketika Anda menggunakan TLS timbal balik dengan Application Load Balancer Anda, Anda menyederhanakan manajemen otentikasi dan mengurangi beban pada aplikasi Anda.

Dengan menggunakan TLS timbal balik, penyeimbang beban Anda dapat mengelola otentikasi klien untuk membantu memastikan bahwa hanya klien tepercaya yang berkomunikasi dengan aplikasi backend Anda. Saat Anda menggunakan fitur ini, penyeimbang beban mengotentikasi klien menggunakan sertifikat dari otoritas sertifikat pihak ketiga (CA) atau dengan menggunakan AWS Private Certificate Authority (PCA), secara opsional, dengan pemeriksaan pencabutan. Load balancer meneruskan informasi sertifikat klien ke backend menggunakan header HTTP, yang dapat digunakan aplikasi Anda untuk otorisasi.

Mutual TLS for Application Load Balancers menyediakan opsi berikut untuk memvalidasi sertifikat klien X.509v3 Anda:

- Passthrough TLS Mutual: Penyeimbang beban mengirimkan seluruh rantai sertifikat klien ke target, tanpa memverifikasinya. Target harus memverifikasi rantai sertifikat klien. Kemudian, dengan menggunakan rantai sertifikat klien, Anda dapat menerapkan otentikasi penyeimbang beban dan logika otorisasi target dalam aplikasi Anda.
- Verifikasi TLS bersama: Penyeimbang beban melakukan otentikasi sertifikat klien X.509 untuk klien saat penyeimbang beban menegosiasikan koneksi TLS.

Untuk menggunakan passthrough TLS timbal balik, Anda harus mengonfigurasi pendengar untuk menerima sertifikat dari klien. Untuk menggunakan TLS timbal balik dengan verifikasi, lihat [Mengkonfigurasi TLS timbal balik pada Application Load Balancer](#).

## Sebelum Anda mulai mengonfigurasi TLS timbal balik pada Application Load Balancer Anda

Sebelum Anda mulai mengonfigurasi TLS timbal balik pada Application Load Balancer Anda, perhatikan hal-hal berikut:

## Kuota

Application Load Balancer mencakup batas-batas tertentu yang terkait dengan jumlah trust store, sertifikat CA, dan daftar pencabutan sertifikat yang digunakan dalam akun Anda. AWS

Untuk informasi selengkapnya, lihat [Kuota untuk Penyeimbang Beban Aplikasi Anda](#).

## Persyaratan untuk sertifikat

Application Load Balancers mendukung hal berikut untuk sertifikat yang digunakan dengan otentikasi TLS timbal balik:

- Sertifikat yang didukung: X.509v3
- Kunci publik yang didukung: RSA 2K - 8K atau ECDSA secp256r1, secp384r1, secp521r1
- Algoritma tanda tangan yang didukung: SHA256, 384, 512 dengan hash RSA/SHA256, 384, 512 with EC/SHA 256.384.512 dengan RSASSA-PSS dengan MGF1

## Bundel sertifikat CA

Berikut ini berlaku untuk bundel otoritas sertifikat (CA):

- Application Load Balancer mengunggah setiap bundel sertifikat otoritas sertifikat (CA) sebagai batch. Application Load Balancers tidak mendukung pengunggahan sertifikat individual. Jika Anda perlu menambahkan sertifikat baru, Anda harus mengunggah file bundel sertifikat.
- Untuk mengganti bundel sertifikat CA, gunakan [ModifyTrustStoreAPI](#).

## Pesanan sertifikat untuk passthrough

Bila Anda menggunakan passthrough TLS timbal balik, Application Load Balancer menyisipkan header untuk menyajikan rantai sertifikat klien ke target backend. Urutan presentasi dimulai dengan sertifikat daun dan diakhiri dengan sertifikat root.

## Dimulainya kembali sesi

Dimulainya kembali sesi tidak didukung saat menggunakan passthrough TLS timbal balik atau mode verifikasi dengan Application Load Balancer.

## Header HTTP

Application Load Balancers menggunakan X-Amzn-Mtls header untuk mengirim informasi sertifikat ketika menegosiasikan koneksi klien menggunakan TLS timbal balik. Untuk informasi selengkapnya dan contoh header, lihat [Header HTTP dan TLS timbal balik](#).

## Berkas sertifikat CA

File sertifikat CA harus memenuhi persyaratan berikut:

- File sertifikat harus menggunakan format PEM (Privacy Enhanced Mail).
- Isi sertifikat harus dilampirkan dalam -----BEGIN CERTIFICATE----- dan -----END CERTIFICATE----- batas-batas.
- Komentar harus didahului oleh # karakter dan tidak boleh mengandung karakter apa pun.
- Tidak mungkin ada garis kosong.

Contoh sertifikat yang tidak diterima (tidak valid):

```
# 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-----
```

Contoh sertifikat yang diterima (valid):

1. Sertifikat tunggal (PEM — dikodekan):

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

2. Beberapa sertifikat (PEM — dikodekan):

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

## Header HTTP dan TLS timbal balik

Bagian ini menjelaskan header HTTP yang digunakan Application Load Balancer untuk mengirim informasi sertifikat saat menegosiasikan koneksi dengan klien menggunakan TLS bersama. `X-Amzn-MtlsHeader` spesifik yang digunakan Application Load Balancer bergantung pada mode TLS timbal balik yang telah Anda tentukan: mode passthrough atau mode verifikasi.

Untuk informasi tentang header HTTP lain yang didukung oleh Application Load Balancers, lihat.

[Header HTTP dan Application Load Balancer](#)

### Header HTTP untuk mode passthrough

Untuk TLS timbal balik dalam mode passthrough, Application Load Balancers menggunakan header berikut.

#### X-Amzn-Mtls-Sertifikat Pelanggan

Header ini berisi format PEM yang dikodekan URL dari seluruh rantai sertifikat klien yang disajikan dalam koneksi, dengan karakter yang aman. +=/

Contoh isi header:

```
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
```

## Header HTTP untuk mode verifikasi

Untuk TLS timbal balik dalam mode verifikasi, Application Load Balancers menggunakan header berikut.

### X-Amzn-MTLS-Nomor Seri Serial Klien

Header ini berisi representasi heksadesimal dari nomor seri sertifikat daun.

Contoh isi header:

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

### X-Amzn-Mtls-Penerbit-Klien

Header ini berisi representasi RFC2253 string dari nama terhormat penerbit (DN).

Contoh isi header:

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

### X-Amzn-Mtls-Clientcert-Subjek

Header ini berisi representasi RFC2253 string dari nama terhormat subjek (DN).

Contoh isi header:

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

### X-Amzn-Mtls-Clientcert-Validitas

Header ini berisi ISO8601 format notBefore dan notAfter tanggal.

Contoh isi header:

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

## X-Amzn-Mtls-Clientcert-Leaf

Header ini berisi format PEM yang dikodekan URL dari sertifikat daun, dengan karakter yang aman.  
+="/

Contoh isi header:

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

## Nama subjek Advertise Certificate Authority (CA)

Nama subjek Advertising Certificate Authority (CA) meningkatkan proses otentikasi dengan membantu klien menentukan sertifikat mana yang akan diterima selama otentikasi TLS bersama.

Saat Anda mengaktifkan Advertise CA nama subjek, Application Load Balancer akan mengiklankan daftar nama subjek Certificate Authority CAs () yang dipercaya, berdasarkan trust store yang terkait dengannya. Ketika klien terhubung ke target melalui Application Load Balancer, klien menerima daftar nama subjek CA tepercaya.

Selama jabat tangan TLS, ketika Application Load Balancer meminta sertifikat klien, sertifikat tersebut menyertakan daftar CA Distinguished Names DNS () tepercaya dalam pesan Permintaan Sertifikat. Ini membantu klien memilih sertifikat valid yang cocok dengan nama subjek CA yang diiklankan, merampingkan proses otentikasi dan mengurangi kesalahan koneksi.

Anda dapat mengaktifkan Advertise CA nama subjek pada pendengar baru dan yang sudah ada. Untuk informasi selengkapnya, lihat [Menambahkan pendengar HTTPS](#).

## Log koneksi untuk Application Load Balancers

Elastic Load Balancing menyediakan log koneksi yang menangkap atribut tentang permintaan yang dikirim ke Application Load Balancers Anda. Log koneksi berisi informasi seperti alamat IP klien dan port, informasi sertifikat klien, hasil koneksi, dan cipher TLS yang digunakan. Log koneksi ini kemudian dapat digunakan untuk meninjau pola permintaan, dan tren lainnya.

Untuk mempelajari lebih lanjut tentang log koneksi, lihat [Log koneksi untuk Application Load Balancer](#)

## Mengkonfigurasi TLS timbal balik pada Application Load Balancer

Untuk menggunakan mode passthrough TLS timbal balik, Anda hanya perlu mengonfigurasi pendengar untuk menerima sertifikat apa pun dari klien. Bila Anda menggunakan passthrough TLS timbal balik, Application Load Balancer mengirimkan seluruh rantai sertifikat klien ke target menggunakan header HTTP, yang memungkinkan Anda untuk menerapkan logika otentikasi dan otorisasi yang sesuai dalam aplikasi Anda. Untuk informasi selengkapnya, lihat [Membuat pendengar HTTPS untuk Application Load Balancer Anda](#).

Saat Anda menggunakan TLS timbal balik dalam mode verifikasi, Application Load Balancer melakukan otentikasi sertifikat klien X.509 untuk klien saat penyeimbang beban menegosiasikan koneksi TLS.

Untuk memanfaatkan modus verifikasi TLS timbal balik, lakukan hal berikut:

- Buat sumber daya toko kepercayaan baru.
- Unggah bundel otoritas sertifikat (CA) Anda dan, secara opsional, daftar pencabutan.
- Lampirkan trust store ke listener yang dikonfigurasi untuk memverifikasi sertifikat klien.

Gunakan prosedur berikut untuk mengonfigurasi modus verifikasi TLS timbal balik pada Application Load Balancer Anda.

### Tugas

- [Buat toko kepercayaan](#)
- [Kaitkan toko kepercayaan](#)
- [Ganti bundel sertifikat CA](#)
- [Menambahkan daftar pencabutan sertifikat](#)
- [Menghapus daftar pencabutan sertifikat](#)
- [Hapus toko kepercayaan](#)

### Buat toko kepercayaan

Jika Anda menambahkan toko kepercayaan saat membuat penyeimbang beban atau pendengar, toko kepercayaan secara otomatis dikaitkan dengan pendengar baru. Jika tidak, Anda harus mengaitkannya dengan pendengar sendiri.

## Prasyarat

- Untuk membuat toko kepercayaan, Anda harus memiliki bundel sertifikat dari Otoritas Sertifikat (CA) Anda.

## Console

Contoh berikut membuat toko kepercayaan menggunakan bagian Trust Store konsol. Atau, Anda dapat membuat toko kepercayaan saat membuat pendengar HTTP.

Untuk membuat toko kepercayaan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Trust Stores.
3. Pilih Buat toko kepercayaan.
4. Konfigurasi toko kepercayaan
  - a. Untuk nama toko Trust, masukkan nama untuk toko kepercayaan Anda.
  - b. Untuk paket otoritas Sertifikat, masukkan jalur Amazon S3 ke bundel sertifikat ca yang akan digunakan.
  - c. (Opsional) Gunakan versi Object untuk memilih versi sebelumnya dari bundel sertifikat ca. Jika tidak, versi saat ini digunakan.
5. (Opsional) Untuk Pencabutan, Anda dapat menambahkan daftar pencabutan sertifikat ke toko kepercayaan Anda.
  - a. Pilih Tambahkan CRL baru dan masukkan lokasi daftar pencabutan sertifikat di Amazon S3.
  - b. (Opsional) Gunakan versi Objek untuk memilih versi sebelumnya dari daftar pencabutan sertifikat. Jika tidak, versi saat ini digunakan.
6. (Opsional) Perluas tag toko Trust dan masukkan hingga 50 tag untuk toko kepercayaan Anda.
7. Pilih Buat toko kepercayaan.

## AWS CLI

Untuk membuat toko kepercayaan

Gunakan perintah [create-trust-store](#).

```
aws elbv2 create-trust-store \  
  --name my-trust-store \  
  --ca-certificates-bundle-s3-bucket amzn-s3-demo-bucket \  
  --ca-certificates-bundle-s3-key certificates/ca-bundle.pem
```

## CloudFormation

Untuk membuat toko kepercayaan

Tentukan sumber daya tipe [AWS::ElasticLoadBalancingV2::TrustStore](#).

```
Resources:  
  myTrustStore:  
    Type: 'AWS::ElasticLoadBalancingV2::TrustStore'  
    Properties:  
      Name: my-trust-store  
      CaCertificatesBundleS3Bucket: amzn-s3-demo-bucket  
      CaCertificatesBundleS3Key: certificates/ca-bundle.pem
```

## Kaitkan toko kepercayaan

Setelah Anda membuat toko kepercayaan, Anda harus mengaitkannya dengan pendengar sebelum Application Load Balancer Anda dapat mulai menggunakan toko kepercayaan. Anda hanya dapat memiliki satu toko kepercayaan yang terkait dengan masing-masing pendengar aman Anda, tetapi satu toko kepercayaan dapat dikaitkan dengan beberapa pendengar.

## Console

Anda dapat mengaitkan toko kepercayaan dengan pendengar yang ada, seperti yang ditunjukkan dalam prosedur berikut. Atau, Anda dapat mengaitkan toko kepercayaan saat membuat pendengar HTTPS. Untuk informasi selengkapnya, lihat [Membuat pendengar HTTPS](#).

Untuk mengaitkan toko kepercayaan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.

4. Pada tab Listeners and rules, pilih link di kolom Protocol:Port untuk membuka halaman detail bagi listener aman.
5. Pada tab Keamanan, pilih Edit pengaturan pendengar aman.
6. Jika TLS timbal balik tidak diaktifkan, pilih Mutual authentication (mTLS) di bawah penanganan sertifikat Klien dan kemudian pilih Verifikasi dengan trust store.
7. Untuk toko Trust, pilih toko kepercayaan.
8. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaitkan toko kepercayaan

Gunakan perintah [modifikasi-listener](#).

```
aws elbv2 modify-listener \  
  --listener-arn listener-arn \  
  --mutual-authentication "Mode=verify,TrustStoreArn=trust-store-arn"
```

## CloudFormation

Untuk mengaitkan toko kepercayaan

Perbarui [AWS::ElasticLoadBalancingV2::Listener](#) sumber daya.

```
Resources:  
  myHTTPSListener:  
    Type: 'AWS::ElasticLoadBalancingV2::Listener'  
    Properties:  
      LoadBalancerArn: !Ref myLoadBalancer  
      Protocol: HTTPS  
      Port: 443  
      DefaultActions:  
        - Type: "forward"  
          TargetGroupArn: !Ref myTargetGroup  
      SslPolicy: ELBSecurityPolicy-TLS13-1-2-2021-06  
      Certificates:  
        - CertificateArn: certificate-arn  
      MutualAuthentication:  
        - Mode: verify
```

```
TrustStoreArn: trust-store-arn
```

## Ganti bundel sertifikat CA

Bundel sertifikat CA adalah komponen yang diperlukan dari toko kepercayaan. Ini adalah kumpulan sertifikat root dan perantara tepercaya yang telah divalidasi oleh otoritas sertifikat. Sertifikat yang divalidasi ini memastikan klien dapat mempercayai sertifikat yang disajikan dimiliki oleh penyeimbang beban.

Toko kepercayaan hanya dapat berisi satu bundel sertifikat CA pada satu waktu, tetapi Anda dapat mengganti bundel sertifikat CA kapan saja setelah toko kepercayaan dibuat.

### Console

Untuk mengganti bundel sertifikat CA

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Trust Stores.
3. Pilih toko kepercayaan.
4. Pilih Tindakan, Ganti bundel CA.
5. Pada halaman bundel Ganti CA, di bawah bundel otoritas Sertifikat, masukkan lokasi Amazon S3 dari bundel CA yang diinginkan.
6. (Opsional) Gunakan versi Objek untuk memilih versi sebelumnya dari daftar pencabutan sertifikat. Jika tidak, versi saat ini digunakan.
7. Pilih Ganti bundel CA.

### AWS CLI

Untuk mengganti bundel sertifikat CA

Gunakan perintah [modify-trust-store](#).

```
aws elbv2 modify-trust-store \  
  --trust-store-arn trust-store-arn \  
  --ca-certificates-bundle-s3-bucket amzn-s3-demo-bucket-new \  
  --ca-certificates-bundle-s3-key certificates/new-ca-bundle-pem
```

## CloudFormation

Untuk memperbarui bundel sertifikat CA

Tentukan sumber daya tipe [AWS::ElasticLoadBalancingV2::TrustStore](#).

```
Resources:
  myTrustStore:
    Type: 'AWS::ElasticLoadBalancingV2::TrustStore'
    Properties:
      Name: my-trust-store
      CaCertificatesBundleS3Bucket: amzn-s3-demo-bucket-new
      CaCertificatesBundleS3Key: certificates/new-ca-bundle.pem
```

## Menambahkan daftar pencabutan sertifikat

Secara opsional, Anda dapat membuat daftar pencabutan sertifikat untuk toko kepercayaan. Daftar pencabutan dirilis oleh otoritas sertifikat dan berisi data untuk sertifikat yang telah dicabut. Application Load Balancers hanya mendukung daftar pencabutan sertifikat dalam format PEM.

Ketika daftar pencabutan sertifikat ditambahkan ke toko kepercayaan, itu akan diberikan ID pencabutan. IDs Peningkatan pencabutan untuk setiap daftar pencabutan yang ditambahkan ke toko kepercayaan, dan mereka tidak dapat diubah.

Application Load Balancers tidak dapat mencabut sertifikat yang memiliki nomor seri negatif dalam daftar pencabutan sertifikat.

## Console

Untuk menambahkan daftar pencabutan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Trust Stores.
3. Pilih toko kepercayaan untuk melihat halaman detailnya.
4. Pada tab Daftar pencabutan sertifikat, pilih Tindakan, Tambahkan daftar pencabutan.
5. Pada halaman Tambahkan daftar pencabutan, di bawah daftar pencabutan sertifikat masukkan lokasi Amazon S3 dari daftar pencabutan sertifikat yang diinginkan
6. (Opsional) Gunakan versi Objek untuk memilih versi sebelumnya dari daftar pencabutan sertifikat. Jika tidak, versi saat ini digunakan.

## 7. Pilih Tambahkan daftar pencabutan

### AWS CLI

Untuk menambahkan daftar pencabutan

Gunakan perintah [add-trust-store-revocations](#).

```
aws elbv2 add-trust-store-revocations \  
  --trust-store-arn trust-store-arn \  
  --revocation-contents "S3Bucket=amzn-s3-demo-bucket,S3Key=crl/revoked-  
list.crl,RevocationType=CRL"
```

### CloudFormation

Untuk menambahkan daftar pencabutan

Tentukan sumber daya tipe [AWS::ElasticLoadBalancingV2::TrustStorePencabutan](#).

```
Resources:  
  myRevocationContents:  
    Type: 'AWS:ElasticLoadBalancingV2::TrustStoreRevocation'  
    Properties:  
      TrustStoreArn: !Ref myTrustStore  
      RevocationContents:  
        - RevocationType: CRL  
          S3Bucket: amzn-s3-demo-bucket  
          S3Key: crl/revoked-list.crl
```

## Menghapus daftar pencabutan sertifikat

Ketika Anda tidak lagi memerlukan daftar pencabutan sertifikat, Anda dapat menghapusnya. Saat Anda menghapus daftar pencabutan sertifikat dari toko kepercayaan, ID pencabutan itu juga dihapus dan tidak digunakan kembali selama masa pakai toko kepercayaan.

### Console

Untuk menghapus daftar pencabutan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>

2. Pada panel navigasi, pilih Trust Stores.
3. Pilih toko kepercayaan.
4. Pada tab Daftar pencabutan sertifikat, pilih Tindakan, Hapus daftar pencabutan.
5. Saat diminta mengonfirmasi, pilih **confirm**.
6. Pilih Hapus.

## AWS CLI

Untuk menghapus daftar pencabutan

Gunakan perintah [remove-trust-store-revocations](#).

```
aws elbv2 remove-trust-store-revocations \  
  --trust-store-arn trust-store-arn \  
  --revocation-ids id-1 id-2 id-3
```

## Hapus toko kepercayaan

Ketika Anda tidak lagi memiliki penggunaan untuk toko kepercayaan, Anda dapat menghapusnya. Anda tidak dapat menghapus toko kepercayaan yang terkait dengan pendengar.

## Console

Untuk menghapus toko kepercayaan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Trust Stores.
3. Pilih toko kepercayaan.
4. Pilih Hapus.
5. Saat diminta konfirmasi, masukkan **confirm**, lalu pilih Hapus.

## AWS CLI

Untuk menghapus toko kepercayaan

Gunakan perintah [delete-trust-store](#).

```
aws elbv2 delete-trust-store \  
  --trust-store-arn trust-store-arn
```

## Bagikan toko kepercayaan Elastic Load Balancing Anda untuk Application Load Balancers

Elastic Load Balancing terintegrasi dengan AWS Resource Access Manager (AWS RAM) untuk memungkinkan berbagi toko kepercayaan. AWS RAM adalah layanan yang memungkinkan Anda berbagi sumber daya penyimpanan kepercayaan Elastic Load Balancing dengan aman Akun AWS di seluruh dan di dalam organisasi atau unit OUs organisasi Anda (.). Jika Anda memiliki beberapa akun, Anda dapat membuat toko kepercayaan sekali dan menggunakannya AWS RAM untuk membuatnya dapat digunakan oleh akun lain. Jika akun Anda dikelola oleh AWS Organizations, Anda dapat berbagi toko kepercayaan dengan semua akun di organisasi atau hanya akun dalam unit organisasi tertentu (OUs).

Dengan AWS RAM, Anda berbagi sumber daya yang Anda miliki dengan membuat pembagian sumber daya. Pembagian sumber daya menentukan sumber daya yang akan dibagikan, dan konsumen yang akan dibagikan. Dalam model ini, Akun AWS yang memiliki toko kepercayaan (pemilik) membaginya dengan yang lain Akun AWS (konsumen). Konsumen dapat mengaitkan toko kepercayaan bersama dengan pendengar Application Load Balancer mereka dengan cara yang sama mereka mengaitkan toko kepercayaan di akun mereka sendiri.

Pemilik toko kepercayaan dapat berbagi toko kepercayaan dengan:

- Khusus Akun AWS di dalam atau di luar organisasinya di AWS Organizations
- Unit organisasi di dalam organisasinya di AWS Organizations
- Seluruh organisasinya di AWS Organizations

### Daftar Isi

- [Prasyarat untuk berbagi toko kepercayaan](#)
- [Izin untuk toko kepercayaan bersama](#)
- [Bagikan toko kepercayaan](#)
- [Berhenti berbagi toko kepercayaan](#)
- [Tagihan dan pengukuran](#)

## Prasyarat untuk berbagi toko kepercayaan

- Anda harus membuat pembagian sumber daya menggunakan AWS Resource Access Manager. Untuk informasi selengkapnya, lihat [Membuat berbagi sumber daya](#) di Panduan AWS RAM Pengguna.
- Untuk berbagi toko kepercayaan, Anda harus memilikinya di toko Anda Akun AWS. Anda tidak dapat berbagi toko kepercayaan yang telah dibagikan dengan Anda.
- Untuk berbagi toko kepercayaan dengan organisasi Anda atau unit organisasi di AWS Organizations, Anda harus mengaktifkan berbagi dengan AWS Organizations. Untuk informasi selengkapnya, lihat [Mengaktifkan Berbagi dengan AWS Organizations](#) di Panduan AWS RAM Pengguna.

## Izin untuk toko kepercayaan bersama

### Pemilik toko kepercayaan

- Pemilik toko kepercayaan dapat membuat toko kepercayaan.
- Pemilik toko kepercayaan dapat menggunakan toko kepercayaan dengan penyeimbang beban di akun yang sama.
- Pemilik toko kepercayaan dapat berbagi toko kepercayaan dengan AWS akun lain atau AWS Organizations.
- Pemilik toko kepercayaan dapat membatalkan berbagi toko kepercayaan dari AWS akun apa pun atau AWS Organizations.
- Pemilik toko kepercayaan tidak dapat mencegah penyeimbang beban menggunakan toko kepercayaan di akun yang sama.
- Pemilik toko kepercayaan dapat mencantumkan semua Application Load Balancer menggunakan toko kepercayaan bersama.
- Pemilik toko kepercayaan dapat menghapus toko kepercayaan jika tidak ada asosiasi saat ini.
- Pemilik toko kepercayaan dapat menghapus asosiasi dengan toko kepercayaan bersama.
- Pemilik toko kepercayaan menerima CloudTrail log saat toko kepercayaan bersama digunakan.

### Konsumen toko kepercayaan

- Konsumen toko kepercayaan dapat melihat toko kepercayaan bersama.

- Konsumen toko kepercayaan dapat membuat atau memodifikasi pendengar menggunakan toko kepercayaan di akun yang sama.
- Konsumen toko kepercayaan dapat membuat atau memodifikasi pendengar menggunakan toko kepercayaan bersama.
- Konsumen toko kepercayaan tidak dapat membuat pendengar menggunakan toko kepercayaan yang tidak lagi dibagikan.
- Konsumen toko kepercayaan tidak dapat memodifikasi toko kepercayaan bersama.
- Konsumen toko kepercayaan dapat melihat ARN toko kepercayaan bersama saat dikaitkan dengan pendengar.
- Konsumen toko kepercayaan menerima CloudTrail log saat membuat atau memodifikasi pendengar menggunakan toko kepercayaan bersama.

## Izin terkelola

Saat berbagi toko kepercayaan, pembagian sumber daya menggunakan izin terkelola untuk mengontrol tindakan mana yang diizinkan oleh konsumen toko kepercayaan. Anda dapat menggunakan izin terkelola `defaultAWSRAMPermissionElasticLoadBalancingTrustStore`, yang mencakup semua izin yang tersedia, atau membuat izin terkelola pelanggan Anda sendiri. `DescribeTrustStoreAssociations` `Izin` `DescribeTrustStores` `DescribeTrustStoreRevocations`, dan selalu diaktifkan dan tidak dapat dihapus.

Izin berikut didukung untuk pembagian sumber daya toko kepercayaan:

`elasticloadbalancing: CreateListener`

Dapat melampirkan toko kepercayaan bersama ke pendengar baru.

`elasticloadbalancing: ModifyListener`

Dapat melampirkan toko kepercayaan bersama ke pendengar yang ada.

`elasticloadbalancing: GetTrustStoreCaCertificatesBundle`

Dapat mengunduh bundel sertifikat ca yang terkait dengan toko kepercayaan bersama.

`elasticloadbalancing: GetTrustStoreRevocationContent`

Dapat mengunduh file pencabutan yang terkait dengan toko kepercayaan bersama.

## elasticloadbalancing: (Default) DescribeTrustStores

Dapat mencantumkan semua toko kepercayaan yang dimiliki dan dibagikan dengan akun.

## elasticloadbalancing: (Default) DescribeTrustStoreRevocations

Dapat mencantumkan semua konten pencabutan untuk toko kepercayaan yang diberikan arn.

## elasticloadbalancing: (Default) DescribeTrustStoreAssociations

Dapat mencantumkan semua sumber daya di akun konsumen toko kepercayaan yang terkait dengan toko kepercayaan bersama.

## Bagikan toko kepercayaan

Untuk berbagi toko kepercayaan, Anda harus menambahkannya ke berbagi sumber daya. Berbagi sumber daya adalah AWS RAM sumber daya yang memungkinkan Anda berbagi sumber daya Akun AWS. Pembagian sumber daya menentukan sumber daya untuk dibagikan, konsumen dengan siapa mereka dibagikan, dan tindakan apa yang dapat dilakukan oleh kepala sekolah. Saat berbagi toko kepercayaan menggunakan konsol Amazon EC2, Anda menambahkannya ke pembagian sumber daya yang ada. Untuk menambahkan toko kepercayaan ke pembagian sumber daya baru, Anda harus terlebih dahulu membuat pembagian sumber daya menggunakan [AWS RAM konsol](#).

Ketika Anda berbagi toko kepercayaan yang Anda miliki dengan orang lain Akun AWS, Anda mengaktifkan akun tersebut untuk mengaitkan pendengar Application Load Balancer mereka dengan toko kepercayaan di akun Anda.

Jika Anda adalah bagian dari organisasi AWS Organizations dan berbagi dalam organisasi Anda diaktifkan, konsumen di organisasi Anda secara otomatis diberikan akses ke toko kepercayaan bersama. Jika tidak, konsumen menerima undangan untuk bergabung dengan pembagian sumber daya dan diberikan akses ke toko kepercayaan bersama setelah menerima undangan.

Anda dapat berbagi toko kepercayaan yang Anda miliki menggunakan konsol, konsol, AWS RAM atau konsol Amazon EC2. AWS CLI

Untuk berbagi toko kepercayaan yang Anda miliki menggunakan konsol Amazon EC2

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Load Balancing, pilih Trust Stores.
3. Pilih nama toko kepercayaan untuk melihat halaman detailnya.

4. Pada tab Berbagi, pilih Bagikan toko kepercayaan.
5. Pada halaman Share trust store, di bawah Pembagian sumber daya, pilih sumber daya mana yang akan dibagikan dengan toko kepercayaan Anda.
6. (Opsional) Jika Anda perlu membuat pembagian sumber daya baru, pilih tautan Buat berbagi sumber daya di konsol RAM.
7. Pilih Bagikan toko kepercayaan.

Untuk berbagi toko kepercayaan yang Anda miliki menggunakan AWS RAM konsol

Lihat [Membuat Sumber Daya Bersama](#) di Panduan Pengguna AWS RAM .

Untuk berbagi toko kepercayaan yang Anda miliki menggunakan AWS CLI

Gunakan perintah [create-resource-share](#).

## Berhenti berbagi toko kepercayaan

Untuk berhenti berbagi toko kepercayaan yang Anda miliki, Anda harus menghapusnya dari pembagian sumber daya. Asosiasi yang ada tetap ada setelah Anda berhenti membagikan toko kepercayaan Anda, namun asosiasi baru ke toko kepercayaan yang sebelumnya dibagikan tidak diizinkan. Ketika pemilik toko kepercayaan atau konsumen toko kepercayaan menghapus asosiasi, itu dihapus dari kedua akun. Jika konsumen toko kepercayaan ingin meninggalkan pembagian sumber daya, mereka harus meminta pemilik pembagian sumber daya untuk menghapus akun.

### Menghapus asosiasi

Pemilik toko kepercayaan dapat secara paksa menghapus asosiasi toko kepercayaan yang ada menggunakan [DeleteTrustStoreAssociation](#) perintah. Ketika asosiasi dihapus, setiap pendengar penyeimbang beban yang menggunakan trust store tidak dapat lagi memverifikasi sertifikat klien dan akan gagal melakukan jabat tangan TLS.

Anda dapat berhenti berbagi toko kepercayaan menggunakan konsol, konsol, AWS RAM atau konsol Amazon EC2. AWS CLI

Untuk berhenti berbagi toko kepercayaan yang Anda miliki menggunakan konsol Amazon EC2

1. Buka konsol Amazon EC2 di. <https://console.aws.amazon.com/ec2/>

2. Pada panel navigasi, di bawah Load Balancing, pilih Trust Stores.
3. Pilih nama toko kepercayaan untuk melihat detailnya.
4. Pada tab Berbagi, di bawah Berbagi sumber daya, pilih pembagian sumber daya untuk berhenti berbagi.
5. Pilih Hapus.

Untuk berhenti berbagi toko kepercayaan yang Anda miliki menggunakan AWS RAM konsol

Lihat [Memperbarui Sumber Daya Bersama](#) di Panduan Pengguna AWS RAM .

Untuk berhenti berbagi toko kepercayaan yang Anda miliki menggunakan AWS CLI

Gunakan perintah [disassociate-resource-share](#).

## Tagihan dan pengukuran

Toko kepercayaan bersama dikenakan tarif toko kepercayaan standar yang sama, ditagih per jam, per asosiasi toko kepercayaan dengan Application Load Balancer.

Untuk informasi selengkapnya, termasuk tarif spesifik per wilayah, lihat harga [Elastic Load Balancing](#)

## Mengautentikasi pengguna menggunakan Application Load Balancer

Anda dapat mengonfigurasi Application Load Balancer untuk mengautentikasi pengguna dengan aman saat mereka mengakses aplikasi Anda. Ini memungkinkan Anda untuk memindahkan pekerjaan mengautentikasi pengguna ke penyeimbang beban Anda sehingga aplikasi Anda dapat fokus pada logika bisnis mereka.

Contoh penggunaan berikut ini didukung:

- Autentikasi pengguna melalui penyedia identitas (IdP) yang sesuai dengan OpenID Connect (OIDC).
- Otentikasi pengguna melalui sosial IdPs, seperti Amazon, Facebook, atau Google, melalui kumpulan pengguna yang didukung oleh Amazon Cognito.
- Mengautentikasi pengguna melalui identitas perusahaan, menggunakan SAMP, OpenID Connect (OIDC), OAuth atau, melalui kumpulan pengguna yang didukung oleh Amazon Cognito.

## Bersiap menggunakan IdP yang sesuai dengan OID

Lakukan hal berikut jika Anda menggunakan IdP yang sesuai dengan OIDC dengan Application Load Balancer Anda:

- Buat aplikasi OIDC baru di IdP Anda. DNS iDP harus dapat diselesaikan secara publik.
- Anda harus mengonfigurasi ID klien dan rahasia klien.
- Dapatkan titik akhir berikut yang diterbitkan oleh IdP: otorisasi, token, dan info pengguna. Anda dapat menemukan informasi ini di konfigurasi.
- Sertifikat endpoint IDP harus dikeluarkan oleh otoritas sertifikat publik terpercaya.
- Entri DNS untuk titik akhir harus dapat diselesaikan secara publik, bahkan jika mereka memutuskan ke alamat IP pribadi.
- Izinkan salah satu pengalihan berikut URLs di aplikasi iDP Anda, mana pun yang akan digunakan pengguna Anda, di mana DNS adalah nama domain penyeimbang beban Anda dan CNAME adalah alias DNS untuk aplikasi Anda:
  - <https://oauth2/idpresponse> *DNS*
  - <https://oauth2/idpresponse> *CNAME*

## Bersiap menggunakan Amazon Cognito

Wilayah Tersedia

Integrasi Amazon Cognito untuk Application Load Balancers tersedia di wilayah berikut:

- Timur AS (N. Virginia)
- AS Timur (Ohio)
- AS Barat (California Utara)
- AS Barat (Oregon)
- Kanada (Pusat)
- Kanada Barat (Calgary)
- Eropa (Stockholm)
- Europe (Milan)
- Eropa (Frankfurt)
- Europe (Zurich)

- Eropa (Irlandia)
- Eropa (London)
- Eropa (Paris)
- Eropa (Spanyol)
- Amerika Selatan (Sao Paulo)
- Asia Pasifik (Hong Kong)
- Asia Pacific (Tokyo)
- Asia Pasifik (Seoul)
- Asia Pasifik (Osaka)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Hyderabad)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Jakarta)
- Asia Pacific (Melbourne)
- Timur Tengah (UAE)
- Timur Tengah (Bahrain)
- Africa (Cape Town)
- Israel (Tel Aviv)

Lakukan hal berikut jika Anda menggunakan kolam pengguna Amazon Cognito dengan Application Load Balancer Anda:

- Membuat pengguna. Untuk informasi lebih lanjut, lihat [Kolam pengguna Amazon Cognito](#) di Panduan Developer Amazon Cognito.
- Membuat klien kolam pengguna. Anda harus mengonfigurasi klien untuk menghasilkan rahasia klien, menggunakan alur hibah kode, dan mendukung OAuth cakupan yang sama dengan yang digunakan penyeimbang beban. Untuk informasi lebih lanjut, lihat [Mengonfigurasi klien aplikasi kolam pengguna](#) di Panduan Developer Amazon Cognito.
- Buat domain kolam pengguna. Untuk informasi selengkapnya, lihat [Mengonfigurasi domain kumpulan pengguna](#) di Panduan Pengembang Amazon Cognito.

- Verifikasi bahwa cakupan yang diminta mengembalikan token ID. Misalnya, cakupan default, `openid` mengembalikan token ID tetapi cakupan `aws.cognito.signin.user.admin` tidak.
- Untuk bergabung dengan IdP sosial atau perusahaan, aktifkan IdP di bagian federasi. Untuk informasi selengkapnya, lihat [Masuk kumpulan pengguna dengan penyedia identitas pihak ketiga](#) di Panduan Pengembang Amazon Cognito.
- Izinkan pengalihan berikut URLs di bidang URL callback untuk Amazon Cognito, di mana DNS adalah nama domain penyeimbang beban Anda, dan CNAME adalah alias DNS untuk aplikasi Anda (jika Anda menggunakannya):
  - `https:///oauth2/idpresponse` *DNS*
  - `https:///oauth2/idpresponse` *CNAME*
- Izinkan domain kolam pengguna Anda di URL panggilan balik aplikasi IdP Anda. Gunakan format untuk IdP Anda. Contoh:
  - `https://domain-prefix.auth.region.amazoncognito.com/saml2/idpresponse`
  - `https:///saml2/idpresponse` *user-pool-domain*

URL callback di setelan klien aplikasi harus menggunakan semua huruf kecil.

Untuk memungkinkan pengguna mengonfigurasi penyeimbang beban agar menggunakan Amazon Cognito untuk mengautentikasi pengguna, Anda harus memberikan izin kepada pengguna untuk memanggil tindakan tersebut. `cognito-idp:DescribeUserPoolClient`

## Bersiaplah untuk menggunakan Amazon CloudFront

Aktifkan pengaturan berikut jika Anda menggunakan CloudFront distribusi di depan Application Load Balancer Anda:

- Header permintaan teruskan (semua) - Memastikan bahwa CloudFront tidak menyimpan respons cache untuk permintaan yang diautentikasi. Ini mencegah respons dilayani dari cache setelah sesi otentikasi berakhir. Atau, untuk mengurangi risiko ini saat caching diaktifkan, pemilik CloudFront distribusi dapat menetapkan nilai time-to-live (TTL) untuk kedaluwarsa sebelum cookie otentikasi berakhir.
- Penerusan string kueri dan caching (semua) — Ensures that the load balancer has access to the query string parameters required to authenticate the user with the IdP.
- Penerusan cookie (semua) — Memastikan bahwa CloudFront meneruskan semua cookie otentikasi ke penyeimbang beban.

- Saat mengonfigurasi otentikasi OpenID Connect (OIDC) bersama CloudFront dengan Amazon, pastikan bahwa port HTTPS 443 digunakan secara konsisten di seluruh jalur koneksi. Jika tidak, kegagalan otentikasi dapat terjadi karena pengalihan OIDC klien URLs tidak cocok dengan nomor port URI yang dihasilkan semula.

## Mengonfigurasi autentikasi pengguna

Anda mengonfigurasi autentikasi pengguna dengan membuat tindakan otentikasi untuk satu atau lebih aturan pendengar. Parameter `authenticate-cognito` dan `authenticate-oidc` jenis tindakan hanya didukung dengan listener HTTPS. Untuk deskripsi bidang terkait, lihat [AuthenticateCognitoActionConfig](#) dan [AuthenticateOidcActionConfig](#) di Referensi API Elastic Load Balancing versi 2015-12-01.

Penyeimbang beban mengirimkan cookie sesi ke klien untuk mempertahankan status autentikasi. Cookie ini selalu berisi atribut `secure`, karena autentikasi pengguna memerlukan listener HTTPS. Cookie ini berisi atribut `SameSite=None` dengan permintaan CORS (berbagi sumber daya lintas-asal).

Untuk penyeimbang beban yang mendukung beberapa aplikasi yang memerlukan otentikasi klien independen, setiap aturan pendengar dengan tindakan otentikasi harus memiliki nama cookie yang unik. Ini memastikan bahwa klien selalu diautentikasi dengan IDP sebelum dirutekan ke grup target yang ditentukan dalam aturan.

Application Load Balancer tidak mendukung nilai cookie yang diencode URL.

Secara default, bidang `SessionTimeout` diatur ke 7 hari. Jika Anda ingin sesi yang lebih singkat, Anda dapat mengonfigurasi batas waktu sesi sesingkat 1 detik. Untuk informasi selengkapnya, lihat [Batas waktu sesi habis](#).

Mengatur `OnUnauthenticatedRequest` bidang yang sesuai untuk aplikasi Anda. Sebagai contoh:

- Aplikasi yang mengharuskan pengguna untuk log in menggunakan identitas sosial atau perusahaan—Ini didukung oleh opsi default, `authenticate`. Jika pengguna tidak log in, penyeimbang beban mengarahkan permintaan ke titik akhir otorisasi IdP dan IdP meminta pengguna untuk masuk menggunakan antarmuka pengguna.
- Aplikasi yang menyediakan tampilan pribadi untuk pengguna yang masuk atau tampilan umum untuk pengguna yang tidak masuk—Untuk mendukung jenis aplikasi ini, gunakan pilihan `allow`. Jika pengguna masuk, penyeimbang beban memberikan klaim pengguna dan aplikasi dapat

memberikan tampilan yang dipersonalisasi. Jika pengguna tidak masuk, penyeimbang beban meneruskan permintaan tanpa klaim pengguna dan aplikasi dapat memberikan tampilan umum.

- Aplikasi satu halaman dengan JavaScript itu dimuat setiap beberapa detik —Jika Anda menggunakan deny opsi, penyeimbang beban mengembalikan kesalahan HTTP 401 Unauthorized ke panggilan AJAX yang tidak memiliki informasi otentikasi. Namun, jika pengguna memiliki informasi autentikasi yang kedaluwarsa, klien akan dialihkan ke titik akhir otorisasi IdP.

Penyeimbang beban harus dapat berkomunikasi dengan titik akhir token IdP (TokenEndpoint) dan titik akhir info pengguna IdP (UserInfoEndpoint). Application Load Balancers hanya mendukung IPv4 saat berkomunikasi dengan endpoint ini. Jika IDP Anda menggunakan alamat publik, pastikan grup keamanan untuk penyeimbang beban Anda dan jaringan untuk VPC ACLs Anda mengizinkan akses ke titik akhir. Saat menggunakan penyeimbang beban internal atau jenis alamat IPdualstack-without-public-ipv4, gateway NAT dapat memungkinkan penyeimbang beban untuk berkomunikasi dengan titik akhir. Untuk informasi lebih lanjut, lihat [dasar-dasar gateway NAT](#) di Panduan Pengguna Amazon VPC.

Gunakan perintah [buat-peraturan](#) berikut untuk mengonfigurasi autentikasi pengguna.

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

Berikut ini adalah contoh file `actions.json` yang menentukan tindakan `authenticate-oidc` dan tindakan `forward`. `AuthenticationRequestExtraParams` memungkinkan Anda meneruskan parameter tambahan ke IdP selama autentikasi. Harap ikuti dokumentasi yang disediakan oleh penyedia identitas Anda untuk menentukan bidang yang didukung

```
[{  
  "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": "abcdefghijklmnopqrstuvwxy123456789",  
    "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
}]

```

Berikut ini adalah contoh file `actions.json` yang menentukan tindakan `authenticate-cognito` dan tindakan `forward`.

```

[
  {
    "Type": "authenticate-cognito",
    "AuthenticateCognitoConfig": {
      "UserPoolArn": "arn:aws:cognito-idp:region-code:account-id:userpool/user-pool-
id",
      "UserPoolClientId": "abcdefghijklmnopqrstuvwxy123456789",
      "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
  }
]

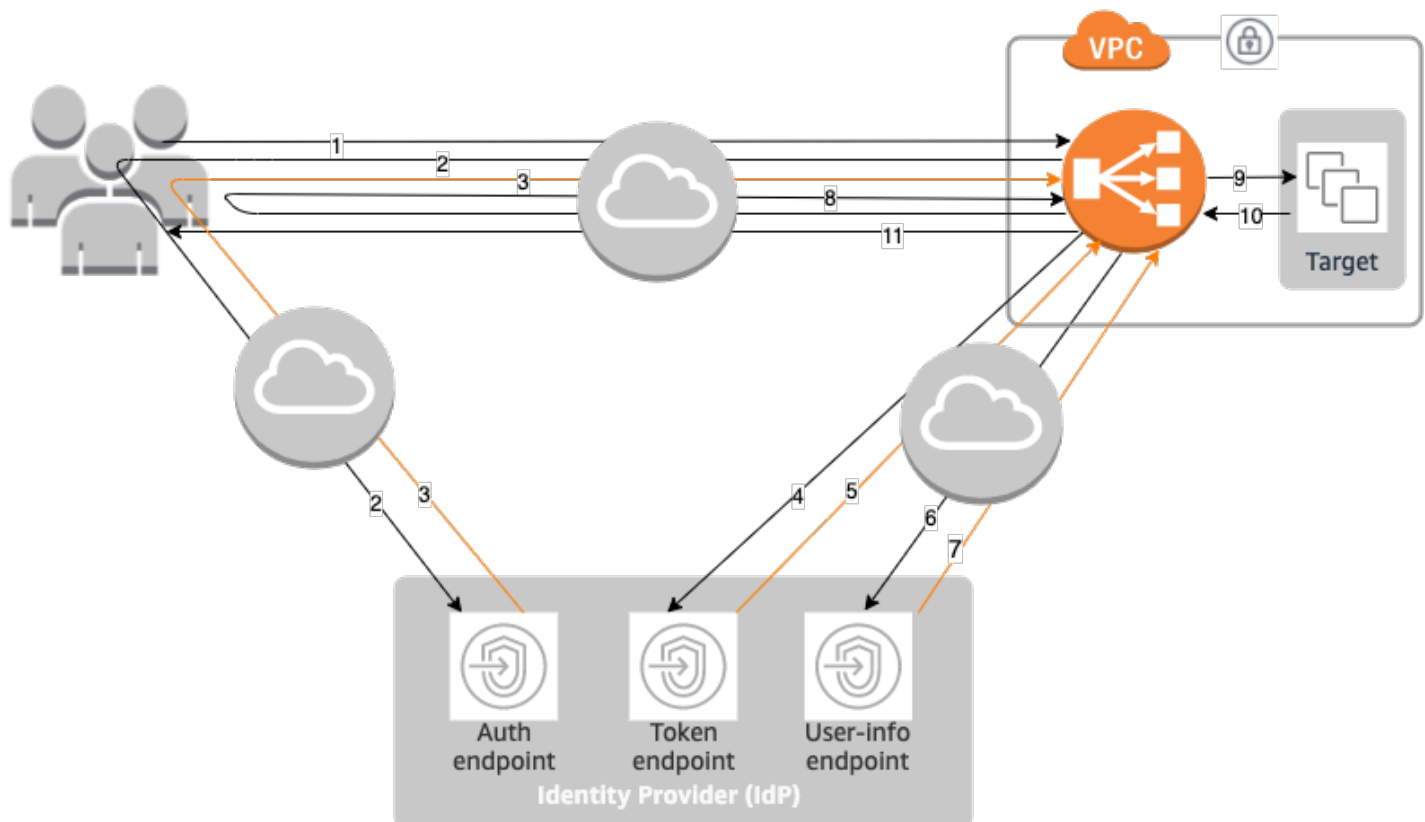
```

}}]

Untuk informasi selengkapnya, lihat [Aturan listener untuk Application Load Balancer Anda](#).

## Alur autentikasi

Diagram jaringan berikut adalah representasi visual tentang bagaimana Application Load Balancer menggunakan OIDC untuk mengautentikasi pengguna.



Item bernomor di bawah ini, menyorot dan menjelaskan elemen yang ditunjukkan dalam diagram jaringan sebelumnya.

1. Pengguna mengirimkan permintaan HTTPS ke situs web yang dihosting di belakang Application Load Balancer. Saat syarat peraturan dengan tindakan autentikasi terpenuhi, penyeimbang beban memeriksa cookie sesi autentikasi di header permintaan.
2. Jika cookie tidak ada, penyeimbang beban mengalihkan pengguna ke titik akhir otorisasi IdP sehingga IdP dapat mengautentikasi pengguna.
3. Setelah pengguna diautentikasi, IdP mengirim pengguna kembali ke penyeimbang beban dengan kode pemberian otorisasi.
4. Penyeimbang beban menyajikan kode pemberian otorisasi ke titik akhir token IdP.

5. Setelah menerima kode pemberian otorisasi yang valid, IdP memberikan token ID dan token akses ke Application Load Balancer.
6. Application Load Balancer kemudian mengirimkan token akses ke titik akhir info pengguna.
7. Titik akhir info pengguna menukar token akses dengan klaim pengguna.
8. Application Load Balancer mengalihkan pengguna dengan cookie sesi AWSELB otentikasi ke URI asli. Karena sebagian besar browser membatasi ukuran cookie hingga 4K, penyeimbang beban membagi cookie yang berukuran lebih besar dari 4K menjadi beberapa cookie. Jika ukuran total klaim pengguna dan token akses yang diterima dari IdP lebih besar dari 11K byte, penyeimbang beban mengembalikan kesalahan HTTP 500 ke klien dan menambah metrik `ELBAuthUserClaimsSizeExceeded`.
9. Application Load Balancer memvalidasi cookie dan meneruskan info pengguna ke target di `X-AMZN-OIDC-*` set header HTTP. Untuk informasi selengkapnya, lihat [Pengkodean klaim pengguna dan verifikasi tanda tangan](#).
10. Target mengirimkan respons kembali ke Application Load Balancer.
11. Application Load Balancer mengirimkan respons akhir kepada pengguna.

Setiap permintaan baru berjalan melalui langkah 1 sampai 11, sementara permintaan berikutnya melalui langkah 9 sampai 11. Artinya, setiap permintaan berikutnya dimulai pada langkah 9 selama cookie belum kedaluwarsa.

`AWSALBAuthNonceCookie` ditambahkan ke header permintaan setelah pengguna mengautentikasi di IDP. Ini tidak mengubah cara Application Load Balancer memproses permintaan pengalihan dari IDP.

Jika IdP menyediakan token penyegaran yang valid dalam token ID, penyeimbang beban akan menyimpan token penyegaran dan menggunakannya untuk menyegarkan klaim pengguna setiap kali token akses kedaluwarsa, hingga waktu sesi habis atau penyegaran IdP gagal. Jika pengguna log out, penyegaran gagal dan penyeimbang beban mengalihkan pengguna ke titik akhir otorisasi IdP. Hal ini memungkinkan penyeimbang beban untuk mengakhiri sesi setelah pengguna log out. Untuk informasi selengkapnya, lihat [Batas waktu sesi habis](#).

#### Note

Kedaluwarsa cookie berbeda dari kedaluwarsa sesi autentikasi. Kedaluwarsa cookie adalah atribut cookie, yang diatur ke 7 hari. Panjang sebenarnya dari sesi autentikasi ditentukan oleh batas waktu sesi yang dikonfigurasi pada Application Load Balancer untuk fitur autentikasi. Waktu habis sesi ini termasuk dalam nilai cookie `Auth`, yang juga dienkrpsi.

## Pengkodean klaim pengguna dan verifikasi tanda tangan

Setelah penyeimbang beban Anda berhasil mengautentikasi pengguna, ia akan mengirimkan klaim pengguna yang diterima dari IdP ke target. Penyeimbang beban menandatangani klaim pengguna sehingga aplikasi dapat memverifikasi tanda tangan dan memverifikasi bahwa klaim dikirim oleh penyeimbang beban.

Penyeimbang beban menambahkan header HTTP berikut:

`x-amzn-oidc-accesstoken`

Token akses dari titik akhir token, dalam teks biasa.

`x-amzn-oidc-identity`

Bidang subjek (sub) dari titik akhir info pengguna, dalam teks biasa.

Catatan: Sub klaim adalah cara terbaik untuk mengidentifikasi pengguna tertentu.

`x-amzn-oidc-data`

Klaim pengguna, dalam format token web JSON (JWT).

Token akses dan klaim pengguna berbeda dari token ID. Token akses dan klaim pengguna hanya mengizinkan akses ke sumber daya server, sementara token ID membawa informasi tambahan untuk mengautentikasi pengguna. Application Load Balancer membuat token akses baru saat mengautentikasi pengguna dan hanya meneruskan token akses dan klaim ke backend, namun tidak meneruskan informasi token ID.

Token ini mengikuti format JWT tetapi bukan ID token. Format JWT mencakup header, payload, dan tanda tangan yang dikodekan URL base64, dan menyertakan karakter padding di bagian akhir. Application Load Balancer menggunakan ES256 (ECDSA menggunakan P-256 dan SHA256) untuk menghasilkan tanda tangan JWT.

Header JWT adalah objek JSON dengan bidang-bidang berikut:

```
{
  "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"  
}
```

Muatan JWT adalah objek JSON yang berisi klaim pengguna yang diterima dari endpoint info pengguna IdP.

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

Jika Anda ingin menyeimbangkan beban mengenkripsi klaim pengguna Anda, Anda harus mengonfigurasi grup target Anda untuk menggunakan HTTPS. Selain itu, sebagai praktik keamanan terbaik, kami sarankan Anda membatasi target Anda untuk hanya menerima lalu lintas dari Application Load Balancer Anda. Anda dapat mencapai ini dengan mengonfigurasi grup keamanan target Anda untuk mereferensikan ID grup keamanan penyeimbang beban.

Untuk memastikan keamanan, Anda harus memverifikasi tanda tangan sebelum melakukan otorisasi berdasarkan klaim dan memvalidasi bahwa `signer` bidang di header JWT berisi ARN Application Load Balancer yang diharapkan.

Untuk mendapatkan kunci publik, dapatkan ID kunci dari header JWT dan gunakan untuk mencari kunci publik dari titik akhir. Titik akhir untuk setiap AWS Wilayah adalah sebagai berikut:

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

Untuk AWS GovCloud (US), titik akhir adalah sebagai berikut:

```
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
```

AWS menyediakan pustaka yang dapat Anda gunakan untuk memverifikasi JWTs ditandatangani oleh Amazon Cognito, Application Load Balancers, dan kompatibel dengan OIDC lainnya. IDPs Untuk informasi selengkapnya, lihat [AWS Verifikasi JWT](#).

## Waktu habis

### Batas waktu sesi habis

Token penyegaran dan batas waktu sesi bekerja bersama sebagai berikut:

- Jika batas waktu sesi lebih pendek dari masa kedaluwarsa token akses, penyeimbang beban menghormati batas waktu sesi. Jika pengguna memiliki sesi aktif dengan IdP, pengguna mungkin tidak diminta untuk log in lagi. Jika tidak, pengguna diarahkan untuk log in.
- Jika batas waktu sesi IdP lebih lama dari batas waktu sesi Application Load Balancer, pengguna tidak perlu memberikan kredensial untuk log in lagi. Sebagai gantinya, IdP mengalihkan kembali ke Application Load Balancer dengan kode pemberian otorisasi baru. Kode otorisasi adalah penggunaan tunggal, bahkan jika tidak ada login ulang.
- Jika batas waktu sesi IdP sama dengan atau lebih pendek dari batas waktu sesi Application Load Balancer, pengguna akan diminta untuk memberikan kredensial untuk log in lagi. Setelah pengguna masuk, IdP mengalihkan kembali ke Application Load Balancer dengan kode pemberian otorisasi baru, dan alur autentikasi lainnya berlanjut hingga permintaan mencapai backend.
- Jika batas waktu sesi lebih lama dari masa berlaku token akses dan IdP tidak mendukung token penyegaran, penyeimbang beban akan mempertahankan sesi autentikasi hingga waktu habis. Kemudian, sesi akan meminta pengguna log in lagi.
- Jika batas waktu sesi lebih lama dari masa berlaku token akses dan IdP mendukung token penyegaran, penyeimbang beban akan menyegarkan sesi pengguna setiap kali token akses kedaluwarsa. Penyeimbang beban meminta pengguna log in lagi hanya setelah sesi autentikasi habis atau aliran penyegaran gagal.

### Batas waktu login klien

Klien harus memulai dan menyelesaikan proses otentikasi dalam waktu 15 menit. Jika klien gagal menyelesaikan otentikasi dalam batas 15 menit, ia menerima kesalahan HTTP 401 dari penyeimbang beban. Batas waktu ini tidak dapat diubah atau dihapus.

Misalnya, jika pengguna memuat halaman login melalui Application Load Balancer, mereka harus menyelesaikan proses login dalam waktu 15 menit. Jika pengguna menunggu dan kemudian mencoba masuk setelah batas waktu 15 menit berakhir, penyeimbang beban mengembalikan kesalahan HTTP 401. Pengguna harus me-refresh halaman dan mencoba masuk lagi.

## Logout autentikasi

Saat aplikasi perlu me-logout pengguna yang diautentikasi, aplikasi harus menyetel waktu kedaluwarsa cookie sesi autentikasi ke -1 dan mengarahkan klien ke titik akhir logout IdP (jika IdP mendukungnya). Untuk mencegah pengguna menggunakan kembali cookie yang dihapus, kami menyarankan Anda untuk mengonfigurasi sesingkat waktu kedaluwarsa untuk token akses yang wajar. Jika klien menyediakan penyeimbang beban dengan cookie sesi yang memiliki token akses kedaluwarsa dengan token penyegaran non-Null, penyeimbang beban akan menghubungi IDP untuk menentukan apakah pengguna masih login.

Halaman arahan logout klien tidak diautentikasi. Ini berarti bahwa mereka tidak dapat berada di belakang aturan Application Load Balancer yang memerlukan otentikasi.

- Ketika permintaan dikirim ke target, aplikasi harus mengatur kedaluwarsa ke -1 untuk semua cookie autentikasi. Application Load Balancer mendukung cookie hingga ukuran 16K dan karenanya dapat membuat hingga 4 pecahan untuk dikirim ke klien.
  - Jika IdP memiliki titik akhir logout, IdP harus mengeluarkan pengalihan ke titik akhir logout IdP, misalnya, [Titik Akhir LOGOUT](#) yang terdokumentasi di Panduan Developer Amazon Cognito.
  - Jika IdP tidak memiliki titik akhir logout, permintaan akan kembali ke halaman arahan logout klien, dan proses login dimulai ulang.
- Dengan asumsi bahwa IdP memiliki titik akhir logout, IdP harus kedaluwarsa token akses dan token penyegaran, dan mengarahkan pengguna kembali ke halaman arahan logout klien.
- Permintaan berikutnya mengikuti alur autentikasi asli.

## Verifikasi JWTs menggunakan Application Load Balancer

Anda dapat mengonfigurasi Application Load Balancer (ALB) untuk memverifikasi JSON Web Tokens (JWT) yang disediakan oleh klien untuk komunikasi aman (S2S) atau service-to-service (M2M). machine-to-machine Penyeimbang beban dapat memverifikasi JWT tidak peduli bagaimana itu dikeluarkan dan tanpa interaksi manusia.

ALB akan memvalidasi tanda tangan token dan membutuhkan dua klaim wajib: 'iss' (penerbit) dan 'exp' (kedaluwarsa). Selain itu, jika ada dalam token, ALB juga akan memvalidasi klaim 'nbf' (bukan sebelumnya) dan 'iat' (dikeluarkan pada waktu). Anda dapat mengonfigurasi hingga 10 klaim tambahan untuk validasi. Klaim ini mendukung tiga format:

- Single-string: Nilai teks tunggal

- Nilai yang dipisahkan spasi: Beberapa nilai dipisahkan oleh spasi (maksimum 10 nilai)
- String-array: Sebuah array nilai teks (maksimum 10 nilai)

Jika token valid, penyeimbang beban meneruskan permintaan dengan token seperti target. Kalau tidak, itu menolak permintaan.

## Bersiaplah untuk menggunakan verifikasi JWT

Lakukan hal-hal berikut:

1. Daftarkan layanan Anda dengan IDP, yang mengeluarkan ID klien dan rahasia klien.
2. Buat panggilan terpisah ke IDP untuk meminta akses ke layanan. IDP merespons dengan token akses. Token ini biasanya merupakan JWT yang ditandatangani oleh IDP.
3. Siapkan titik akhir JSON Web Key Sets (JWKS). Load balancer memperoleh kunci publik yang diterbitkan oleh IDP di lokasi terkenal yang Anda konfigurasi.
4. Sertakan JWT dalam header permintaan, dan teruskan ke Application Load Balancer di setiap permintaan. Catatan: Hanya RS256 algoritma yang didukung

## Batas validasi JWT

Saat menggunakan validasi JWT dengan Application Load Balancer Anda, titik akhir JWKS (JSON Web Key Set) harus memenuhi persyaratan berikut:

- Ukuran respons maksimal: 150 KB
- Jumlah maksimum tombol: 10 tombol

Jika respons JWKS dari penyedia identitas Anda melebihi salah satu dari batasan ini, Application Load Balancer tidak akan meneruskan permintaan ke target backend Anda.

Jika titik akhir JWKS penyedia identitas Anda melebihi batas ini, pertimbangkan untuk menerapkan validasi JWT dalam kode aplikasi Anda atau menggunakan penyedia identitas dengan set kunci yang lebih kecil.

Untuk mengkonfigurasi verifikasi JWT menggunakan konsol

1. Buka konsol konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah PENYEIMBANGAN BEBAN, pilih Penyeimbang beban.

3. Pilih Application Load Balancer Anda dan pilih tab Listeners.
4. Pilih pendengar HTTPS dan pilih Kelola aturan.
5. Pilih Tambahkan aturan.
6. (Opsional) Untuk menentukan nama aturan Anda, perluas Nama dan tag, dan masukkan nama. Untuk menambahkan tag tambahan, pilih Tambahkan tag tambahan dan masukkan kunci tag dan nilai tag.
7. Di bawah Kondisi, tentukan 1-5 nilai kondisi
8. (Opsional) Untuk menambahkan transformasi, pilih Tambahkan transformasi, pilih jenis transformasi, dan masukkan ekspresi reguler untuk mencocokkan dan string pengganti.
9. Untuk Tindakan, Tindakan pra-perutean, pilih Validasi token.
  - a. Untuk titik akhir JWKS, masukkan URL titik akhir Set Kunci Web JSON Anda. Titik akhir ini harus dapat diakses publik dan mengembalikan kunci publik yang digunakan untuk memverifikasi tanda tangan JWT.
  - b. Untuk Penerbit, masukkan nilai yang diharapkan dari klaim iss di token JWT Anda.
  - c. (Opsional) Untuk memvalidasi klaim tambahan, pilih Klaim tambahan.
    - i. Untuk nama Klaim, masukkan nama klaim untuk memvalidasi.
    - ii. Untuk Format, pilih bagaimana nilai klaim harus ditafsirkan:
      1. String tunggal: Klaim harus sama persis dengan satu nilai yang ditentukan.
      2. String array: Klaim harus cocok dengan salah satu nilai dalam array.
      3. Nilai yang dipisahkan spasi: Klaim berisi nilai yang dipisahkan spasi yang harus menyertakan nilai yang ditentukan.
    - iii. Untuk Nilai, masukkan nilai yang diharapkan untuk klaim.
    - iv. Ulangi untuk klaim tambahan (maksimal 10 klaim).
10. Untuk Tindakan, tindakan Perutean, pilih tindakan utama (Teruskan ke, Alihkan ke, atau Kembalikan respons tetap) yang harus dilakukan setelah validasi token berhasil.
11. Konfigurasi tindakan utama sesuai kebutuhan
12. Pilih Simpan.

## Untuk mengkonfigurasi verifikasi JWT menggunakan CLI

Gunakan perintah [create-rule](#) berikut untuk mengkonfigurasi verifikasi JWT.

Buat aturan pendengar dengan tindakan untuk memverifikasi JWTs. Pendengar harus menjadi pendengar HTTPS.

### Note

Saat mengonfigurasi validasi JWT, pastikan respons titik akhir JWKS Anda tidak melebihi 150 KB atau berisi lebih dari 10 kunci. Tanggapan yang melebihi batas ini akan mencegah penerusan permintaan ke target Anda.

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

Berikut ini adalah contoh `actions.json` file yang menentukan `jwt-validation` tindakan dan `forward` tindakan. Harap ikuti dokumentasi yang disediakan oleh penyedia identitas Anda untuk menentukan bidang yang didukung

```
--actions '[
  {
    "Type":"jwt-validation",
    "JwtValidationConfig":{
      "JwksEndpoint":"https://issuer.example.com/.well-known/jwks.json",
      "Issuer":"https://issuer.com"
    },
    "Order":1
  },
  {
    "Type":"forward",
    "TargetGroupArn":"target-group-arn",
    "Order":2
  }
]'
```

Contoh berikut menentukan klaim tambahan untuk memvalidasi.

```
--actions '[
  {
```

```
"Type": "jwt-validation",
"JwtValidationConfig": {
  "JwksEndpoint": "https://issuer.example.com/.well-known/jwks.json",
  "Issuer": "https://issuer.com",
  "AdditionalClaims": [
    {
      "Format": "string-array",
      "Name": "claim_name",
      "Values": ["value1", "value2"]
    }
  ],
},
"Order": 1
},
{
  "Type": "forward",
  "TargetGroupArn": "target-group-arn",
  "Order": 2
}
]'
```

Untuk informasi selengkapnya, lihat [the section called “Aturan pendengar”](#).

## Header HTTP dan Application Load Balancer

Permintaan HTTP dan respons HTTP menggunakan bidang header untuk mengirim informasi tentang pesan HTTP. Header HTTP ditambahkan secara otomatis. Bidang header adalah pasangan nama-nilai yang dipisahkan titik dua yang dipisahkan oleh carriage return (CR) dan line feed (LF). Satu set standar bidang header HTTP didefinisikan dalam RFC 2616, [Header Pesan](#). Ada juga header HTTP non-standar yang tersedia secara otomatis ditambahkan dan digunakan secara luas oleh aplikasi. Beberapa header HTTP non-standar memiliki awalan X-Forwarded. Application Load Balancer mendukung header X-Forwarded berikut.

Untuk informasi lebih lanjut tentang koneksi HTTP, lihat [Permintaan perutean](#) di Panduan Pengguna Elastic Load Balancing.

### Header X-Diteruskan

- [X-Diteruskan-Untuk](#)
- [X-Diteruskan-Proto](#)
- [Port-X-Diteruskan](#)

## X-Diteruskan-Untuk

Header `X-Forwarded-For` permintaan membantu Anda mengidentifikasi alamat IP klien saat Anda menggunakan penyeimbang beban HTTP atau HTTPS. Karena penyeimbang beban mencegat lalu lintas antara klien dan server, log akses server Anda hanya berisi alamat IP penyeimbang beban. Untuk melihat alamat IP klien, gunakan `routing.http.xff_header_processing.mode` atribut. Atribut ini memungkinkan Anda untuk memodifikasi, mempertahankan, atau menghapus `X-Forwarded-For` header dalam permintaan HTTP sebelum Application Load Balancer mengirimkan permintaan ke target. Nilai yang mungkin untuk atribut ini adalah `append`, `preserve`, dan `remove`. Nilai default untuk atribut ini adalah `append`.

### Important

`X-Forwarded-For` Header harus digunakan dengan hati-hati karena potensi risiko keamanan. Entri hanya dapat dianggap dapat dipercaya jika ditambahkan oleh sistem yang diamankan dengan benar dalam jaringan.

### Mode pemrosesan

- [Menambahkan](#)
- [Pertahankan](#)
- [Menghapus](#)

### Menambahkan

Secara default, Application Load Balancer menyimpan alamat IP klien di header `X-Forwarded-For` permintaan dan meneruskan header ke server Anda. Jika header `X-Forwarded-For` permintaan tidak disertakan dalam permintaan asli, penyeimbang beban membuat satu dengan alamat IP klien sebagai nilai permintaan. Jika tidak, penyeimbang beban menambahkan alamat IP klien ke header yang ada dan kemudian meneruskan header ke server Anda. Header permintaan `X-Forwarded-For` mungkin berisi beberapa alamat IP yang dipisahkan koma.

Header permintaan `X-Forwarded-For` memiliki bentuk berikut:

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

Berikut adalah contoh header permintaan X-Forwarded-For untuk klien dengan alamat IP 203.0.113.7.

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

Berikut ini adalah contoh header X-Forwarded-For permintaan untuk klien dengan IPv6 alamat 2001:DB8::21f:5bff:febf:ce22:8a2e.

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

Ketika atribut pelestarian port klien (`routing.http.xff_client_port.enabled`) diaktifkan pada penyeimbang beban, header X-Forwarded-For permintaan menyertakan yang `client-port-number` ditambahkan ke `client-ip-address`, dipisahkan oleh titik dua. Header kemudian mengambil bentuk berikut:

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

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

Untuk IPv6, perhatikan bahwa ketika penyeimbang beban menambahkan `client-ip-address` ke header yang ada, itu melampirkan alamat dalam tanda kurung siku.

Berikut ini adalah contoh header X-Forwarded-For permintaan untuk klien dengan IPv4 alamat 12.34.56.78 dan nomor port 8080.

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

Berikut ini adalah contoh header X-Forwarded-For permintaan untuk klien dengan IPv6 alamat 2001:db8:85a3:8d3:1319:8a2e:370:7348 dan nomor port 8080.

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

## Pertahankan

`preserveMode` dalam atribut memastikan bahwa X-Forwarded-For header dalam permintaan HTTP tidak dimodifikasi dengan cara apa pun sebelum dikirim ke target.

## Menghapus

`removeMode` dalam atribut menghapus `X-Forwarded-For` header dalam permintaan HTTP sebelum dikirim ke target.

Jika Anda mengaktifkan atribut pelestarian port klien (`routing.http.xff_client_port.enabled`), dan juga memilih `preserve` atau `remove` untuk `routing.http.xff_header_processing.mode` atribut, Application Load Balancer akan mengganti atribut pelestarian port klien. Itu membuat `X-Forwarded-For` header tidak berubah, atau menghapusnya tergantung pada mode yang Anda pilih, sebelum mengirimkannya ke target.

Tabel berikut menunjukkan contoh `X-Forwarded-For` header yang diterima target ketika Anda memilih salah satu `append`, `preserve` atau `remove` mode. Dalam contoh ini, alamat IP dari hop terakhir adalah `127.0.0.1`.

Minta deskripsi	Contoh permintaan	append	preserve	remove
Permintaan dikirim tanpa header XFF	GET / index.ht ml HTTP/1.1 Host: example.com	X-Forward ed-For: 127.0.0.1	Tidak hadir	Tidak hadir
Permintaan dikirim dengan header XFF dan alamat IP klien.	GET / index.ht ml HTTP/1.1 Host: example.com X-Forward ed-For: 127.0.0.4	X-Forward ed-For: 127.0.0.4, 127.0.0.1	X-Forward ed-For: 127.0.0.4	Tidak hadir
Permintaan dikirim dengan header XFF dengan	GET / index.ht ml HTTP/1.1 Host: example.com	X-Forward ed-For: 127.0.0.4, 127.0.0.8, 127.0.0.1	X-Forward ed-For: 127.0.0.4, 127.0.0.8	Tidak hadir

Minta deskripsi	Contoh permintaan	append	preserve	remove
beberapa alamat IP klien.	X-Forwarded-For: 127.0.0.4, 127.0.0.8			

## Console

Untuk mengelola X-Forwarded-For header

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.
4. Pada tab Atribut, pilih Edit.
5. Di bagian Traffic configuration, di bawah Packet handling, untuk X-Forwarded-For header, pilih Append (default), Preserve, or Remove.
6. Pilih Simpan perubahan.

## AWS CLI

Untuk mengelola X-Forwarded-For header

Gunakan [modify-load-balancer-attributes](#) perintah dengan `routing.http.xff_header_processing.mode` atribut. Nilai yang mungkin adalah `append`, `preserve`, dan `remove`. Nilai default-nya `append`.

```
aws elbv2 modify-load-balancer-attributes \
  --load-balancer-arn load-balancer-arn \
  --attributes "Key=routing.http.xff_header_processing.mode,Value=preserve"
```

## CloudFormation

Untuk mengelola X-Forwarded-For header

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya untuk menyertakan `routing.http.xff_header_processing.mode` atribut. Nilai yang mungkin adalah `append`, `preserve`, dan `remove`. Nilai default-nya `append`.

```
Resources:
  myLoadBalancer:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
    Properties:
      Name: my-alb
      Type: application
      Scheme: internal
      Subnets:
        - !Ref subnet-AZ1
        - !Ref subnet-AZ2
      SecurityGroups:
        - !Ref mySecurityGroup
      LoadBalancerAttributes:
        - Key: "routing.http.xff_header_processing.mode"
          Value: "preserve"
```

## X-Diteruskan-Proto

Header permintaan `X-Forwarded-Proto` membantu Anda mengidentifikasi protokol (HTTP atau HTTPS) yang digunakan klien untuk terhubung ke penyeimbang beban Anda. Log akses server Anda hanya berisi protokol yang digunakan antara server dan penyeimbang beban; mereka tidak berisi informasi tentang protokol yang digunakan antara klien dan penyeimbang beban. Untuk menentukan protokol yang digunakan antara klien dan penyeimbang beban, gunakan header permintaan `X-Forwarded-Proto`. Elastic Load Balancing menyimpan protokol yang digunakan antara klien dan penyeimbang beban di header permintaan `X-Forwarded-Proto` dan meneruskan header dan meneruskan tajuk ke server Anda ke server Anda.

Aplikasi atau situs web Anda dapat menggunakan protokol yang tersimpan di header permintaan `X-Forwarded-Proto` untuk membuat respons yang mengarahkan ke URL yang sesuai.

Header permintaan `X-Forwarded-Proto` mengambil bentuk berikut:

```
X-Forwarded-Proto: originatingProtocol
```

Contoh berikut berisi header permintaan `X-Forwarded-Proto` untuk permintaan yang berasal dari klien sebagai permintaan HTTPS:

```
X-Forwarded-Proto: https
```

## Port-X-Diteruskan

Header permintaan `X-Forwarded-Port` membantu Anda mengidentifikasi port tujuan yang digunakan klien untuk menyambung ke penyeimbang beban.

## Modifikasi header HTTP untuk Application Load Balancer

Modifikasi header HTTP didukung oleh Application Load Balancers, untuk header permintaan dan respons. Tanpa harus memperbarui kode aplikasi Anda, modifikasi header memungkinkan Anda lebih banyak kontrol atas lalu lintas dan keamanan aplikasi Anda.

Untuk mengaktifkan modifikasi header, lihat [Aktifkan modifikasi header](#).

## Ganti nama header mTLS/TLS

Kemampuan penggantian nama header memungkinkan Anda mengonfigurasi nama header mTLS dan TLS yang dihasilkan dan ditambahkan oleh Application Load Balancer ke permintaan.

Kemampuan untuk memodifikasi header HTTP ini memungkinkan Application Load Balancer Anda untuk dengan mudah mendukung aplikasi yang menggunakan header permintaan dan respons yang diformat secara khusus.

Header	Deskripsi
<code>X-Amzn-Mtls-Clientcert-Serial-Number</code>	Memastikan bahwa target dapat mengidentifikasi dan memverifikasi sertifikat spesifik yang disajikan oleh klien selama jabat tangan TLS.
<code>X-Amzn-Mtls-Clientcert-Issuer</code>	Membantu target memvalidasi dan mengotentikasi sertifikat klien dengan mengidentifikasi otoritas sertifikat yang mengeluarkan sertifikat.
<code>X-Amzn-Mtls-Clientcert-Subject</code>	Memberikan target informasi terperinci tentang entitas yang dikeluarkan sertifikat klien, yang membantu dalam identifikasi, otentikasi,

Header	Deskripsi
	otorisasi, dan pencatatan selama otentikasi mTLS.
X-Amzn-Mtls-Clientcert-Validity	Memungkinkan target untuk memverifikasi bahwa sertifikat klien yang digunakan berada dalam periode validitas yang ditentukan, memastikan sertifikat tidak kedaluwarsa atau digunakan sebelum waktunya.
X-Amzn-Mtls-Clientcert-Leaf	Menyediakan sertifikat klien yang digunakan dalam jabatan tangan mTLS, memungkinkan server untuk mengotentikasi klien dan memvalidasi rantai sertifikat. Ini memastikan koneksi aman dan resmi.
X-Amzn-Mtls-Clientcert	Membawa sertifikat klien lengkap. Memungkinkan target untuk memverifikasi keaslian sertifikat, memvalidasi rantai sertifikat, dan mengautentikasi klien selama proses jabatan tangan mTLS.
X-Amzn-TLS-Version	Menunjukkan versi protokol TLS yang digunakan untuk koneksi. Ini memfasilitasi menentukan tingkat keamanan komunikasi, memecahkan masalah koneksi dan memastikan kepatuhan.
X-Amzn-TLS-Cipher-Suite	Menunjukkan kombinasi algoritma kriptografi yang digunakan untuk mengamankan koneksi di TLS. Ini memungkinkan server untuk menilai keamanan koneksi, membantu pemecahan masalah kompatibilitas, dan memastikan kepatuhan terhadap kebijakan keamanan.

## Tambahkan header respons

Dengan menggunakan header insert, Anda dapat mengonfigurasi Application Load Balancer untuk menambahkan header terkait keamanan ke respons. Dengan atribut ini, Anda dapat menyisipkan header termasuk HSTS, CORS, dan CSP.

Secara default, header ini kosong. Ketika ini terjadi, Application Load Balancer tidak mengubah header respons ini.

Saat Anda mengaktifkan header respons, Application Load Balancer menambahkan header dengan nilai yang dikonfigurasi ke semua respons. Jika respons dari target menyertakan header respons HTTP, penyeimbang beban memperbarui nilai header menjadi nilai yang dikonfigurasi. Jika tidak, penyeimbang beban menambahkan header respons HTTP ke respons dengan nilai yang dikonfigurasi.

Header	Deskripsi
Strict-Transport-Security	Menerapkan koneksi HTTPS-only oleh browser untuk durasi tertentu, membantu melindungi terhadap man-in-the-middle serangan, penurunan protokol, dan kesalahan pengguna. memastikan semua komunikasi antara klien dan target dienkripsi.
Access-Control-Allow-Origin	Mengontrol apakah sumber daya pada target dapat diakses dari asal yang berbeda. Ini memungkinkan interaksi lintas asal yang aman sekaligus mencegah akses yang tidak sah.
Access-Control-Allow-Methods	Menentukan metode HTTP yang diizinkan saat membuat permintaan lintas asal ke target. Ini memberikan kontrol atas tindakan mana yang dapat dilakukan dari asal yang berbeda.
Access-Control-Allow-Headers	Menentukan header kustom atau non-sederhana yang dapat disertakan dalam permintaan lintas asal. Header ini memberikan target

Header	Deskripsi
	kontrol atas header mana yang dapat dikirim oleh klien dari asal yang berbeda.
Access-Control-Allow-Credentials	Menentukan apakah klien harus menyertakan kredensial seperti cookie, otentikasi HTTP atau sertifikat klien dalam permintaan lintas asal.
Access-Control-Expose-Headers	Memungkinkan target untuk menentukan header respons tambahan mana yang dapat diakses oleh klien dalam permintaan lintas asal.
Access-Control-Max-Age	Mendefinisikan berapa lama browser dapat men-cache hasil permintaan preflight, mengurangi kebutuhan untuk pemeriksaan preflight berulang. Ini membantu mengoptimalkan kinerja dengan mengurangi jumlah permintaan OPTIONS yang diperlukan untuk permintaan lintas asal tertentu.
Content-Security-Policy	Fitur keamanan yang mencegah serangan injeksi kode seperti XSS dengan mengontrol sumber daya seperti skrip, gaya, gambar, dll. Dapat dimuat dan dieksekusi oleh situs web.
X-Content-Type-Options	Dengan arahan no-sniff, meningkatkan keamanan web dengan mencegah browser menebak jenis sumber daya MIME. Ini memastikan bahwa browser hanya menafsirkan konten sesuai dengan Content-Type yang dideklarasikan

Header	Deskripsi
X-Frame-Options	Mekanisme keamanan header yang membantu mencegah serangan click-jacking dengan mengontrol apakah halaman web dapat disematkan dalam bingkai. Nilai seperti DENY dan SAMEORIGIN dapat memastikan bahwa konten tidak disematkan di situs web berbahaya atau tidak tepercaya.

## Nonaktifkan header

Menggunakan header nonaktifkan, Anda dapat mengonfigurasi Application Load Balancer Anda untuk menonaktifkan header `server:awseb/2.0` dari tanggapan. Ini mengurangi paparan informasi spesifik server, sambil menambahkan lapisan perlindungan ekstra ke aplikasi Anda.

Nama atributnya adalah `routing.http.response.server.enabled`. Nilai yang tersedia adalah `true` atau `false`. Nilai default-nya adalah `true`.

## Batasan

- Nilai header dapat berisi karakter berikut
  - Karakter alfanumerik: `a-z`, dan `A-Z 0-9`
  - Karakter khusus: `_ ; . , \ / ' ? ! ( ) { } [ ] @ < > = - + * # & ` | ~ ^ %`
- Nilai untuk atribut tidak dapat melebihi 1K byte dalam ukuran.
- Elastic Load Balancing melakukan validasi input dasar untuk memverifikasi nilai header valid. Namun validasi tidak dapat mengkonfirmasi apakah nilai didukung untuk header tertentu.
- Menyetel nilai kosong untuk atribut apa pun akan menyebabkan Application Load Balancer kembali ke perilaku default.

## Aktifkan modifikasi header HTTP untuk Application Load Balancer

Modifikasi header dimatikan secara default dan harus diaktifkan pada setiap pendengar. Untuk informasi selengkapnya, lihat [Modifikasi header HTTP](#).

## Console

Untuk mengaktifkan modifikasi header

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih Application Load Balancer.
4. Pada tab Listeners and rules, pilih protokol dan port untuk membuka halaman detail untuk listener Anda.
5. Pada tab Atribut, pilih Edit.

Atribut pendengar diatur ke dalam kelompok. Anda akan memilih fitur mana yang akan diaktifkan.

6. [Pendengar HTTPS] Nama header yang dapat dimodifikasi mTLS/TLS
  - a. Perluas nama mTLS/TLS header yang dapat dimodifikasi.
  - b. Aktifkan header permintaan untuk memodifikasi dan memberikan nama untuk mereka. Untuk informasi selengkapnya, lihat [the section called “Ganti nama header mTLS/TLS”](#).
7. Tambahkan header respons
  - a. Perluas Tambahkan header respons.
  - b. Aktifkan header respons untuk menambahkan dan memberikan nilai bagi mereka. Untuk informasi selengkapnya, lihat [the section called “Tambahkan header respons”](#).
8. Header respons server ALB
  - Aktifkan atau nonaktifkan header Server.
9. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaktifkan modifikasi header

Gunakan perintah [modify-listener-attributes](#). Untuk daftar atribut, lihat [the section called “Atribut modifikasi header”](#).

```
aws elbv2 modify-listener-attributes \  
  --listener-arn listener-arn \  
  --
```

```
--attributes "Key=attribute-name,Value=attribute-value"
```

## CloudFormation

Untuk mengaktifkan modifikasi header

Perbarui [AWS::ElasticLoadBalancingV2::Listener](#) sumber daya untuk menyertakan atribut. Untuk daftar atribut, lihat [the section called "Atribut modifikasi header"](#).

```
Resources:
  myHTTPListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      LoadBalancerArn: !Ref myLoadBalancer
      Protocol: HTTP
      Port: 80
      DefaultActions:
        - Type: "forward"
          TargetGroupArn: !Ref myTargetGroup
      ListenerAttributes:
        - Key: "attribute-name"
          Value: "attribute-value"
```

## Atribut modifikasi header

Berikut ini adalah atribut modifikasi header yang didukung oleh Application Load Balancers.

```
routing.http.request.x_amzn_mtls_clientcert_serial_number.header_name
```

Ubah nama header X-Amzn-Mtls-Clientcert-Serial-Number.

```
routing.http.request.x_amzn_mtls_clientcert_issuer.header_name
```

Ubah nama header X-Amzn-Mtls-Clientcert-Issuer.

```
routing.http.request.x_amzn_mtls_clientcert_subject.header_name
```

Ubah nama header X-Amzn-Mtls-Clientcert-Subject.

```
routing.http.request.x_amzn_mtls_clientcert_validity.header_name
```

Ubah nama header X-Amzn-Mtls-Clientcert-Validity.

```
routing.http.request.x_amzn_mtls_clientcert_leaf.header_name
```

Ubah nama header X-Amzn-Mtls-Clientcert-Leaf.

```
routing.http.request.x_amzn_mtls_clientcert.header_name
```

Ubah nama header X-Amzn-Mtls-Clientcert.

```
routing.http.request.x_amzn_tls_version.header_name
```

Ubah nama header X-Amzn-Tls-Version.

```
routing.http.request.x_amzn_tls_cipher_suite.header_name
```

Ubah nama header X-Amzn-Tls-Cipher-Suite.

```
routing.http.response.server.enabled
```

Menunjukkan apakah akan mengizinkan atau menghapus header server respons HTTP.

```
routing.http.response.strict_transport_security.header_value
```

Tambahkan header Strict-Transport-Security untuk menginformasikan browser bahwa situs hanya boleh diakses menggunakan HTTPS, dan bahwa setiap upaya future untuk mengaksesnya menggunakan HTTP harus secara otomatis dikonversi ke HTTPS.

```
routing.http.response.access_control_allow_origin.header_value
```

Tambahkan header Access-Control-Allow-Origin untuk menentukan asal mana yang diizinkan untuk mengakses server.

```
routing.http.response.access_control_allow_methods.header_value
```

Tambahkan header Access-Control-Allow-Methods untuk menentukan metode HTTP mana yang diizinkan saat mengakses server dari asal yang berbeda.

```
routing.http.response.access_control_allow_headers.header_value
```

Tambahkan header Access-Control-Allow-Headers untuk menentukan header mana yang diizinkan selama permintaan lintas asal.

```
routing.http.response.access_control_allow_credentials.header_value
```

Tambahkan header Access-Control-Allow-Credentials untuk menunjukkan apakah browser harus menyertakan kredensial seperti cookie atau otentikasi dalam permintaan lintas asal.

```
routing.http.response.access_control_expose_headers.header_value
```

Tambahkan header Access-Control-Expose-Headers untuk menunjukkan header mana yang dapat diekspos browser ke klien yang meminta.

```
routing.http.response.access_control_max_age.header_value
```

Tambahkan header Access-Control-Max-Age untuk menentukan berapa lama hasil permintaan preflight dapat di-cache, dalam hitungan detik.

```
routing.http.response.content_security_policy.header_value
```

Tambahkan header Content-Security-Policy untuk menentukan pembatasan yang diberlakukan oleh browser untuk membantu meminimalkan risiko jenis ancaman keamanan tertentu.

```
routing.http.response.x_content_type_options.header_value
```

Tambahkan header X-Content-Type-Options untuk menunjukkan apakah tipe MIME yang diiklankan di header Content-Type harus diikuti dan tidak diubah.

```
routing.http.response.x_frame_options.header_value
```

Tambahkan header X-Frame-Options untuk menunjukkan apakah browser diizinkan untuk merender halaman dalam bingkai, iframe, embed, atau objek.

## Menghapus listener untuk Application Load Balancer Anda

Sebelum Anda menghapus listener, pertimbangkan dampaknya pada aplikasi Anda:

- Penyeimbang beban segera berhenti menerima koneksi baru di port pendengar.
- Koneksi aktif ditutup. Permintaan apa pun yang sedang berlangsung saat pendengar dihapus kemungkinan akan gagal.

### Console

Untuk menghapus pendengar

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban.

4. Pada tab Listener dan aturan, pilih kotak centang untuk listener dan pilih Kelola listener, Hapus listener.
5. Saat diminta konfirmasi, masukkan **confirm**, lalu pilih Hapus.

## AWS CLI

Untuk menghapus pendengar

Gunakan perintah [hapus-listener](#).

```
aws elbv2 delete-listener \  
  --listener-arn listener-arn
```

# Kelompok-kelompok target untuk Application Load Balancers

Grup target merutekan permintaan ke target terdaftar individual, seperti instans EC2, menggunakan protokol dan nomor port yang Anda tentukan. Anda dapat mendaftarkan target dengan beberapa grup target. Anda dapat mengonfigurasi pemeriksaan kondisi berdasarkan per grup target.

Pemeriksaan kondisi dilakukan pada semua target yang terdaftar ke grup target yang ditentukan dalam aturan listener untuk penyeimbang beban Anda.

Setiapkelompok target terbiasa merutekan permintaan untuk satu atau lebih target terdaftar. Ketika Anda membuat setiap aturan pendengar, Anda menentukan kelompok target dan kondisi. Ketika kondisi aturan terpenuhi, lalu lintas diteruskan ke kelompok target yang sesuai. Anda dapat membuat kelompok-kelompok target yang berbeda untuk berbagai jenis permintaan. Misalnya, membuat satu kelompok target untuk permintaan umum dan kelompok target lain untuk permintaan ke layanan mikro untuk aplikasi Anda. Anda dapat menggunakan setiap grup target hanya dengan satu penyeimbang beban. Untuk informasi selengkapnya, lihat [Komponen Application Load Balancer](#).

Tentukan pengaturan pemeriksaan kesehatan untuk Load Balancer Anda berdasarkan per kelompok target. Setiap kelompok target menggunakan pengaturan pemeriksaan kondisi yang sudah ada, kecuali jika Anda menimpa mereka saat Anda membuat kelompok target atau mengubahnya nanti. Setelah Anda menentukan kelompok target dalam aturan untuk pendengar, load balancer terus memantau health semua target yang terdaftar dengan kelompok target yang berada di Availability Zone diaktifkan untuk penyeimbang beban. Load balancer merutekan permintaan ke target terdaftar yang sehat.

## Daftar Isi

- [Konfigurasi perutean](#)
- [Tipe target](#)
- [Jenis alamat IP](#)
- [Versi protokol](#)
- [Target-target terdaftar.](#)
- [Pengoimal Target](#)
- [Atribut grup target](#)
- [Kesehatan kelompok sasaran](#)
- [Buat grup target untuk Application Load Balancer Anda](#)
- [Pemeriksaan kondisi untuk grup target Penyeimbang Beban Aplikasi](#)

- [Mengedit atribut grup target untuk Application Load Balancer](#)
- [Daftarkan target dengan kelompok sasaran Application Load Balancer Anda](#)
- [Gunakan fungsi Lambda sebagai target Application Load Balancer](#)
- [Tag untuk kelompok target Application Load Balancer Anda](#)
- [Menghapus grup target Application Load Balancer](#)

## Konfigurasi perutean

Secara default, load balancer merutekan permintaan ke targetnya menggunakan protokol dan nomor port yang Anda tentukan saat Anda membuat grup target. Atau, Anda dapat mengganti port yang digunakan untuk merutekan lalu lintas ke target saat Anda mendaftarkannya dengan grup target.

Kelompok target mendukung protokol dan port berikut ini:

- Protokol: HTTP, HTTPS
- Port: 1-65535

Ketika grup target dikonfigurasi dengan protokol HTTPS atau menggunakan pemeriksaan kesehatan HTTPS, jika ada pendengar HTTPS yang menggunakan kebijakan keamanan TLS 1.3, kebijakan `ELBSecurityPolicy-TLS13-1-0-2021-06` keamanan akan digunakan untuk koneksi target. Jika tidak, kebijakan `ELBSecurityPolicy-2016-08` keamanan digunakan. Load balancer menetapkan koneksi TLS dengan target menggunakan sertifikat yang Anda instal pada target. Load balancer tidak memvalidasi sertifikat ini. Oleh karena itu, Anda dapat menggunakan sertifikat ditandatangani sendiri atau sertifikat yang telah kedaluwarsa. Karena load balancer, dan targetnya berada di virtual private cloud (VPC), lalu lintas antara load balancer dan target diautentikasi pada level paket, sehingga tidak berisiko terkena man-in-the-middle serangan atau spoofing meskipun sertifikat pada target tidak valid. Lalu lintas yang pergi tidak AWS akan memiliki perlindungan yang sama, dan langkah-langkah tambahan mungkin diperlukan untuk mengamankan lalu lintas lebih lanjut.

## Tipe target

Bila Anda membuat grup target, Anda menentukan jenis targetnya, yang menentukan jenis target yang Anda tentukan saat mendaftarkan target dengan grup target ini. Setelah Anda membuat grup target, Anda tidak dapat mengubah jenis targetnya.

Status yang mungkin muncul adalah sebagai berikut:

## instance

Target ditentukan oleh contoh ID.

## ip

Targetnya adalah alamat IP.

## lambda

Targetnya adalah fungsi Lambda.

Ketika jenis target `ip`, Anda dapat menentukan alamat IP dari salah satu blok CIDR berikut:

- Subnet dari VPC untuk kelompok target
- 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

Anda tidak dapat menentukan alamat IP yang dapat dirutekan publik.

Semua blok CIDR yang didukung memungkinkan Anda untuk mendaftarkan target berikut dengan grup target:

- Contoh dalam VPC yang diintip ke VPC penyeimbang beban (Wilayah yang sama atau Wilayah yang berbeda).
- AWS sumber daya yang dapat dialamatkan oleh alamat IP dan port (misalnya, database).
- Sumber daya lokal yang ditautkan ke AWS melalui Direct Connect atau koneksi Site-to-Site VPN.

### Note

Untuk Application Load Balancer yang ditempatkan di dalam Zona Lokal, `ip` target harus berada di Zona Lokal yang sama untuk menerima lalu lintas.

Untuk informasi selengkapnya, lihat [Apa itu AWS Local Zones?](#)

Jika Anda menentukan target menggunakan ID instance, lalu lintas dialihkan ke instance menggunakan alamat IP pribadi utama yang ditentukan dalam antarmuka jaringan utama untuk instance. Jika Anda menentukan target menggunakan alamat IP, Anda dapat mengarahkan lalu lintas ke instance menggunakan alamat IP pribadi dari satu atau beberapa antarmuka jaringan. Hal ini memungkinkan beberapa aplikasi pada contoh untuk menggunakan port yang sama. Setiap antarmuka jaringan dapat memiliki grup keamanan sendiri.

Jika jenis target grup target Anda `Lambda`, Anda dapat mendaftarkan fungsi Lambda tunggal. Ketika load balancer menerima permintaan untuk fungsi Lambda, fungsi Lambda akan terpicu. Untuk informasi selengkapnya, lihat [Gunakan fungsi Lambda sebagai target Application Load Balancer](#).

Anda dapat mengonfigurasi Amazon Elastic Container Service (Amazon ECS) sebagai target Application Load Balancer Anda. Untuk informasi selengkapnya, lihat [Menggunakan Application Load Balancer untuk Amazon ECS](#) di Panduan Pengembang Layanan Amazon Elastic Container.

## Jenis alamat IP

Saat membuat grup target baru, Anda dapat memilih jenis alamat IP grup target Anda. Ini mengontrol versi IP yang digunakan untuk berkomunikasi dengan target dan memeriksa status kesehatan mereka.

Grup target untuk Application Load Balancers Anda mendukung jenis alamat IP berikut:

### **ipv4**

Penyeimbang beban berkomunikasi dengan target menggunakan IPv4

### **ipv6**

Penyeimbang beban berkomunikasi dengan target menggunakan IPv6

### Pertimbangan-pertimbangan

- Load balancer berkomunikasi dengan target berdasarkan jenis alamat IP dari kelompok target. Target kelompok IPv4 sasaran harus menerima IPv4 lalu lintas dari penyeimbang beban dan target kelompok IPv6 sasaran harus menerima IPv6 lalu lintas dari penyeimbang beban.

- Anda tidak dapat menggunakan grup IPv6 target dengan penyeimbang `ipv4` beban.
- Anda tidak dapat mendaftarkan fungsi Lambda dengan grup IPv6 target.

## Versi protokol

Secara default, Application Load Balancers mengirim permintaan ke target menggunakan HTTP/1.1. Anda dapat menggunakan versi protokol untuk mengirim permintaan ke target menggunakan HTTP/2 atau gRPC.

Tabel berikut merangkum hasil untuk kombinasi protokol permintaan dan versi protokol kelompok target.

Protokol permintaan	Versi protokol	Hasil
HTTP/1.1	HTTP/1.1	Sukses
HTTP/2	HTTP/1.1	Sukses
gRPC	HTTP/1.1	Kesalahan
HTTP/1.1	HTTP/2	Kesalahan
HTTP/2	HTTP/2	Sukses
gRPC	HTTP/2	Sukses jika target mendukung gRPC
HTTP/1.1	gRPC	Kesalahan
HTTP/2	gRPC	Sukses jika permintaan POST
gRPC	gRPC	Sukses

### Pertimbangan untuk versi protokol gRPC

- Satu-satunya protokol pendengar yang didukung adalah HTTPS.
- Satu-satunya jenis tindakan yang didukung untuk aturan pendengar adalah `forward`.
- Jenis-jenis target yang didukung hanya `instance` dan `ip`.

- Load balancer mem-parsing permintaan gRPC dan rute panggilan gRPC ke kelompok target yang sesuai berdasarkan paket, layanan, dan metode.
- Load balancer mendukung streaming unary, client-side, streaming sisi server, dan streaming bi-directional.
- Anda harus menyediakan metode pemeriksaan kesehatan kustom dengan format/`package.service/method`.
- Anda harus menentukan kode status gRPC untuk digunakan ketika memeriksa untuk respon sukses dari target.
- Anda tidak dapat menggunakan fungsi Lambda sebagai target.

### Pertimbangan untuk versi protokol HTTP/2

- Satu-satunya protokol pendengar yang didukung adalah HTTPS.
- Satu-satunya jenis tindakan yang didukung untuk aturan pendengar adalah `forward`.
- Jenis-jenis target yang didukung hanya `instance` dan `ip`.
- Load balancer mendukung streaming unary, client-side, streaming sisi server, dan streaming bi-directional. Jumlah maksimum aliran per koneksi HTTP/2 klien adalah 128.

## Target-target terdaftar.

Load balancer Anda berfungsi sebagai titik kontak tunggal untuk klien dan mendistribusikan lalu lintas masuk ke seluruh target terdaftar yang sehat. Anda dapat mendaftarkan setiap target dengan satu atau lebih kelompok target.

Jika permintaan pada aplikasi Anda meningkat, Anda dapat mendaftarkan target tambahan dengan satu atau lebih kelompok target untuk menangani permintaan. Penyeimbang beban mulai merutekan lalu lintas ke target yang baru terdaftar segera setelah proses pendaftaran selesai dan target melewati pemeriksaan kesehatan awal pertama, terlepas dari ambang batas yang dikonfigurasi.

Jika permintaan pada aplikasi Anda menurun, atau Anda perlu untuk melayani target Anda, Anda dapat membatalkan pendaftaran (deregistrasi) target dari kelompok target Anda. Proses deregistrasi target menghapus itu dari kelompok target Anda, tetapi tidak mempengaruhi target sebaliknya. Load balancer berhenti routing permintaan ke target segera setelah pendaftaran terbatal. Target memasuki keadaan `draining` hingga permintaan dalam penerbangan telah selesai. Anda dapat mendaftarkan target dengan kelompok target lagi ketika target Anda siap untuk untuk melanjutkan menerima permintaan.

Jika Anda mendaftarkan target berdasarkan ID instans, Anda dapat menggunakan load balancer dengan grup Auto Scaling. Setelah Anda melampirkan grup target ke grup Auto Scaling, Auto Scaling akan mendaftarkan target Anda dengan grup target untuk Anda saat meluncurkannya. Untuk informasi selengkapnya, lihat Memasang load balancer ke grup Auto Scaling Anda dalam Amazon EC2 Auto Scaling User Guide.

## Batas

- Anda tidak dapat mendaftarkan alamat IP Application Load Balancer lain di VPC yang sama. Jika Application Load Balancer lainnya ada di VPC yang mengintip ke VPC load balancer, Anda dapat mendaftarkan alamat IP-nya.
- Anda tidak dapat mendaftarkan instance berdasarkan ID instans jika berada di VPC yang diintip ke VPC penyeimbang beban (Wilayah yang sama atau Wilayah yang berbeda). Anda dapat mendaftarkan instnas ini dengan alamat IP.

## Pengoptimal Target

Anda dapat mengaktifkan pengoptimal target pada grup target. Pengoptimal target memungkinkan Anda secara akurat menerapkan jumlah maksimum permintaan bersamaan pada target. Ia bekerja dengan bantuan agen yang Anda instal dan konfigurasi pada target. Untuk mengaktifkan pengoptimal target, Anda menentukan port kontrol target untuk grup target. Port ini digunakan untuk manajemen lalu lintas antara agen dan penyeimbang beban. Pengoptimal target hanya dapat diaktifkan selama pembuatan grup target. Port kontrol target setelah ditentukan tidak dapat dimodifikasi. Untuk informasi selengkapnya, lihat [the section called “Pengoptimal Target”](#).

## Atribut grup target

Anda dapat mengonfigurasi grup target dengan mengedit atributnya. Untuk informasi selengkapnya, lihat [Edit atribut grup target](#).

Atribut grup target berikut didukung jika jenis grup target `instance` atau `ip`:

`deregistration_delay.timeout_seconds`

Jumlah waktu tunggu untuk Elastic Load Balancing sebelum membatalkan pendaftaran target. Rentangnya adalah 0—3600 detik. Nilai default adalah 300 detik.

## `load_balancing.algorithm.type`

Algoritma routing menentukan bagaimana load balancer memilih target saat routing request. Nilainya adalah `round_robin`, `least_outstanding_requests`, atau `weighted_random`. Nilai default-nya `round_robin`.

## `load_balancing.algorithm.anomaly_mitigation`

Hanya tersedia bila `load_balancing.algorithm.type` adalah `weighted_random`. Menunjukkan apakah mitigasi anomali diaktifkan. Nilainya adalah `on` atau `off`. Default adalah `off`.

## `load_balancing.cross_zone.enabled`

Menunjukkan apakah penyeimbangan beban lintas zona diaktifkan. Nilainya adalah `true`, `false` atau `use_load_balancer_configuration`. Nilai default-nya `use_load_balancer_configuration`.

## `slow_start.duration_seconds`

Jangka waktu, dalam hitungan detik, di mana load balancer mengirimkan target yang baru terdaftar peningkatan secara linier bagian lalu lintas ke kelompok target. Jangkauannya adalah 30-900 detik (15 menit). Waktu default adalah 0 detik (tidak diaktifkan).

## `stickiness.enabled`

Menunjukkan apakah sesi lengket diaktifkan. Nilai dari `true` adalah `false`. Default adalah `false`.

## `stickiness.app_cookie.cookie_name`

Nama cookie aplikasi. Nama cookie aplikasi tidak dapat memiliki awalan berikut: `AWSALB`, `AWSALBAPP`, atau `AWSALBTG`; mereka dicadangkan untuk digunakan oleh penyeimbang beban.

## `stickiness.app_cookie.duration_seconds`

Periode kedaluwarsa cookie berbasis aplikasi, dalam hitungan detik. Setelah periode ini, cookie dianggap basi. Nilai minimum adalah 1 detik dan nilai maksimum adalah 7 hari (604800 detik). Nilai default adalah 1 hari (86400 detik).

## `stickiness.lb_cookie.duration_seconds`

Periode kedaluwarsa cookie berbasis durasi, dalam hitungan detik. Setelah periode ini, cookie dianggap basi. Nilai minimum adalah 1 detik dan nilai maksimum adalah 7 hari (604800 detik). Nilai default adalah 1 hari (86400 detik).

## `stickiness.type`

Jenis kelengketan. Nilai yang mungkin adalah ... dan ... .

## `target_group_health.dns_failover.minimum_healthy_targets.count`

Jumlah minimum target yang harus sehat. Jika jumlah target sehat di bawah nilai ini, tandai node sebagai tidak sehat di DNS, sehingga lalu lintas dirutekan hanya ke node yang sehat. Nilai yang mungkin adalah `off` atau bilangan bulat dari 1 ke jumlah maksimum target. Ketika `off`, DNS gagal dinonaktifkan, artinya meskipun semua target dalam grup target tidak sehat, node tidak dihapus dari DNS. Default-nya adalah 1.

## `target_group_health.dns_failover.minimum_healthy_targets.percentage`

Persentase minimum target yang harus sehat. Jika persentase target sehat di bawah nilai ini, tandai node sebagai tidak sehat di DNS, sehingga lalu lintas hanya diarahkan ke node yang sehat. Nilai yang mungkin adalah `off` atau bilangan bulat dari 1 hingga 100. Ketika `off`, DNS gagal dinonaktifkan, artinya meskipun semua target dalam grup target tidak sehat, node tidak dihapus dari DNS. Nilai default-nya `off`.

## `target_group_health.unhealthy_state_routing.minimum_healthy_targets.count`

Jumlah minimum target yang harus sehat. Jika jumlah target sehat di bawah nilai ini, kirim lalu lintas ke semua target, termasuk target yang tidak sehat. Kisarannya adalah 1 hingga jumlah target maksimum. Default-nya adalah 1.

## `target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage`

Persentase minimum target yang harus sehat. Jika persentase target sehat di bawah nilai ini, kirim lalu lintas ke semua target, termasuk target yang tidak sehat. Nilai yang mungkin adalah `off` atau bilangan bulat dari 1 hingga 100. Nilai default-nya `off`.

Atribut grup target berikut didukung jika jenis grup target `lambda`:

## `lambda.multi_value_headers.enabled`

Menunjukkan apakah permintaan dan respon header dipertukarkan antara load balancer dan fungsi Lambda termasuk array nilai atau string. Nilai yang mungkin adalah `true` atau `false`. Nilai default-nya adalah `false`. Untuk informasi selengkapnya, lihat [Header nilai ganda](#).

# Kesehatan kelompok sasaran

Secara default, kelompok sasaran dianggap sehat selama memiliki setidaknya satu target yang sehat. Jika Anda memiliki armada besar, hanya memiliki satu target yang sehat yang melayani lalu lintas tidak cukup. Sebagai gantinya, Anda dapat menentukan jumlah minimum atau persentase target yang harus sehat, dan tindakan apa yang dilakukan penyeimbang beban ketika target sehat jatuh di bawah ambang batas yang ditentukan. Ini meningkatkan ketersediaan aplikasi Anda.

## Daftar Isi

- [Tindakan negara yang tidak sehat](#)
- [Persyaratan dan pertimbangan](#)
- [Memantau](#)
- [Contoh](#)
- [Menggunakan failover DNS Route 53 untuk penyeimbang beban Anda](#)

## Tindakan negara yang tidak sehat

Anda dapat mengonfigurasi ambang batas yang sehat untuk tindakan berikut:

- DNS failover — Ketika target sehat di zona jatuh di bawah ambang batas, kami menandai alamat IP node penyeimbang beban untuk zona sebagai tidak sehat di DNS. Oleh karena itu, ketika klien menyelesaikan nama DNS penyeimbang beban, lalu lintas dialihkan hanya ke zona sehat.
- Routing failover — Ketika target sehat di zona jatuh di bawah ambang batas, penyeimbang beban mengirimkan lalu lintas ke semua target yang tersedia untuk node penyeimbang beban, termasuk target yang tidak sehat. Hal ini meningkatkan kemungkinan koneksi klien berhasil, terutama ketika target sementara gagal lulus pemeriksaan kesehatan, dan mengurangi risiko kelebihan beban target yang sehat.

## Persyaratan dan pertimbangan

- Jika Anda mengaktifkan pengoptimal target pada grup target, kami sarankan Anda mengatur port pemeriksaan kesehatan grup target agar sama dengan port di TARGET\_CONTROL\_DATA\_ADDRESS. Ini memastikan bahwa target akan gagal dalam pemeriksaan kesehatan jika agen tidak sehat. Untuk informasi selengkapnya, lihat [the section called “Pengoptimal Target”](#).

- Anda tidak dapat menggunakan fitur ini dengan grup target yang targetnya adalah fungsi Lambda. Jika Application Load Balancer adalah target Network Load Balancer atau Global Accelerator, jangan mengkonfigurasi ambang batas untuk failover DNS.
- Jika Anda menentukan kedua jenis ambang batas untuk suatu tindakan (hitungan dan persentase), penyeimbang beban akan mengambil tindakan ketika salah satu ambang batas dilanggar.
- Jika Anda menentukan ambang batas untuk kedua tindakan, ambang batas untuk failover DNS harus lebih besar dari atau sama dengan ambang batas untuk routing failover, sehingga failover DNS terjadi baik dengan atau sebelum routing failover.
- Jika Anda menentukan ambang batas sebagai persentase, kami menghitung nilai secara dinamis, berdasarkan jumlah total target yang terdaftar dengan kelompok target.
- Jumlah total target didasarkan pada apakah penyeimbangan beban lintas zona mati atau aktif. Jika penyeimbangan beban lintas zona tidak aktif, setiap node mengirimkan lalu lintas hanya ke target di zonanya sendiri, yang berarti bahwa ambang batas berlaku untuk jumlah target di setiap zona yang diaktifkan secara terpisah. Jika penyeimbangan beban lintas zona aktif, setiap node mengirimkan lalu lintas ke semua target di semua zona yang diaktifkan, yang berarti bahwa ambang batas yang ditentukan berlaku untuk target jumlah total di semua zona yang diaktifkan. Untuk informasi selengkapnya, lihat [Penyeimbangan beban lintas zona](#).
- Ketika failover DNS terjadi, itu berdampak pada semua kelompok target yang terkait dengan penyeimbang beban. Pastikan Anda memiliki kapasitas yang cukup di zona yang tersisa untuk menangani lalu lintas tambahan ini, terutama jika penyeimbangan beban lintas zona tidak aktif.
- Dengan failover DNS, kami menghapus alamat IP zona tidak sehat dari nama host DNS untuk penyeimbang beban. Namun, cache DNS klien lokal mungkin berisi alamat IP ini sampai time-to-live (TTL) dalam catatan DNS berakhir (60 detik).
- Dengan failover DNS, jika ada beberapa kelompok sasaran yang melekat pada Application Load Balancer dan satu kelompok sasaran tidak sehat di zona, pemeriksaan kesehatan DNS berhasil jika setidaknya satu kelompok sasaran lainnya sehat di zona tersebut.
- Dengan failover DNS, jika semua zona penyeimbang beban dianggap tidak sehat, penyeimbang beban mengirimkan lalu lintas ke semua zona, termasuk zona yang tidak sehat.
- Ada faktor selain apakah ada cukup target sehat yang dapat menyebabkan kegagalan DNS, seperti kesehatan zona.

## Memantau

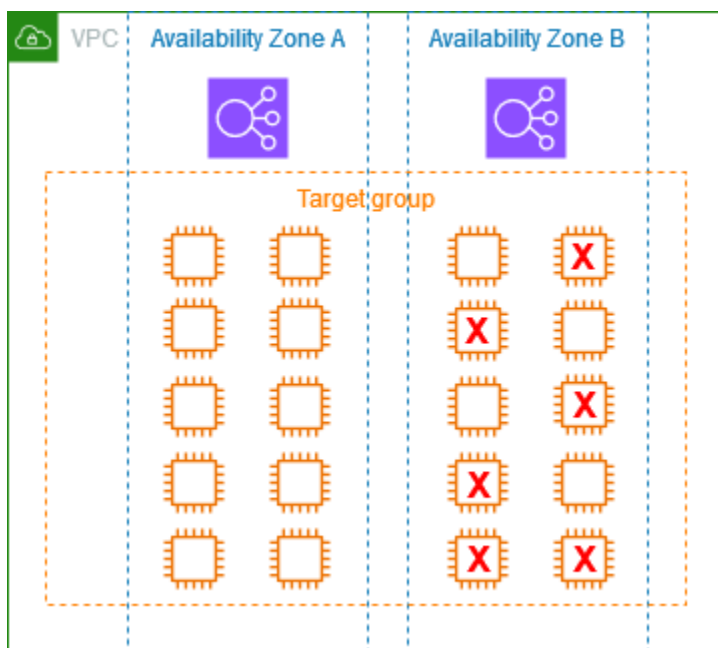
Untuk memantau kesehatan kelompok sasaran Anda, lihat [CloudWatch metrik untuk kesehatan kelompok sasaran](#).

## Contoh

Contoh berikut menunjukkan bagaimana pengaturan kesehatan kelompok target diterapkan.

### Skenario

- Penyeimbang beban yang mendukung dua Availability Zone, A dan B
- Setiap Availability Zone berisi 10 target terdaftar
- Kelompok sasaran memiliki pengaturan kesehatan kelompok sasaran berikut:
  - DNS failover - 50%
  - Routing failover - 50%
- Enam target gagal di Availability Zone B



Jika penyeimbangan beban lintas zona tidak aktif

- Node penyeimbang beban di setiap Availability Zone hanya dapat mengirim lalu lintas ke 10 target di Availability Zone.

- Ada 10 target sehat di Availability Zone A, yang memenuhi persentase target sehat yang diperlukan. Load balancer terus mendistribusikan lalu lintas antara 10 target sehat.
- Hanya ada 4 target sehat di Availability Zone B, yaitu 40% dari target untuk node penyeimbang beban di Availability Zone B. Karena ini kurang dari persentase target sehat yang dibutuhkan, penyeimbang beban mengambil tindakan berikut:
  - DNS failover - Availability Zone B ditandai sebagai tidak sehat di DNS. Karena klien tidak dapat menyelesaikan nama penyeimbang beban ke node penyeimbang beban di Availability Zone B, dan Availability Zone A sehat, klien mengirim koneksi baru ke Availability Zone A.
  - Routing failover - Ketika koneksi baru dikirim secara eksplisit ke Availability Zone B, load balancer mendistribusikan lalu lintas ke semua target di Availability Zone B, termasuk target yang tidak sehat. Ini mencegah pemadaman di antara target sehat yang tersisa.

Jika penyeimbangan beban lintas zona aktif

- Setiap node penyeimbang beban dapat mengirim lalu lintas ke semua 20 target terdaftar di kedua Availability Zone.
- Ada 10 target sehat di Availability Zone A dan 4 target sehat di Availability Zone B, dengan total 14 target sehat. Ini adalah 70% dari target untuk node penyeimbang beban di kedua Availability Zone, yang memenuhi persentase target sehat yang diperlukan.
- Penyeimbang beban mendistribusikan lalu lintas antara 14 target sehat di kedua Availability Zone.

## Menggunakan failover DNS Route 53 untuk penyeimbang beban Anda

Jika Anda menggunakan Route 53 untuk merutekan kueri DNS ke penyeimbang beban, Anda juga dapat mengonfigurasi failover DNS untuk penyeimbang beban menggunakan Route 53. Dalam konfigurasi failover, Route 53 memeriksa kesehatan target kelompok target untuk penyeimbang beban untuk menentukan apakah target tersebut tersedia. Jika tidak ada target sehat yang terdaftar di penyeimbang beban, atau jika penyeimbang beban itu sendiri tidak sehat, Route 53 mengarahkan lalu lintas ke sumber daya lain yang tersedia, seperti penyeimbang beban yang sehat atau situs web statis di Amazon S3.

Misalnya, misalkan Anda memiliki aplikasi web untuk `www.example.com`, dan Anda ingin instance redundan berjalan di belakang dua penyeimbang beban yang berada di Wilayah yang berbeda. Anda ingin lalu lintas terutama diarahkan ke penyeimbang beban di satu Wilayah, dan Anda ingin menggunakan penyeimbang beban di Wilayah lain sebagai cadangan selama kegagalan. Jika Anda mengonfigurasi failover DNS, Anda dapat menentukan penyeimbang beban primer dan sekunder

(cadangan) Anda. Rute 53 mengarahkan lalu lintas ke penyeimbang beban utama jika tersedia, atau ke penyeimbang beban sekunder sebaliknya.

Bagaimana mengevaluasi kesehatan target bekerja

- Jika evaluasi kesehatan target ditetapkan Yes pada catatan alias untuk Application Load Balancer, Route 53 mengevaluasi kesehatan sumber daya yang ditentukan oleh nilai `alias target` Route 53 menggunakan pemeriksaan kesehatan kelompok sasaran.
- Jika semua kelompok sasaran yang dilampirkan pada Application Load Balancer sehat, Route 53 menandai catatan alias sebagai sehat. Jika Anda mengonfigurasi ambang batas untuk grup target dan memenuhi ambang batasnya, itu melewati pemeriksaan kesehatan. Jika tidak, jika kelompok sasaran berisi setidaknya satu target yang sehat, ia melewati pemeriksaan kesehatan. Jika pemeriksaan kesehatan berlalu, Route 53 mengembalikan catatan sesuai dengan kebijakan perutean Anda. Jika kebijakan perutean failover digunakan, Route 53 mengembalikan catatan utama.
- Jika salah satu kelompok sasaran yang dilampirkan pada Application Load Balancer tidak sehat, catatan alias gagal dalam pemeriksaan kesehatan Route 53 (fail-open). Jika menggunakan evaluasi kesehatan target, kebijakan perutean failover mengarahkan lalu lintas ke sumber daya sekunder.
- Jika semua kelompok target yang dilampirkan ke Application Load Balancer kosong (tidak ada target), Route 53 menganggap catatan tidak sehat (fail-open). Jika menggunakan evaluasi kesehatan target, kebijakan perutean failover mengarahkan lalu lintas ke sumber daya sekunder.

Untuk informasi selengkapnya, lihat [Menggunakan ambang batas kesehatan grup target penyeimbang beban untuk meningkatkan ketersediaan](#) di AWS Blog dan [Mengonfigurasi failover DNS di Panduan Pengembang Amazon Route 53](#).

## Buat grup target untuk Application Load Balancer Anda

Anda mendaftarkan target Anda dengan grup target. Secara default, load balancer mengirimkan permintaan ke target terdaftar menggunakan port dan protokol yang Anda tentukan untuk kelompok target. Anda dapat mengganti port ini ketika Anda mendaftarkan setiap target dengan kelompok target.

Setelah membuat grup target, Anda dapat menambahkan tanda (tag).

Untuk merutekan lalu lintas ke target dalam kelompok target, tentukan kelompok target dalam suatu tindakan saat Anda membuat pendengar atau membuat aturan untuk pendengar Anda.

Untuk informasi selengkapnya, lihat [Aturan listener untuk Application Load Balancer Anda](#). Anda dapat menentukan grup target yang sama di beberapa pendengar, tetapi pendengar ini harus termasuk dalam Application Load Balancer yang sama. Untuk menggunakan grup target dengan penyeimbang beban, Anda harus memverifikasi bahwa grup target tidak digunakan oleh pendengar untuk penyeimbang beban lainnya.

Anda dapat menambah atau menghapus target dari grup target Anda kapan saja. Untuk informasi selengkapnya, lihat [Daftarkan target dengan kelompok sasaran Application Load Balancer Anda](#). Anda juga dapat mengubah pengaturan pemeriksaan kesehatan untuk grup target Anda. Untuk informasi selengkapnya, lihat [Memperbarui pengaturan pemeriksaan kesehatan dari grup target Application Load Balancer](#).

## Console

Untuk membuat grup target

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih Buat grup target.
4. Untuk Pilih jenis target Pilih Instans untuk mendaftarkan target berdasarkan contoh ID, Alamat IP untuk mendaftarkan target berdasarkan alamat IP, atau Fungsi Lambda untuk mendaftarkan fungsi Lambda sebagai target.
5. Untuk Nama kelompok target, ketik nama untuk grup target. Nama ini harus unik per wilayah per akun, dapat memiliki maksimum 32 karakter, harus berisi hanya karakter alfanumerik atau tanda hubung, dan tidak harus dimulai atau diakhiri dengan tanda hubung.
6. (Opsional) Protokol dan Port, memodifikasi nilai default yang diperlukan.
7. Jika jenis target adalah Instans atau alamat IP, pilih IPv4 atau IPv6 sebagai jenis alamat IP, jika tidak, lewati ke langkah berikutnya.

Perhatikan bahwa hanya target yang memiliki jenis alamat IP yang dipilih yang dapat dimasukkan dalam grup target ini. Jenis alamat IP tidak dapat diubah setelah grup target dibuat.

8. Untuk VPC, pilih Virtual Private Cloud (VPC). Perhatikan bahwa untuk jenis target alamat IP, pilihan yang VPCs tersedia adalah yang mendukung jenis alamat IP yang Anda pilih pada langkah sebelumnya.
9. (Opsional) Versi protokol, mengubah nilai default yang diperlukan. Untuk informasi selengkapnya, lihat [the section called "Versi protokol"](#).

10. (Opsional) pada bagian Pemeriksaan Health, ubah pengaturan default sesuai kebutuhan. Untuk informasi selengkapnya, lihat [the section called “Pengaturan pemeriksaan kondisi”](#).
11. Jika jenis target adalah Fungsi Lambda, Anda dapat mengaktifkan pemeriksaan kesehatan dengan memilih Mengaktifkan di bagian Pemeriksaan Health.
12. (Opsional) Untuk mengaktifkan pengoptimal Target pada grup target, tentukan port kontrol target. Port tidak dapat dimodifikasi setelah pembuatan grup target. Pengoptimal target bekerja dengan bantuan agen yang Anda instal pada target. Untuk informasi selengkapnya, lihat [the section called “Pengoptimal Target”](#).
13. (Opsional) Tambahkan satu atau lebih tag sebagai berikut:
  - a. Perluas bagian Tag.
  - b. Pilih Tambahkan tanda.
  - c. Masukkan kunci dan nilai untuk tanda tersebut.
14. Pilih Berikutnya.
15. (Opsional) Tambahkan satu atau lebih target sebagai berikut:
  - Jika jenis target adalah Instans Pilih satu atau beberapa instans, masukkan satu atau beberapa port, lalu pilih Sertakan sebagai tertunda di bawah ini.

Catatan: Instance harus memiliki IPv6 alamat utama yang ditetapkan untuk didaftarkan dengan grup IPv6 target.
  - Jika jenis targetnya adalah alamat IP, lakukan hal berikut:
    - a. Pilih VPC jaringan dari daftar, atau pilih Alamat IP pribadi lainnya.
    - b. Masukkan alamat IP secara manual, atau temukan alamat IP menggunakan detail instance. Anda dapat memasukkan hingga lima alamat IP sekaligus.
    - c. Masukkan port untuk merutekan lalu lintas ke alamat IP yang ditentukan.
    - d. Pilih Sertakan sebagai tertunda di bawah ini.
  - Jika jenis target adalah fungsi Lambda, tentukan satu fungsi Lambda atau hilangkan langkah ini dan tentukan fungsi Lambda nanti.
16. Pilih Buat grup target.

## AWS CLI

Untuk membuat grup target

Gunakan perintah [create-target-group](#). Contoh berikut membuat grup target dengan protokol HTTP, target yang terdaftar berdasarkan alamat IP, satu tag, dan pengaturan pemeriksaan kesehatan default.

```
aws elbv2 create-target-group \  
  --name my-target-group \  
  --protocol HTTP \  
  --port 80 \  
  --target-type ip \  
  --vpc-id vpc-1234567890abcdef0 \  
  --tags Key=department,Value=123
```

Untuk mendaftarkan target

Gunakan perintah [register-target](#) untuk mendaftarkan target dengan kelompok target. Sebagai contoh, lihat [the section called “Daftarkan target”](#).

## CloudFormation

Untuk membuat grup target

Tentukan sumber daya tipe [AWS::ElasticLoadBalancingV2::TargetGroup](#). Contoh berikut membuat grup target dengan protokol HTTP, target yang terdaftar berdasarkan alamat IP, satu tag, pengaturan pemeriksaan kesehatan default, dan dua target terdaftar.

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      Protocol: HTTP  
      Port: 80  
      TargetType: ip  
      VpcId: !Ref myVPC  
      Tags:  
        - Key: 'department'  
          Value: '123'  
      Targets:  
        - Id: 10.0.50.10  
          Port: 80  
        - Id: 10.0.50.20  
          Port: 80
```

# Pemeriksaan kondisi untuk grup target Penyeimbang Beban Aplikasi

Application Load Balancer Anda secara berkala mengirimkan permintaan ke target yang terdaftar untuk menguji statusnya. Uji ini disebut pemeriksaan kondisi.

Setiap rute node penyeimbang beban hanya meminta target dengan kondisi baik di Availability Zone yang diaktifkan untuk penyeimbang beban. Setiap node penyeimbang beban memeriksa kondisi setiap target, menggunakan pengaturan pemeriksaan kondisi untuk kelompok target yang target terdaftar. Setelah target Anda terdaftar, target itu harus lulus satu pemeriksaan kondisi agar dapat dianggap sehat. Setelah setiap pemeriksaan kondisi selesai, node penyeimbang beban menutup koneksi yang dibuat untuk pemeriksaan kondisi.

Jika kelompok sasaran hanya berisi target terdaftar yang tidak sehat, penyeimbang beban merutekan permintaan ke semua target tersebut, terlepas dari status kesehatannya. Ini berarti bahwa jika semua target gagal pemeriksaan kesehatan pada saat yang sama di semua Availability Zone yang diaktifkan, penyeimbang beban gagal dibuka. Efek dari fail-open adalah memungkinkan lalu lintas ke semua target di semua Availability Zone yang diaktifkan, terlepas dari status kesehatannya, berdasarkan algoritma load balancing.

Pemeriksaan kesehatan tidak mendukung WebSockets.

Untuk informasi selengkapnya, lihat [the section called “Kesehatan kelompok sasaran”](#).

Anda dapat menggunakan log pemeriksaan kesehatan untuk menangkap informasi terperinci tentang pemeriksaan kesehatan yang dilakukan ke target terdaftar untuk penyeimbang beban Anda dan menyimpannya sebagai file log di Amazon S3. Anda dapat menggunakan log pemeriksaan kesehatan ini untuk memecahkan masalah dengan target Anda. Untuk informasi selengkapnya, lihat [Log pemeriksaan kesehatan](#).

## Daftar Isi

- [Pengaturan pemeriksaan kondisi](#)
- [Status kondisi target](#)
- [Kode alasan pemeriksaan kondisi](#)
- [Periksa kesehatan target Application Load Balancer Anda](#)
- [Memperbarui pengaturan pemeriksaan kesehatan dari grup target Application Load Balancer](#)

## Pengaturan pemeriksaan kondisi

Anda mengonfigurasi pemeriksaan kondisi untuk target dalam grup target seperti yang dijelaskan dalam tabel berikut. Nama pengaturan yang digunakan dalam tabel adalah nama yang digunakan dalam API. Penyeimbang beban mengirimkan permintaan pemeriksaan kesehatan ke setiap target yang terdaftar setiap `HealthCheckIntervalSeconds` detik, menggunakan port, protokol, dan jalur pemeriksaan kesehatan yang ditentukan. Setiap permintaan pemeriksaan kondisi bersifat independen dan hasilnya berlaku selama seluruh interval. Waktu yang dibutuhkan untuk target untuk merespons tidak memengaruhi interval untuk permintaan pemeriksaan kondisi berikutnya. Jika pemeriksaan kesehatan melebihi kegagalan `UnhealthyThresholdCount` berturut-turut, penyeimbang beban mengeluarkan target dari layanan. Ketika pemeriksaan kesehatan melebihi keberhasilan `HealthyThresholdCount` berturut-turut, penyeimbang beban menempatkan target kembali dalam layanan.

Perhatikan bahwa ketika Anda membatalkan pendaftaran target, ini berkurang `HealthyHostCount` tetapi tidak meningkat. `UnhealthyHostCount`

Pengaturan	Deskripsi
<code>HealthCheckProtocol</code>	<p>Protokol yang digunakan penyeimbang beban saat melakukan pemeriksaan kondisi pada target. Untuk Application Load Balancers protokol yang mungkin adalah HTTP dan HTTPS. Defaultnya adalah protokol HTTP.</p> <p>Protokol ini menggunakan metode HTTP GET untuk mengirim permintaan pemeriksaan kesehatan.</p>
<code>HealthCheckPort</code>	<p>Port penyeimbang beban digunakan saat melakukan pemeriksaan kondisi pada target. Defaultnya adalah dengan menggunakan port di mana setiap target menerima lalu lintas dari penyeimbang beban.</p>
<code>HealthCheckPath</code>	<p>Tujuan pemeriksaan kondisi pada target.</p>

Pengaturan	Deskripsi
	<p>Jika versi protokol adalah HTTP/1.1 atau HTTP/2, tentukan URI yang valid (/PATH?query). Defaultnya adalah /.</p> <p>Jika versi protokol adalah gRPC, tentukan jalur metode pemeriksaan kondisi kustom dengan format /package.service/method . Nilai default-nya /AWS.ALB/healthcheck .</p>
HealthCheckTimeoutSeconds	<p>Jumlah waktu, dalam detik, di mana tidak ada respons dari target berarti pemeriksaan kondisi gagal. Rentangnya adalah 2–120 detik. Nilai default adalah 5 detik jika jenis target adalah instance atau ip dan 30 detik jika jenis target adalah lambda.</p>
HealthCheckIntervalSeconds	<p>Perkiraan jumlah waktu, dalam hitungan detik, antara pemeriksaan kondisi dari target individu. Rentangnya adalah 5-300 detik. Defaultnya adalah 30 detik jika jenis target adalah instance atau ip dan 35 detik jika jenis target adalah lambda.</p>
HealthyThresholdCount	<p>Jumlah pemeriksaan kondisi yang berhasil berturut-turut diperlukan sebelum menganggap target yang tidak sehat memiliki kondisi sehat. Rentangnya adalah 2–10. Defaultnya adalah 5.</p>
UnhealthyThresholdCount	<p>Jumlah pemeriksaan kondisi yang gagal berturut-turut diperlukan sebelum menganggap target yang tidak memiliki kondisi sehat. Rentangnya adalah 2–10. Defaultnya adalah 2.</p>

Pengaturan	Deskripsi
Matcher	<p>Kode yang digunakan saat memeriksa respons yang berhasil dari target. Ini disebut Kode berhasil pada konsol.</p> <p>Jika versi protokol HTTP/1.1 atau HTTP/2, nilai yang mungkin adalah 200 hingga 499. Anda dapat menentukan beberapa nilai (misalnya, "200,202") atau rentang nilai (misalnya, "200-299"). Nilai default adalah 200.</p> <p>Jika versi protokol adalah gRPC, nilai yang mungkin adalah dari 0 sampai 99. Anda dapat menentukan beberapa nilai (misalnya, "0,1") atau rentang nilai (misalnya, "0-5"). Nilai default adalah 12.</p>

## Status kondisi target

Sebelum penyeimbang beban mengirimkan permintaan pemeriksaan kondisi ke target, Anda harus mendaftarkannya dengan grup target, menentukan kelompok targetnya dalam aturan listener, dan memastikan bahwa Availability Zone target diaktifkan untuk penyeimbang beban. Sebelum target dapat menerima permintaan dari penyeimbang beban, target harus lulus pemeriksaan kondisi awal. Setelah target melewati pemeriksaan kondisi awal, statusnya adalah `Healthy`.

Tabel berikut menjelaskan nilai yang mungkin untuk status kondisi target terdaftar.

Nilai	Deskripsi
<code>initial</code>	<p>Penyeimbang beban sedang dalam proses mendaftarkan target atau melakukan pemeriksaan kondisi awal pada target.</p> <p>Kode alasan terkait: <code>Elb.RegistrationInProgress</code>   <code>Elb.InitialHealthChecking</code></p>
<code>healthy</code>	Targetnya sehat.

Nilai	Deskripsi
	Kode alasan terkait: Tidak ada
unhealthy	<p>Target tidak merespons pemeriksaan kondisi atau gagal dalam pemeriksaan kondisi.</p> <p>Kode alasan terkait: <code>Target.ResponseCodeMismatch</code>   <code>Target.Timeout</code>   <code>Target.FailedHealthChecks</code>   <code>Elb.InternalError</code></p>
unused	<p>Target tidak terdaftar dengan grup target, kelompok target tidak digunakan dalam aturan listener, target ada di Availability Zone yang tidak diaktifkan, atau target dalam keadaan berhenti atau dihentikan.</p> <p>Kode alasan terkait: <code>Target.NotRegistered</code>   <code>Target.NotInUse</code>   <code>Target.InvalidState</code>   <code>Target.IpUnusable</code></p>
draining	<p>Target membatalkan pendaftaran dan pengosongan koneksi sedang dalam proses.</p> <p>Kode alasan terkait: <code>Target.DeregistrationInProgress</code></p>
unavailable	<p>Pemeriksaan kondisi dinonaktifkan untuk grup target.</p> <p>Kode alasan terkait: <code>Target.HealthCheckDisabled</code></p>

## Kode alasan pemeriksaan kondisi

Jika status target adalah nilai apa pun selain `Healthy`, API mengembalikan kode alasan dan deskripsi masalah, dan konsol menampilkan deskripsi yang sama. Kode alasan yang dimulai dengan `Elb` berasal dari sisi penyeimbang beban dan kode alasan yang dimulai dengan `Target` berasal dari sisi target. Untuk informasi selengkapnya tentang kemungkinan penyebab kegagalan pemeriksaan kesehatan, lihat [Pemecahan masalah](#).

Kode alasan	Deskripsi
Elb.InitialHealthChecking	Pemeriksaan kondisi awal sedang berlangsung
Elb.InternalError	Pemeriksaan kondisi gagal karena kesalahan internal
Elb.RegistrationInProgress	Pendaftaran target sedang berlangsung
Target.DeregistrationInProgress	Pembatalan pendaftaran target sedang berlangsung
Target.FailedHealthChecks	Pemeriksaan kondisi gagal
Target.HealthCheckDisabled	Pemeriksaan kondisi dinonaktifkan
Target.InvalidState	Target dalam keadaan berhenti Target dalam keadaan dihentikan Target berada dalam keadaan dihentikan atau berhenti Target dalam keadaan tidak valid
Target.IpUnusable	Alamat IP tidak dapat digunakan sebagai target, karena digunakan oleh penyeimbang beban
Target.NotInUse	Grup target tidak dikonfigurasi untuk menerima lalu lintas dari penyeimbang beban Target berada di Availability Zone yang tidak diaktifkan untuk penyeimbang beban
Target.NotRegistered	Target tidak terdaftar ke grup target
Target.ResponseCodeMismatch	Pemeriksaan kondisi gagal dengan kode-kode ini: [code]
Target.Timeout	Batas waktu permintaan habis



```
--query "TargetHealthDescriptions[?TargetHealth.State!='healthy'].  
[Target.Id,TargetHealth.State,TargetHealth.Reason]" \  
--output table
```

Berikut ini adalah output contoh.

```
-----  
|          DescribeTargetHealth          |  
+-----+-----+-----+  
| 172.31.0.57 | unused | Target.NotInUse |  
| 172.31.0.50 | unused | Target.NotInUse |  
+-----+-----+-----+
```

## Status target dan kode alasan

Daftar berikut menunjukkan kode alasan yang mungkin untuk setiap status target.

### Status target adalah healthy

Kode alasan tidak disediakan.

### Status target adalah initial

- `Elb.RegistrationInProgress`- Targetnya sedang dalam proses didaftarkan pada load balancer.
- `Elb.InitialHealthChecking`- Load balancer masih mengirimkan target jumlah minimum pemeriksaan kesehatan yang diperlukan untuk menentukan status kesehatannya.

### Status target adalah unhealthy

- `Target.ResponseCodeMismatch`- Pemeriksaan kesehatan tidak mengembalikan kode HTTP yang diharapkan.
- `Target.Timeout`- Permintaan pemeriksaan kesehatan habis.
- `Target.FailedHealthChecks`- Penyeimbang beban menerima kesalahan saat membuat koneksi ke target atau respons target salah bentuk.
- `Elb.InternalError`- Pemeriksaan kesehatan gagal karena kesalahan internal.

### Status target adalah unused

- `Target.NotRegistered`- Target tidak terdaftar dengan kelompok sasaran.
- `Target.NotInUse`- Grup sasaran tidak digunakan oleh penyeimbang beban atau target berada di Availability Zone yang tidak diaktifkan untuk penyeimbang muatannya.

- `Target.InvalidState`- Target dalam keadaan berhenti atau dihentikan.
- `Target.IpUnusable`- Alamat IP target dicadangkan untuk digunakan oleh penyeimbang beban.

Status target adalah draining

- `Target.DeregistrationInProgress`- Target sedang dalam proses dideregistrasi dan periode penundaan deregistrasi belum kedaluwarsa.

Status target adalah unavailable

- `Target.HealthCheckDisabled`- Pemeriksaan kesehatan dinonaktifkan untuk kelompok sasaran.

## Memperbarui pengaturan pemeriksaan kesehatan dari grup target Application Load Balancer

Anda dapat memperbarui pengaturan pemeriksaan kesehatan untuk grup target Anda kapan saja. Untuk daftar pengaturan pemeriksaan kesehatan, lihat [the section called “Pengaturan pemeriksaan kondisi”](#).

### Console

Untuk memperbarui pengaturan pemeriksaan kesehatan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Pemeriksaan kondisi, pilih Edit.
5. Pada halaman Edit pengaturan pemeriksaan kesehatan, ubah pengaturan sesuai kebutuhan.
6. Pilih Simpan perubahan.

### AWS CLI

Untuk memperbarui pengaturan pemeriksaan kesehatan

Gunakan perintah [modify-target-group](#). Contoh berikut memperbarui `HealthyThresholdCount` dan `HealthCheckTimeoutSeconds` pengaturan.

```
aws elbv2 modify-target-group \
```

```
--target-group-arn target-group-arn \  
--healthy-threshold-count 3 \  
--health-check-timeout-seconds 20
```

## CloudFormation

Untuk memperbarui pengaturan pemeriksaan kesehatan

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya untuk menyertakan pengaturan pemeriksaan kesehatan yang diperbarui. Contoh berikut memperbarui `HealthyThresholdCount` dan `HealthCheckTimeoutSeconds` pengaturan.

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      Protocol: HTTP  
      Port: 80  
      TargetType: instance  
      VpcId: !Ref myVPC  
      HealthyThresholdCount: 3  
      HealthCheckTimeoutSeconds: 20
```

## Mengedit atribut grup target untuk Application Load Balancer

Setelah membuat grup target untuk Application Load Balancer, Anda dapat mengedit atribut grup targetnya.

### Atribut grup target

- [Penundaan Pembatalan Pendaftaran](#)
- [Algoritma perutean](#)
- [Mode mulai lambat](#)
- [Pengaturan Kesehatan](#)
- [Penyeimbangan beban lintas zona](#)
- [Bobot Target Otomatis \(ATW\)](#)
- [Sesi lengket](#)

## Penundaan Pembatalan Pendaftaran

Elastic Load Balancing berhenti mengirim permintaan ke target yang membatalkan pendaftaran. Secara default, Elastic Load Balancing menunggu 300 detik sebelum menyelesaikan proses pembatalan pendaftaran, yang dapat membantu permintaan dalam penerbangan ke target untuk diselesaikan. Untuk mengubah jumlah waktu tunggu Elastic Load Balancing, memperbarui nilai penundaan pembatalan registrasi.

Keadaan awal dari target deregistering adalah `draining`. Setelah penundaan deregistrasi berlalu, proses deregistrasi selesai dan keadaan target adalah `unused`. Jika target adalah bagian dari grup Auto Scaling, maka dapat dihentikan dan diganti.

Jika target pembatalan pendaftaran tidak memiliki permintaan dalam penerbangan dan tidak ada koneksi aktif, Elastic Load Balancing akan segera menyelesaikan proses pembatalan pendaftaran, tanpa menunggu penundaan pembatalan pendaftaran berlalu. Namun, meskipun deregistrasi target selesai, status target ditampilkan `draining` hingga batas waktu tunda deregistrasi berakhir. Setelah batas waktu berakhir, target bertransisi ke status `unused`.

Jika proses deregistrasi target mengakhiri sambungan sebelum penundaan deregistrasi berlalu, klien menerima respons error tingkat 500.

### Console

Untuk memperbarui nilai penundaan deregistrasi

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Di panel manajemen deregistrasi target, masukkan nilai baru untuk penundaan deregistrasi.
6. Pilih Simpan perubahan.

### AWS CLI

Untuk memperbarui nilai penundaan deregistrasi

Gunakan [modify-target-group-attributes](#) perintah dengan `deregistration_delay.timeout_seconds` atribut.

```
aws elbv2 modify-target-group-attributes \  
  --target-group-arn target-group-arn \  
  --attributes "Key=deregistration_delay.timeout_seconds,Value=60"
```

## CloudFormation

Untuk memperbarui nilai penundaan deregistrasi

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya untuk menyertakan `deregistration_delay.timeout_seconds` atribut.

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      Protocol: HTTP  
      Port: 80  
      TargetType: ip  
      VpcId: !Ref myVPC  
      TargetGroupAttributes:  
        - Key: "deregistration_delay.timeout_seconds"  
          Value: "60"
```

## Algoritma perutean

Algoritma routing adalah metode yang digunakan oleh load balancer saat menentukan target mana yang akan menerima permintaan. Algoritma routing round robin digunakan secara default untuk merutekan permintaan di tingkat grup target. Permintaan yang paling tidak menonjol dan algoritme perutean acak tertimbang juga tersedia berdasarkan kebutuhan aplikasi Anda. Grup target hanya dapat memiliki satu algoritma routing aktif pada satu waktu, namun algoritma routing dapat diperbarui kapan pun diperlukan.

Jika Anda mengaktifkan sesi lengket, algoritme perutean yang dipilih akan digunakan untuk pemilihan target awal. Permintaan masa depan dari klien yang sama akan diteruskan ke target yang sama, melewati algoritma routing yang dipilih. Jika Anda telah mengaktifkan pengoptimal target, algoritme perutean hanya bisa berupa round robin.

## Round robin

- Algoritma routing round robin merutekan permintaan secara merata di seluruh target sehat dalam kelompok target, dalam urutan berurutan.
- Algoritma ini biasanya digunakan ketika permintaan yang diterima memiliki kompleksitas yang serupa, target yang terdaftar serupa dalam kemampuan pemrosesan, atau jika Anda perlu mendistribusikan permintaan secara merata di antara target.

## Permintaan paling tidak tertunda

- Algoritma perutean permintaan yang paling tidak menonjol merutekan permintaan ke target dengan jumlah permintaan yang sedang berlangsung terendah.
- Algoritma ini biasanya digunakan ketika permintaan yang diterima bervariasi dalam kompleksitas, target terdaftar bervariasi dalam kemampuan pemrosesan.
- Ketika penyeimbang beban yang mendukung HTTP/2 menggunakan target yang hanya mendukung HTTP/1.1, ia mengubah permintaan menjadi beberapa permintaan HTTP/1.1. Dalam konfigurasi ini, algoritma permintaan yang paling tidak beredar akan memperlakukan setiap permintaan HTTP/2 sebagai beberapa permintaan.
- Saat menggunakan WebSockets, target dipilih menggunakan algoritma permintaan yang paling tidak beredar. Setelah target dipilih, penyeimbang beban membuat koneksi ke target dan mengirim semua pesan melalui koneksi ini.
- Algoritma routing permintaan yang paling tidak menonjol tidak dapat digunakan dengan mode start lambat.

## Acak tertimbang

- Algoritma perutean acak tertimbang merutekan permintaan secara merata di seluruh target sehat dalam kelompok target, dalam urutan acak.
- Algoritma ini mendukung mitigasi anomali Automatic Target Weights (ATW).
- Algoritma routing acak tertimbang tidak dapat digunakan dengan mode start lambat.
- Algoritma routing acak tertimbang tidak dapat digunakan dengan sesi lengket.

## Console

Untuk memperbarui algoritma routing

1. Buka konsol Amazon EC2 di. <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Di panel konfigurasi Lalu lintas, untuk algoritma Load balancing, pilih Round robin, Least outstanding request, atau Weighted random.
6. Pilih Simpan perubahan.

## AWS CLI

Untuk memperbarui algoritma routing

Gunakan [modify-target-group-attributes](#) perintah dengan `load_balancing.algorithm.type` atribut.

```
aws elbv2 modify-target-group-attributes \  
  --target-group-arn target-group-arn \  
  --attributes  
  "Key=load_balancing.algorithm.type,Value=least_outstanding_requests"
```

## CloudFormation

Untuk memperbarui algoritma routing

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya untuk menyertakan `load_balancing.algorithm.type` atribut.

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      Protocol: HTTP  
      Port: 80
```

```
TargetType: ip
VpcId: !Ref myVPC
TargetGroupAttributes:
  - Key: "load_balancing.algorithm.type"
    Value: "least_outstanding_requests"
```

## Mode mulai lambat

Secara default, target mulai menerima bagian penuh dari permintaan segera setelah terdaftar dengan kelompok target dan melewati pemeriksaan kesehatan awal. Menggunakan mode start lambat memberikan target waktu untuk pemanasan sebelum load balancer mengirimkan bagian penuh permintaan.

Setelah Anda mengaktifkan lambat mulai untuk kelompok target, target memasuki mode mulai lambat ketika mereka dianggap sehat oleh kelompok target. Target dalam mode start lambat keluar dari mode mulai lambat ketika periode durasi mulai lambat dikonfigurasi berlalu atau target menjadi tidak sehat. Load balancer secara linear meningkatkan jumlah permintaan yang dapat dikirim ke target dalam mode start lambat. Setelah target yang sehat keluar dari mode start yang lambat, load balancer dapat mengirimkan bagian penuh permintaan.

### Pertimbangan-pertimbangan

- Ketika Anda mengaktifkan mode mulai lambat untuk kelompok target, target sehat yang telah terdaftar dengan kelompok target tidak masuk mode tersebut.
- Ketika Anda mengaktifkan mulai lambat untuk kelompok target kosong, lalu mendaftarkan target menggunakan operasi pendaftaran tunggal, target ini tidak masuk mode mulai lambat. Target yang baru terdaftar memasuki mode mulai lambat hanya ketika ada setidaknya satu target sehat yang tidak dalam mode start lambat.
- Jika Anda membatalkan pendaftaran (deregister) target dalam mode mulai lambat, target keluar dari mode start lambat. Jika Anda mendaftarkan target yang sama lagi, memasuki mode start lambat ketika dianggap sehat oleh kelompok target.
- Jika target dalam mode start lambat menjadi tidak sehat, target keluar dari mode start lambat. Ketika target menjadi sehat, ia memasuki mode mulai lambat lagi.
- Anda tidak dapat mengaktifkan mode mulai lambat saat menggunakan permintaan yang paling tidak beredar atau algoritme perutean acak tertimbang.

## Console

Untuk memperbarui nilai durasi mulai lambat

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Di panel konfigurasi Lalu lintas, masukkan nilai baru untuk Durasi mulai lambat. Untuk menonaktifkan mode mulai lambat, masukkan 0.
6. Pilih Simpan perubahan.

## AWS CLI

Untuk memperbarui nilai durasi mulai lambat

Gunakan [modify-target-group-attributes](#) perintah dengan `slow_start.duration_seconds` atribut.

```
aws elbv2 modify-target-group-attributes \  
  --target-group-arn target-group-arn \  
  --attributes "Key=slow_start.duration_seconds,Value=30"
```

## CloudFormation

Untuk memperbarui nilai durasi mulai lambat

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya untuk menyertakan `slow_start.duration_seconds` atribut.

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      Protocol: HTTP  
      Port: 80  
      TargetType: ip  
      VpcId: !Ref myVPC  
      TargetGroupAttributes:
```

```
- Key: "slow_start.duration_seconds"  
  Value: "30"
```

## Pengaturan Kesehatan

Secara default, Application Load Balancer memantau kesehatan target dan merutekan permintaan ke target yang sehat. Namun, jika penyeimbang beban tidak memiliki target yang cukup sehat, maka secara otomatis mengirimkan lalu lintas ke semua target yang terdaftar (gagal terbuka). Anda dapat mengubah pengaturan kesehatan grup target untuk grup target Anda untuk menentukan ambang batas untuk failover DNS dan failover routing. Untuk informasi selengkapnya, lihat [the section called "Kesehatan kelompok sasaran"](#).

### Console

Untuk memodifikasi pengaturan kesehatan kelompok sasaran

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbang Beban, pilih Grup Target.
3. Pilih nama target grup untuk menampilkan halaman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Periksa apakah penyeimbangan beban lintas zona dihidupkan atau dimatikan. Perbarui pengaturan ini sesuai kebutuhan untuk memastikan bahwa Anda memiliki kapasitas yang cukup untuk menangani lalu lintas tambahan jika zona gagal.
6. Perluas persyaratan kesehatan kelompok sasaran.
7. Untuk jenis Konfigurasi, sebaiknya pilih Konfigurasi terpadu, yang menetapkan ambang batas yang sama untuk kedua tindakan tersebut.
8. Untuk persyaratan keadaan Sehat, lakukan salah satu hal berikut:
  - Pilih Jumlah target sehat minimum, lalu masukkan angka dari 1 hingga jumlah target maksimum untuk kelompok target Anda.
  - Pilih Persentase target sehat minimum, lalu masukkan angka dari 1 hingga 100.
9. Pilih Simpan perubahan.

### AWS CLI

Untuk memodifikasi pengaturan kesehatan kelompok sasaran

Gunakan perintah [modify-target-group-attributes](#). Contoh berikut menetapkan ambang batas yang sehat untuk kedua tindakan negara yang tidak sehat menjadi 50%.

```
aws elbv2 modify-target-group-attributes \  
  --target-group-arn target-group-arn \  
  --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"
```

## CloudFormation

Untuk memodifikasi pengaturan kesehatan kelompok sasaran

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya. Contoh berikut menetapkan ambang batas yang sehat untuk kedua tindakan negara yang tidak sehat menjadi 50%.

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      Protocol: HTTP  
      Port: 80  
      TargetType: ip  
      VpcId: !Ref myVPC  
      TargetGroupAttributes:  
        - 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"
```

## Penyeimbangan beban lintas zona

Node untuk Load Balancer Anda mendistribusikan permintaan dari klien ke target yang telah terdaftar. Saat penyeimbangan beban lintas zona aktif, setiap node penyeimbang beban mendistribusikan lalu lintas ke seluruh target terdaftar di semua Availability Zone yang terdaftar. Ketika penyeimbangan beban lintas zona mati, setiap node penyeimbang beban mendistribusikan

lalu lintas hanya di seluruh target yang terdaftar di Availability Zone. Ini bisa terjadi jika domain kegagalan zona lebih disukai daripada regional, memastikan bahwa zona sehat tidak terpengaruh oleh zona yang tidak sehat, atau untuk peningkatan latensi secara keseluruhan.

Dengan Application Load Balancers, penyeimbangan beban lintas zona selalu dihidupkan pada tingkat penyeimbang beban, dan tidak dapat dimatikan. Untuk grup target, defaultnya adalah menggunakan pengaturan penyeimbang beban, tetapi Anda dapat mengganti default dengan menonaktifkan penyeimbangan beban lintas zona secara eksplisit di tingkat grup target.

### Pertimbangan-pertimbangan

- Kelengkapan target tidak didukung saat penyeimbangan beban lintas zona mati.
- Lambda berfungsi sebagai target tidak didukung saat penyeimbangan beban lintas zona tidak aktif.
- Mencoba mematikan penyeimbangan beban lintas zona melalui `ModifyTargetGroupAttributes` API jika ada target yang memiliki parameter yang `AvailabilityZone` disetel untuk `all` menghasilkan kesalahan.
- Saat mendaftarkan target, `AvailabilityZone` parameter diperlukan. Nilai Zona Ketersediaan Khusus hanya diperbolehkan saat penyeimbangan beban lintas zona tidak aktif. Jika tidak, parameter diabaikan dan diperlakukan sebagai `all`.

### Praktik terbaik

- Rencanakan kapasitas target yang cukup di semua Availability Zone yang Anda harapkan untuk digunakan, per kelompok target. Jika Anda tidak dapat merencanakan kapasitas yang cukup di semua Availability Zone yang berpartisipasi, sebaiknya Anda tetap mengaktifkan penyeimbangan beban lintas zona.
- Saat mengonfigurasi Application Load Balancer Anda dengan beberapa grup target, pastikan semua grup target berpartisipasi dalam Availability Zone yang sama, di dalam Region yang dikonfigurasi. Ini untuk menghindari Availability Zone kosong saat penyeimbangan beban lintas zona tidak aktif, karena ini memicu kesalahan 503 untuk semua permintaan HTTP yang masuk ke Availability Zone kosong.
- Hindari membuat subnet kosong. Application Load Balancers mengekspos alamat IP zonal melalui DNS untuk subnet kosong, yang memicu 503 kesalahan untuk permintaan HTTP.
- Mungkin ada kejadian di mana kelompok target dengan penyeimbangan beban lintas zona dimatikan memiliki kapasitas target yang direncanakan yang cukup per Availability Zone, tetapi semua target di Availability Zone menjadi tidak sehat. Ketika ada setidaknya satu kelompok target

dengan semua target yang tidak sehat, alamat IP dari node penyeimbang beban dihapus dari DNS. Setelah kelompok target memiliki setidaknya satu target yang sehat, alamat IP dikembalikan ke DNS.

## Matikan penyeimbangan beban lintas zona

Anda dapat mematikan penyeimbangan beban lintas zona untuk grup target Application Load Balancer Anda kapan saja.

### Console

Untuk mematikan penyeimbangan beban lintas zona

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Di panel konfigurasi pemilihan Target, pilih Off for Cross-zone load balancing.
6. Pilih Simpan perubahan.

### AWS CLI

Untuk mematikan penyeimbangan beban lintas zona

Gunakan [modify-target-group-attributes](#) perintah dan atur `load_balancing.cross_zone.enabled` atribut ke `false`.

```
aws elbv2 modify-target-group-attributes \  
  --target-group-arn target-group-arn \  
  --attributes "Key=load_balancing.cross_zone.enabled,Value=false"
```

### CloudFormation

Untuk mematikan penyeimbangan beban lintas zona

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya untuk menyertakan `load_balancing.cross_zone.enabled` atribut.

```
Resources :
```

```
myTargetGroup:
  Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
  Properties:
    Name: my-target-group
    Protocol: HTTP
    Port: 80
    TargetType: ip
    VpcId: !Ref myVPC
    TargetGroupAttributes:
      - Key: "load_balancing.cross_zone.enabled"
        Value: "false"
```

## Aktifkan penyeimbangan beban lintas zona

Anda dapat mengaktifkan penyeimbangan beban lintas zona untuk grup target Application Load Balancer Anda kapan saja. Pengaturan penyeimbangan beban lintas zona pada tingkat kelompok target mengesampingkan pengaturan di tingkat penyeimbang beban.

### Console

Untuk mematikan penyeimbangan beban lintas zona

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Di panel konfigurasi pemilihan target, pilih Aktif untuk penyeimbangan beban lintas zona.
6. Pilih Simpan perubahan.

### AWS CLI

Untuk mengaktifkan penyeimbangan beban lintas zona

Gunakan [modify-target-group-attributes](#) perintah dan atur `load_balancing.cross_zone.enabled` atribut ke `true`.

```
aws elbv2 modify-target-group-attributes \
  --target-group-arn target-group-arn \
  --attributes "Key=load_balancing.cross_zone.enabled,Value=true"
```

## CloudFormation

Untuk mengaktifkan penyeimbangan beban lintas zona

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya untuk menyertakan `load_balancing.cross_zone.enabled` atribut.

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
    Properties:
      Name: my-target-group
      Protocol: HTTP
      Port: 80
      TargetType: ip
      VpcId: !Ref myVPC
      TargetGroupAttributes:
        - Key: "load_balancing.cross_zone.enabled"
          Value: "true"
```

## Bobot Target Otomatis (ATW)

Automatic Target Weights (ATW) secara konstan memonitor target yang menjalankan aplikasi Anda, mendeteksi penyimpangan kinerja yang signifikan, yang dikenal sebagai anomali. ATW menyediakan kemampuan untuk secara dinamis menyesuaikan jumlah lalu lintas yang diarahkan ke target, melalui deteksi anomali data waktu nyata.

Automatic Target Weights (ATW) melakukan deteksi anomali pada setiap Application Load Balancer di akun Anda secara otomatis. Ketika target anomali diidentifikasi, ATW dapat secara otomatis mencoba menstabilkannya dengan mengurangi jumlah lalu lintas yang dialihkan, yang dikenal sebagai mitigasi anomali. ATW terus mengoptimalkan distribusi lalu lintas untuk memaksimalkan tingkat keberhasilan per target sambil meminimalkan tingkat kegagalan kelompok sasaran.

Pertimbangan:

- Deteksi anomali saat ini memantau kode respons HTTP 5xx yang berasal dari, dan kegagalan koneksi ke, target Anda. Deteksi anomali selalu aktif dan tidak dapat dimatikan.
- ATW tidak didukung saat menggunakan Lambda sebagai target.

## Daftar Isi

- [Deteksi anomali](#)
- [Mitigasi anomali](#)

## Deteksi anomali

Deteksi anomali ATW memantau untuk setiap target yang menampilkan penyimpangan perilaku yang signifikan dari target lain dalam kelompok target mereka. Penyimpangan ini, yang disebut anomali, ditentukan dengan membandingkan persen kesalahan satu target dengan persen kesalahan target lain dalam kelompok sasaran. Kesalahan ini dapat berupa kesalahan koneksi dan kode kesalahan HTTP. Target yang melaporkan secara signifikan lebih tinggi daripada rekan-rekan mereka kemudian dianggap anomali.

Deteksi anomali membutuhkan minimal tiga target sehat dalam kelompok sasaran. Ketika target terdaftar ke kelompok sasaran, target harus lulus pemeriksaan kesehatan sebelum menerima lalu lintas. Setelah target mulai menerima lalu lintas, ATW mulai memantau target dan terus menerbitkan hasil anomali. Untuk target tanpa anomali, hasil anomali adalah `normal`. Untuk target dengan anomali, hasil anomali adalah `anomalous`.

Deteksi anomali ATW bekerja secara independen dari pemeriksaan kesehatan kelompok sasaran. Target dapat melewati semua pemeriksaan kesehatan kelompok sasaran, tetapi masih ditandai anomali karena tingkat kesalahan yang meningkat. Target yang menjadi anomali tidak mempengaruhi status pemeriksaan kesehatan kelompok sasaran mereka.

### Status deteksi anomali

Anda dapat melihat status deteksi anomali saat ini. Berikut ini adalah nilai yang mungkin:

- `normal`Tidak ada anomali yang terdeteksi.
- `anomalous`Anomali terdeteksi.

### Console

Untuk melihat status deteksi anomali

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan detailnya.
4. Pilih tab Target.

5. Dalam tabel Target terdaftar, kolom hasil deteksi anomali menampilkan status anomali setiap target.

## AWS CLI

Untuk melihat status deteksi anomali

Gunakan perintah [describe-target-health](#). Contoh berikut menampilkan status untuk setiap target dalam kelompok target yang ditentukan.

```
aws elbv2 describe-target-health \  
  --target-group-arn target-group-arn \  
  --include AnomalyDetection
```

## Mitigasi anomali

Mitigasi anomali ATW mengarahkan lalu lintas menjauh dari target anomali secara otomatis, memberi mereka kesempatan untuk pulih.

### Persyaratan

Fungsi mitigasi anomali ATW hanya tersedia saat menggunakan algoritma perutean acak tertimbang.

Selama mitigasi:

- ATW secara berkala menyesuaikan jumlah lalu lintas yang diarahkan ke target anomali. Saat ini, periodenya setiap lima detik.
- ATW mengurangi jumlah lalu lintas yang diarahkan ke target anomali ke jumlah minimum yang diperlukan untuk melakukan mitigasi anomali.
- Target yang tidak lagi terdeteksi sebagai anomali secara bertahap akan memiliki lebih banyak lalu lintas yang diarahkan ke mereka sampai mereka mencapai paritas dengan target normal lainnya dalam kelompok sasaran.

## Console

Untuk mengaktifkan mitigasi anomali

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.

3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Di panel konfigurasi Lalu lintas, verifikasi bahwa nilai yang dipilih untuk algoritma Load balancing adalah Weighted random.

Ketika algoritma acak tertimbang awalnya dipilih, deteksi anomali diaktifkan secara default.

6. Di bawah mitigasi anomali, pastikan bahwa Aktifkan mitigasi anomali dipilih.
7. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaktifkan mitigasi anomali

Gunakan [modify-target-group-attributes](#) perintah dengan `load_balancing.algorithm.anomaly_mitigation` atribut.

```
aws elbv2
```

## Status mitigasi

Anda dapat memeriksa apakah ATW melakukan mitigasi pada target. Berikut ini adalah nilai yang mungkin:

- `yes` Mitigasi sedang berlangsung.
- `no` Mitigasi tidak sedang berlangsung.

## Console

Untuk melihat status mitigasi anomali

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan detailnya.
4. Pilih tab Target.
5. Dalam tabel Target terdaftar, Anda dapat melihat status mitigasi anomali setiap target di kolom Mitigasi berlaku.

## AWS CLI

Untuk melihat status mitigasi anomali

Gunakan perintah [describe-target-health](#). Contoh berikut menampilkan status untuk setiap target dalam kelompok target yang ditentukan.

```
aws elbv2 describe-target-health \  
  --target-group-arn target-group-arn \  
  --include AnomalyDetection
```

## Sesi lengket

Secara default, Application Load Balancer merutekan setiap permintaan secara independen ke target terdaftar berdasarkan algoritma load-balancing yang dipilih. Namun, Anda dapat menggunakan fitur sesi lekat (juga dikenal sebagai sesi afinitas atau sesi gabungan) untuk mengaktifkan penyeimbang beban untuk mengikat sesi pengguna ke target tertentu. Hal ini memastikan bahwa semua permintaan dari pengguna selama sesi dikirim ke target yang sama. Fitur ini berguna untuk server yang mempertahankan informasi negara untuk memberikan pengalaman terus-menerus untuk klien. Untuk menggunakan sesi lekat, klien harus mendukung cookie.

Application Load Balancer mendukung cookie berbasis durasi dan cookie berbasis aplikasi. Sesi lekat diaktifkan pada tingkat kelompok target. Anda dapat menggunakan kombinasi kekakuan berbasis durasi, kelengketan berbasis aplikasi, dan tidak lengket di seluruh grup target Anda.

Kunci untuk mengelola sesi lekat adalah menentukan berapa lama penyeimbang beban Anda harus secara konsisten mengarahkan permintaan pengguna ke target yang sama. Jika aplikasi Anda memiliki cookie sesi sendiri, maka Anda dapat menggunakan kekakuan berbasis aplikasi dan cookie sesi penyeimbang beban mengikuti durasi yang ditentukan oleh cookie sesi aplikasi. Jika aplikasi Anda tidak memiliki cookie sesi sendiri, maka Anda dapat menggunakan lengket berbasis durasi untuk menghasilkan cookie sesi penyeimbang beban dengan durasi yang Anda tentukan.

Isi cookie yang dihasilkan penyeimbang beban dienkripsi menggunakan tombol berputar. Anda tidak dapat mendekripsi atau memodifikasi cookie yang dihasilkan penyeimbang beban.

Untuk kedua jenis lengket, Application Load Balancer mengatur ulang berakhirnya cookie yang dihasilkannya setelah setiap permintaan. Jika cookie berakhir, sesi tidak lagi lekat dan klien harus menghapus cookie dari toko cookie.

## Persyaratan

- Penyeimbang HTTP/HTTPS beban.
- Setidaknya satu contoh sehat di setiap Availability Zone.

## Pertimbangan-pertimbangan

- Sesi lengket tidak didukung jika [penyeimbangan beban lintas zona](#) dinonaktifkan. Upaya untuk mengaktifkan sesi lengket sementara penyeimbangan beban lintas zona dinonaktifkan gagal.
- Untuk cookie berbasis aplikasi, nama cookie harus ditentukan secara individual untuk setiap kelompok target. Namun, untuk cookie berbasis durasi, AWSALB adalah satu-satunya nama yang digunakan di semua kelompok target.
- Jika Anda menggunakan beberapa lapisan Balancers Beban Aplikasi, Anda dapat mengaktifkan sesi yang lekat di semua lapisan dengan cookie berbasis aplikasi. Namun, dengan cookie berbasis durasi, Anda dapat mengaktifkan sesi lengket hanya pada satu lapisan, karena AWSALB adalah satu-satunya nama yang tersedia.
- Jika Application Load Balancer menerima cookie lengket AWSALB berbasis AWSALBCORS dan durasi, nilai dalam akan diutamakan. AWSALBCORS
- Stickiness berbasis aplikasi tidak bekerja dengan kelompok target tertimbang.
- Jika Anda memiliki [Tindakan ke depan](#) dengan beberapa kelompok target, dan sesi lengket diaktifkan untuk satu atau lebih kelompok target, Anda harus mengaktifkan kelekatan di tingkat grup target.
- WebSocket koneksi secara inheren lengket. Jika klien meminta upgrade koneksi ke WebSockets, target yang mengembalikan kode status HTTP 101 untuk menerima upgrade koneksi adalah target yang digunakan dalam WebSockets koneksi. Setelah WebSockets upgrade selesai, kekakuan berbasis cookie tidak digunakan.
- Application Load Balancers menggunakan `Expires` atribut dalam header cookie bukan `Max-Age` atribut.
- Application Load Balancers tidak mendukung nilai-nilai cookie yang URL dikodekan.
- Jika Application Load Balancer menerima permintaan baru saat target terkuras karena deregistrasi, permintaan dialihkan ke target yang sehat.
- Sesi lengket tidak didukung jika pengoptimal target diaktifkan.

## Jenis lengket

- [Kelekatan berbasis durasi](#)
- [Kelekatan berbasis aplikasi](#)

## Kelekatan berbasis durasi

Rute lekat berbasis durasi meminta target yang sama di grup target menggunakan cookie yang dihasilkan load balancer (AWSALB). Cookie ini digunakan untuk memetakan sesi ke target. Jika aplikasi Anda tidak memiliki cookie sesi sendiri, Anda dapat menentukan durasi lekat Anda sendiri dan mengelola berapa lama load balancer Anda harus secara konsisten mengarahkan permintaan pengguna ke target yang sama.

Ketika load balancer pertama kali menerima permintaan dari klien, load balancer merutekan permintaan ke target (berdasarkan algoritma yang dipilih), dan menghasilkan cookie bernama AWSALB. Ini mengkodekan informasi tentang target yang dipilih, mengenkripsi cookie, dan melibatkan cookie dalam menanggapi klien. Load balancer yang dihasilkan cookie memiliki kadaluwarsa sendiri 7 hari yang tidak dapat dikonfigurasi.

Dalam permintaan berikutnya, klien harus mencakup cookie AWSALB. Ketika load balancer menerima permintaan dari klien yang berisi cookie, mendeteksi dan rute permintaan ke target yang sama. Jika cookie ada tetapi tidak dapat diterjemahkan, atau jika mengacu pada target yang tidak terdaftar atau tidak sehat, penyeimbang beban memilih target baru dan memperbarui cookie dengan informasi tentang target baru.

Untuk permintaan berbagi sumber daya lintas asal (CORS), beberapa browser `SameSite=None`; `Secure` perlu mengaktifkan kekakuan. Untuk mendukung browser ini, penyeimbang beban selalu menghasilkan cookie lengket kedua `AWSALBCORS`, yang mencakup informasi yang sama dengan cookie lengket asli, serta atributnya. `SameSite` Klien menerima kedua cookie, termasuk permintaan non-CORS.

## Console

Untuk mengaktifkan kelengketan berbasis durasi

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Di bawah Konfigurasi pemilihan Target, lakukan hal berikut:

- a. Pilih Nyalakan lengket.
  - b. Untuk Jenis kelekatan Pilih Cookies yang dihasilkan load balancer.
  - c. Untuk Durasi kelekatan, tentukan nilai antara 1 detik dan 7 hari.
6. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaktifkan kelekatan berbasis durasi

Gunakan [modify-target-group-attributes](#) perintah dengan `stickiness.lb_cookie.duration_seconds` atribut `stickiness.enabled` dan.

```
aws elbv2 modify-target-group-attributes \  
  --target-group-arn target-group-arn \  
  --attributes \  
    "Key=stickiness.enabled,Value=true" \  
    "Key=stickiness.lb_cookie.duration_seconds,Value=300"
```

## CloudFormation

Untuk mengaktifkan kelekatan berbasis durasi

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya untuk menyertakan `stickiness.enabled` dan `stickiness.lb_cookie.duration_seconds` atribut.

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      Protocol: HTTP  
      Port: 80  
      TargetType: ip  
      VpcId: !Ref myVPC  
      TargetGroupAttributes:  
        - Key: "stickiness.enabled"  
          Value: "true"  
        - Key: "stickiness.lb_cookie.duration_seconds"  
          Value: "300"
```

## Kelekatan berbasis aplikasi

Stickiness berbasis aplikasi memberi Anda fleksibilitas untuk menetapkan kriteria Anda sendiri untuk kelekatan target klien. Bila Anda mengaktifkan kelekatan berbasis aplikasi, penyeimbang beban akan mengarahkan permintaan pertama ke target dalam grup target berdasarkan algoritme yang dipilih. Target diharapkan untuk menetapkan cookie aplikasi kustom yang cocok dengan cookie yang dikonfigurasi pada penyeimbang beban untuk mengaktifkan kelekatan. Cookie kustom ini dapat mencakup salah satu atribut cookie yang diperlukan oleh aplikasi.

Ketika Application Load Balancer menerima cookie aplikasi kustom dari target, maka secara otomatis menghasilkan cookie aplikasi terenkripsi baru untuk menangkap informasi sesi kelekatan. Cookie aplikasi yang dihasilkan load balancer ini menangkap informasi lekat untuk setiap grup target yang mengaktifkan kelekatan berbasis aplikasi.

Cookie aplikasi yang dihasilkan load balancer tidak menyalin atribut cookie kustom yang ditetapkan oleh target. Cookie aplikasi ini akan berakhir dengan sendirinya dalam 7 hari yang tidak dapat dikonfigurasi. Dalam menanggapi klien, Application Load Balancer hanya memvalidasi nama yang dikonfigurasi cookie kustom pada tingkat kelompok target dan bukan nilai atau atribut kadaluwarsa cookie kustom. Selama nama cocok, load balancer mengirimkan kedua cookie, cookie kustom yang ditetapkan oleh target, dan cookie aplikasi yang dihasilkan oleh load balancer, dalam menanggapi klien.

Dalam permintaan berikutnya, klien harus mengirim kembali kedua cookie untuk mempertahankan kelekatan atau sesi afinitas. Load balancer mendekripsi cookie aplikasi, dan memeriksa apakah durasi lekat yang dikonfigurasi masih berlaku. Kemudian informasi dalam cookie digunakan untuk mengirim permintaan ke target yang sama dalam kelompok target untuk mempertahankan kelekatan. Load balancer juga proxy cookie aplikasi kustom ke target tanpa memeriksa atau memodifikasinya. Dalam tanggapan berikutnya, berakhirnya load balancer yang dihasilkan cookie aplikasi dan durasi kelekatan yang dikonfigurasi pada load balancer diatur ulang. Untuk menjaga kelekatan antara klien dan target, kedaluwarsanya cookie, dan durasi kelekatan seharusnya tidak terlewat.

Jika target gagal atau menjadi tidak sehat, load balancer akan berhenti merutekan permintaan ke target tersebut, dan memilih target baru yang sehat berdasarkan algoritma load balancing yang dipilih. Load balancer memperlakukan sesi tersebut seakan “terjebak” ke target baru yang sehat, dan terus merutekan permintaan ke target sehat yang baru bahkan jika target yang gagal kembali.

Dengan permintaan cross-origin resource sharing (CORS), untuk mengaktifkan kelekatan, load balancer menambahkan `SameSite=None; Secure` atribut ke cookie aplikasi yang dihasilkannya hanya jika versi agen pengguna adalah Chromium80 ke atas.

Karena sebagian besar browser membatasi cookie hingga ukuran 4K, penyeimbang beban memecah cookie aplikasi lebih besar dari 4K menjadi beberapa cookie. Application Load Balancer mendukung cookie hingga 16K dalam ukuran dan karena itu dapat membuat hingga 4 pecahan yang dikirimkan ke klien. Nama cookie aplikasi yang dilihat klien dimulai dengan "AWSALBAPP-" dan termasuk nomor fragmen. Misalnya, jika ukuran cookie 0-4K, klien melihat AWSALBAPP -0. Jika ukuran cookie 4-8k, klien melihat AWSALBAPP -0 dan AWSALBAPP -1, dan seterusnya.

## Console

Untuk mengaktifkan kelengkapan berbasis aplikasi

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Di bawah Konfigurasi pemilihan Target, lakukan hal berikut:
  - a. Pilih Nyalakan lengket.
  - b. Untuk Jenis kelekatan Pilih Cookie berbasis aplikasi.
  - c. Untuk Durasi lengket, tentukan nilai antara 1 detik dan 7 hari.
  - d. Untuk Nama cookie aplikasi, masukkan nama untuk cookie berbasis aplikasi Anda.

Jangan gunakan AWSALB, AWSALBAPP, atau AWSALBTG untuk nama cookie; karena sudah dicadangkan untuk digunakan oleh load balancer.

6. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaktifkan kelengkapan berbasis aplikasi

Gunakan [modify-target-group-attributes](#) perintah dengan atribut berikut:

- `stickiness.enabled`
- `stickiness.type`
- `stickiness.app_cookie.cookie_name`
- `stickiness.app_cookie.duration_seconds`

```
aws elbv2 modify-target-group-attributes \  
  --target-group-arn target-group-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=300"
```

## CloudFormation

Untuk mengaktifkan kelengketan berbasis aplikasi

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya untuk menyertakan atribut berikut:

- `stickiness.enabled`
- `stickiness.type`
- `stickiness.app_cookie.cookie_name`
- `stickiness.app_cookie.duration_seconds`

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      Protocol: HTTP  
      Port: 80  
      TargetType: ip  
      VpcId: !Ref myVPC  
      TargetGroupAttributes:  
        - 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: "300"
```

## Penyeimbangan ulang manual

Saat meningkatkan skala, jika jumlah target meningkat secara signifikan, ada potensi distribusi beban yang tidak merata karena afinitas. Dalam skenario ini, Anda dapat menyeimbangkan beban pada target Anda menggunakan dua pilihan berikut:

- Mengatur kadaluwarsa pada cookie yang dihasilkan oleh aplikasi sebelum tanggal dan waktunya. Ini mencegah klien mengirim cookie ke Application Load Balancer, yang akan memulai kembali proses pembentukan lengket.
- Tetapkan durasi singkat pada konfigurasi lengket berbasis aplikasi penyeimbang beban; misalnya, 1 detik. Ini memaksa Application Load Balancer untuk membangun kembali kekakuan meskipun cookie yang ditetapkan oleh target tidak kadaluwarsa.

## Daftarkan target dengan kelompok sasaran Application Load Balancer Anda

Anda mendaftarkan target Anda dengan grup target. Bila Anda membuat grup target, Anda menentukan jenis targetnya, yang menentukan bagaimana Anda mendaftarkan targetnya. Misalnya, Anda dapat mendaftarkan instance IDs, alamat IP, atau fungsi Lambda. Untuk informasi selengkapnya, lihat [Kelompok-kelompok target untuk Application Load Balancers](#).

Jika permintaan pada target Anda saat ini terdaftar meningkat, Anda dapat mendaftarkan target tambahan untuk menangani permintaan. Ketika target Anda siap untuk menangani permintaan, daftarkan ke grup target Anda. Load balancer mulai routing permintaan ke target segera setelah proses pendaftaran selesai dan target melewati pemeriksaan kesehatan awal.

Jika permintaan pada target terdaftar Anda menurun, atau Anda perlu untuk melayani target, Anda dapat membatalkan pendaftaran dari kelompok target Anda. Load balancer berhenti routing permintaan ke target segera setelah Anda membatalkan pendaftaran. Ketika target siap untuk menerima permintaan, Anda dapat mendaftarkannya dengan kelompok target lagi.

Saat Anda deregistrasi target, load balancer menunggu hingga permintaan dalam penerbangan selesai. Hal ini dikenal sebagai Pengurusan koneksi. Status target adalah `draining` sementara koneksi pengeringan sedang berlangsung.

Ketika Anda membatalkan pendaftaran (deregister) target yang telah terdaftar oleh alamat IP, Anda harus menunggu penundaan pembatalan untuk selesai sebelum Anda dapat mendaftarkan alamat IP yang sama lagi.

Jika Anda mendaftarkan target berdasarkan ID instans, Anda dapat menggunakan load balancer dengan grup Auto Scaling. Setelah Anda melampirkan grup target ke grup Auto Scaling dan skala grup keluar, contoh yang diluncurkan oleh grup Auto Scaling terdaftar dengan grup target. Jika Anda memisahkan grup target dari grup Auto Scaling, maka instans tersebut secara otomatis dihapus dari grup target. Untuk Informasi Selengkapnya, Lihat [Memasang load balancer to your Auto Scaling group](#) pada Amazon EC2 Auto Scaling User Guide.

Saat mematikan aplikasi pada target, Anda harus terlebih dahulu membatalkan pendaftaran target dari grup targetnya dan memberikan waktu untuk koneksi yang ada terkuras. Anda dapat memantau status deregistrasi menggunakan perintah describe-target-health CLI, atau dengan menyegarkan tampilan grup target di Konsol Manajemen AWS. Setelah mengonfirmasi target dideregistrasi, Anda dapat melanjutkan dengan menghentikan atau mengakhiri aplikasi. Urutan ini mencegah pengguna mengalami kesalahan 5XX saat aplikasi dihentikan saat masih memproses lalu lintas.

## Menargetkan grup keamanan

Saat Anda mendaftarkan instans EC2 sebagai target, Anda harus memastikan keamanan grup agar memungkinkan bagi load balancer untuk mengkomunikasikan dengan instans anda baik pada port pendengar dan port pemeriksaan kesehatan.

Aturan yang disarankan

Inbound

Source	Port Range	Comment
<i>load balancer security group</i>	<i>instance listener</i>	Izinkan lalu lintas dari penyeimbang beban pada port pendengar instance
<i>load balancer security group</i>	<i>health check</i>	Izinkan lalu lintas dari penyeimbang beban di port pemeriksaan kesehatan

Kami juga merekomendasikan Anda untuk mengizinkan inbound ICMP lalu lintas untuk mendukung jalan MTU penemuan. Untuk informasi selengkapnya, lihat [Path MTU Discovery](#) di Panduan Pengguna Amazon EC2.

## Pengoptimal Target

Pengoptimal target memungkinkan Anda menerapkan konkurensi yang ketat pada target dalam grup target. Ia bekerja dengan bantuan agen yang Anda instal dan konfigurasi pada target. Agen berfungsi sebagai proxy inline antara penyeimbang beban dan aplikasi Anda. Anda mengonfigurasi agen untuk menerapkan jumlah maksimum permintaan bersamaan yang dapat dikirim penyeimbang beban ke target. Agen melacak jumlah permintaan yang diproses target. Ketika angka jatuh di bawah nilai maksimum yang dikonfigurasi, agen mengirimkan sinyal ke penyeimbang beban yang memberitahukan bahwa target siap untuk memproses permintaan lain.

Untuk mengaktifkan pengoptimal target, Anda menentukan port kontrol target saat membuat grup target. Penyeimbang beban menetapkan saluran kontrol dengan agen di port ini untuk lalu lintas manajemen. Port ini berbeda dengan port tempat load balancer mengirimkan lalu lintas aplikasi. Target yang terdaftar pada kelompok sasaran harus memiliki agen yang menjalankannya.

Catatan: Pengoptimal target hanya dapat diaktifkan selama pembuatan grup target. Port kontrol target tidak dapat dimodifikasi setelah pembuatan.

Agan tersedia sebagai gambar Docker di: `public.ecr.aws/aws-elb/target-optimizer/target-control-agent:latest`. Anda mengonfigurasi variabel lingkungan berikut saat menjalankan wadah agen:

### TARGET\_CONTROL\_DATA\_ADDRESS

Agan menerima lalu lintas aplikasi dari penyeimbang beban pada soket ini (IP: port). Port di soket ini adalah port lalu lintas aplikasi yang Anda konfigurasi untuk grup target. Secara default, agen dapat menerima koneksi plaintext dan TLS.

### TARGET\_CONTROL\_CONTROL\_ADDRESS

Agan menerima lalu lintas manajemen dari penyeimbang beban pada soket ini (IP: port). Port di soket adalah port kontrol target yang Anda konfigurasi untuk grup target.

### TARGET\_CONTROL\_DESTINATION\_ADDRESS

Agan proksi lalu lintas aplikasi ke soket ini (IP: port). Aplikasi Anda harus mendengarkan di soket ini.

### (Opsional) TARGET\_CONTROL\_MAX\_CONCURRENCY

Jumlah maksimum permintaan bersamaan yang akan diterima target dari penyeimbang beban. Bisa antara 0-1000. Default-nya adalah 1.

### (Opsional) TARGET\_CONTROL\_TLS\_CERT\_PATH

Lokasi sertifikat TLS yang diberikan agen kepada penyeimbang beban selama jabat tangan TLS. Secara default, agen menghasilkan sertifikat yang ditandatangani sendiri dalam memori.

### (Opsional) TARGET\_CONTROL\_TLS\_KEY\_PATH

Lokasi kunci pribadi yang sesuai dengan sertifikat TLS yang diberikan agen kepada penyeimbang beban selama jabat tangan TLS. Secara default, agen menghasilkan kunci pribadi dalam memori.

### (Opsional) TARGET\_CONTROL\_TLS\_SECURITY\_POLICY

Kebijakan keamanan ELB yang Anda konfigurasi untuk grup target. Nilai default-nya `ELBSecurityPolicy-2016-08`.

### (Opsional) TARGET\_CONTROL\_PROTOCOL\_VERSION

Protokol di mana penyeimbang beban berkomunikasi dengan agen. Nilai yang mungkin adalah `HTTP1`, `HTTP2`, `GRPC`. Nilai default-nya `HTTP1`.

### (Opsional) RUST\_LOG

Tingkat log dari proses agen. Perangkat lunak agen ditulis dalam Rust. Nilai yang mungkin adalah `debug`, `info`, dan `error`. Nilai default-nya `info`.

Untuk memodifikasi nilai variabel lingkungan apa pun, Anda harus me-restart agen dengan nilai baru. Anda dapat memantau pengoptimal target dengan metrik berikut: `TargetControlRequestCount`, `TargetControlRequestRejectCount`, `TargetControlActiveConnectionsCount`, `TargetControlNewChannelCount`, `TargetControlChannelErrorCount`, `TargetControlWorkQueueLength`, `TargetControlProcessedBytes` [Untuk informasi selengkapnya, lihat Metrik pengoptimal target Untuk informasi pemecahan masalah, lihat Mengatasi masalah pengoptimal target](#)

## Subnet bersama

Peserta dapat membuat Application Load Balancer di VPC bersama. Peserta tidak dapat mendaftarkan target yang berjalan di subnet yang tidak dibagikan dengan mereka.

## Daftarkan target

Setiap grup target harus memiliki setidaknya satu target yang terdaftar di setiap Availability Zone yang diaktifkan untuk penyeimbang beban.

Jenis target dari grup target Anda menentukan bagaimana Anda mendaftarkan target dengan grup target tersebut. Untuk informasi selengkapnya, lihat [Tipe target](#).

### Persyaratan dan pertimbangan

- Suatu instans harus berada di negara running saat Anda mendaftarkannya.
- Instance target harus berada di virtual private cloud (VPC) yang Anda tentukan untuk grup target.
- Saat mendaftarkan target dengan ID instans untuk grup IPv6 target, target harus memiliki IPv6 alamat utama yang ditetapkan. Untuk mempelajari lebih lanjut, lihat [IPv6 alamat](#) di Panduan Pengguna Amazon EC2
- Saat mendaftarkan target berdasarkan alamat IP untuk grup IPv4 target, alamat IP yang Anda daftarkan harus dari salah satu blok CIDR berikut:
  - Subnet dari kelompok target VPC
  - 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)
- Saat mendaftarkan target berdasarkan alamat IP untuk grup IPv6 target, alamat IP yang Anda daftarkan harus berada di dalam blok CIDR VPC atau di dalam blok IPv6 CIDR dari VPC IPv6 peered.
- Anda tidak dapat mendaftarkan alamat IP Application Load Balancer lain di VPC yang sama. Jika Application Load Balancer lainnya ada di VPC yang mengintip ke VPC load balancer, Anda dapat mendaftarkan alamat IP-nya.

### Console

#### Untuk mendaftarkan target

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan detailnya.
4. Pilih tab Target.
5. Pilih Daftarkan target.
6. Jika jenis target grup target adalah instance, pilih instance yang tersedia, ganti port default jika diperlukan, lalu pilih Sertakan sebagai tertunda di bawah ini.

7. Jika jenis target grup target adalah `ip`, untuk setiap alamat IP, pilih jaringan, masukkan alamat IP dan port, dan pilih Sertakan sebagai tertunda di bawah ini.
8. Jika jenis target dari grup target adalah `lambda`, pilih fungsi Lambda atau masukkan ARN-nya. Untuk informasi selengkapnya, lihat [Gunakan fungsi Lambda sebagai target](#).
9. Pilih Daftarkan target yang tertunda.

## AWS CLI

Untuk mendaftarkan target

Gunakan perintah [register-target](#). Contoh berikut mendaftarkan target dengan ID instance. Karena port tidak ditentukan, penyeimbang beban menggunakan port grup target.

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

Contoh berikut mendaftarkan target dengan alamat IP. Karena port tidak ditentukan, penyeimbang beban menggunakan port grup target.

```
aws elbv2 register-targets \
  --target-group-arn target-group-arn \
  --targets Id=10.0.50.10 Id=10.0.50.20
```

Contoh berikut mendaftarkan fungsi Lambda sebagai target.

```
aws elbv2 register-targets \
  --target-group-arn target-group-arn \
  --targets Id=lambda-function-arn
```

## CloudFormation

Untuk mendaftarkan target

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya untuk memasukkan target baru. Contoh berikut mendaftarkan dua target dengan ID instance.

```
Resources:
```

```
myTargetGroup:
  Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
  Properties:
    Name: my-target-group
    Protocol: HTTP
    Port: 80
    TargetType: instance
    VpcId: !Ref myVPC
    Targets:
      - Id: !GetAtt Instance1.InstanceId
        Port: 80
      - Id: !GetAtt Instance2.InstanceId
        Port: 80
```

## Target deregister

Jika permintaan pada aplikasi Anda menurun, atau jika Anda perlu melayani target Anda, Anda dapat membatalkan pendaftaran target dari grup target Anda. Proses deregisterasi target menghapus itu dari kelompok target Anda, tetapi tidak mempengaruhi target sebaliknya.

### Console

Untuk membatalkan pendaftaran target

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Target, pilih target yang akan dihapus.
5. Pilih Batalkan pendaftaran.
6. Ketika konfirmasi diminta, pilih Batalkan Pendaftaran.

### AWS CLI

Untuk membatalkan pendaftaran target

Gunakan perintah [Target deregister](#). Contoh berikut deregister dua target yang terdaftar oleh ID instance.

```
aws elbv2 deregister-targets \
```

```
--target-group-arn target-group-arn \  
--targets Id=i-1234567890abcdef0 Id=i-0abcdef1234567890
```

## Gunakan fungsi Lambda sebagai target Application Load Balancer

Anda dapat mendaftarkan fungsi Lambda Anda sebagai target dan mengkonfigurasi aturan pendengar untuk meneruskan permintaan ke kelompok target untuk fungsi Lambda Anda. Ketika load balancer meneruskan permintaan ke kelompok target dengan fungsi Lambda sebagai target, ia memanggil fungsi Lambda Anda dan melewati isi dari permintaan ke fungsi Lambda, dalam format JSON.

Load balancer memanggil fungsi Lambda secara langsung alih-alih menggunakan koneksi jaringan. Oleh karena itu, tidak ada persyaratan untuk aturan keluar dari kelompok keamanan Application Load Balancer.

### Batas

- Fungsi Lambda dan kelompok target harus dalam akun dan di wilayah yang sama.
- Ukuran maksimum tubuh permintaan yang dapat Anda kirim ke fungsi Lambda adalah 1 MB. Untuk batas ukuran terkait, lihat [Batas header HTTP](#).
- Ukuran maksimum respon JSON bahwa fungsi Lambda dapat mengirim 1 MB.
- WebSockets tidak didukung. Permintaan upgrade ditolak dengan kode HTTP 400.
- Local Zones tidak didukung.
- Timbangan Target Otomatis (ATW) tidak didukung.

### Daftar Isi

- [Siapkan fungsi Lambda](#)
- [Buat grup target untuk fungsi Lambda](#)
- [Menerima peristiwa dari load balancer](#)
- [Menanggapi load balancer](#)
- [Header nilai ganda](#)
- [Aktifkan pemeriksaan kesehatan](#)
- [Daftarkan fungsi Lambda](#)
- [Deregistrasi fungsi Lambda](#)

Untuk demo, lihat [Target Lambda pada Application Load Balancer](#).

## Siapkan fungsi Lambda

Rekomendasi berikut berlaku jika Anda menggunakan fungsi Lambda Anda dengan Application Load Balancer.

Izin untuk mengaktifkan fungsi Lambda

Jika Anda membuat kelompok target dan mendaftarkan fungsi Lambda menggunakan Konsol Manajemen AWS, konsol menambahkan izin yang diperlukan untuk kebijakan fungsi Lambda Anda atas nama Anda. Jika tidak, setelah Anda membuat grup target dan mendaftarkan fungsi menggunakan AWS CLI, Anda harus menggunakan perintah [add-permission untuk memberikan izin](#) Elastic Load Balancing untuk menjalankan fungsi Lambda Anda. Kami menyarankan Anda menggunakan tombol `aws:SourceAccount` dan `aws:SourceArn` kondisi untuk membatasi pemanggilan fungsi ke grup target yang ditentukan. Untuk informasi selengkapnya, lihat [Masalah deputy yang membingungkan](#) di Panduan Pengguna IAM,

```
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
```

## Versioning fungsi Lambda

Anda dapat mendaftarkan satu fungsi Lambda per kelompok target. Untuk memastikan bahwa Anda dapat mengubah fungsi Lambda Anda dan bahwa load balancer selalu memanggil versi terkini dari fungsi Lambda, membuat alias fungsi dan menyertakan alias dalam fungsi ARN ketika Anda mendaftarkan fungsi Lambda dengan load balancer. Untuk informasi selengkapnya, lihat [alias AWS Lambda fungsi](#) di Panduan AWS Lambda Pengembang.

## Fungsi waktu habis

Load balancer menunggu sampai fungsi Lambda Anda merespons atau kehabisan waktu. Kami merekomendasikan Anda untuk mengkonfigurasi timeout pada fungsi Lambda didasarkan pada waktu penggunaan yang diperkirakan. Untuk informasi tentang nilai batas waktu default dan cara mengubahnya, lihat [Mengonfigurasi batas waktu fungsi Lambda](#). Untuk informasi tentang nilai batas waktu maksimum yang dapat Anda konfigurasi, lihat [AWS Lambda kuota](#).

## Buat grup target untuk fungsi Lambda

Buat grup target, yang digunakan dalam routing permintaan. Jika konten permintaan cocok dengan aturan pendengar dengan tindakan untuk meneruskannya ke kelompok target ini, load balancer memacu fungsi Lambda yang telah terdaftar.

### Console

Untuk membuat grup target dan mendaftarkan fungsi Lambda

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih Buat grup target.
4. Untuk Pilih jenis target Pilih Fungsi Lambda.
5. Untuk Name, masukkan nama untuk grup target.
6. (Opsional) Untuk mengaktifkan pemeriksaan kesehatan, pilih Mengaktifkan pemeriksaan Health Bagian.
7. (Opsional) Perluas Tag. Untuk setiap tag, pilih Tambahkan tag baru dan masukkan kunci tag dan nilai tag.
8. Pilih Berikutnya.
9. Jika Anda siap untuk mendaftarkan fungsi Lambda, pilih Pilih fungsi Lambda dan pilih fungsi Lambda dari daftar, atau pilih Masukkan fungsi Lambda ARN dan masukkan ARN dari fungsi Lambda,

Jika Anda belum siap untuk mendaftarkan fungsi Lambda, pilih fungsi Register Lambda nanti dan daftarkan target nanti. Untuk informasi selengkapnya, lihat [the section called “Daftarkan target”](#).

10. Pilih Buat grup target.

### AWS CLI

Untuk membuat kelompok target dari tipe lambda

Gunakan perintah [create-target-group](#).

```
aws elbv2 create-target-group \
```

```
--name my-target-group \  
--target-type lambda
```

Untuk mendaftarkan fungsi Lambda

Gunakan perintah [register-target](#).

```
aws elbv2 register-targets \  
--target-group-arn target-group-arn \  
--targets Id=lambda-function-arn
```

## CloudFormation

Untuk membuat grup target dan mendaftarkan fungsi Lambda

Tentukan sumber daya tipe [AWS::ElasticLoadBalancingV2::TargetGroup](#). Jika Anda belum siap untuk mendaftarkan fungsi Lambda sekarang, Anda dapat menghilangkan Targets properti dan menambahkannya nanti.

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      TargetType: lambda  
      Tags:  
        - Key: 'department'  
          Value: '123'  
      Targets:  
        - Id: !Ref myLambdaFunction
```

## Menerima peristiwa dari load balancer

Load balancer mendukung permohonan Lambda untuk permintaan atas HTTP dan HTTPS. Load balancer mengirimkan peristiwa dalam format JSON. Load balancer menambahkan header berikut untuk setiap permintaan: `X-Amzn-Trace-Id`, `X-Forwarded-For`, `X-Forwarded-Port`, dan `X-Forwarded-Proto`.

Jika `content-encoding` header hadir, load balancer Base64 mengkodekan tubuh dan memasang `isBase64Encoded` ke `true`.

Jika `content-encoding` header tidak hadir, encoding Base64 tergantung pada jenis konten. Untuk jenis berikut, penyeimbang beban mengirimkan tubuh apa adanya dan disetel `isBase64Encoded` ke `false`: `text/*`, `application/json`, `application/javascript`, and `application/xml`. Jika tidak, load balancer Base64 mengkodekan tubuh dan memasang `isBase64Encoded` ke `true`.

Berikut adalah contoh kasusnya.

```
{
  "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"
}
```

## Menanggapi load balancer

Respon dari fungsi Lambda Anda harus mencakup status encoding Base64, kode status, dan header. Anda bisa menghilangkan bagian tubuhnya.

Untuk memasukkan konten biner dalam tubuh respon, Anda harus mengkodekan Base64 konten dan mengatur `isBase64Encoded` ke `true`. Load balancer membaca kode konten untuk mengambil konten biner dan mengirimkannya ke klien dalam tubuh respon HTTP.

Penyeimbang beban tidak menghormati hop-by-hop header, seperti Connection atau Transfer-Encoding. Anda dapat menghilangkan header Content-Length karena load balancer menghitung sebelum mengirim tanggapan ke klien.

Berikut ini adalah contoh respon dari fungsi Lambda berbasis Node.js.

```
{
  "isBase64Encoded": false,
  "statusCode": 200,
  "statusDescription": "200 OK",
  "headers": {
    "Set-cookie": "cookies",
    "Content-Type": "application/json"
  },
  "body": "Hello from Lambda (optional)"
}
```

[Untuk template fungsi Lambda yang bekerja dengan Application Load Balancers, lihat application-load-balancer-serverless-app di github.](#) Atau, buka [konsol Lambda](#), pilih Aplikasi, Buat aplikasi, dan pilih salah satu dari berikut ini dari: AWS Serverless Application Repository

- ALB-Lambda-Target- S3 UploadFileto
- ALB-Lambda-Target BinaryResponse
- Target ALB-Lambda- IP WhatisMy

## Header nilai ganda

Jika permintaan dari klien atau tanggapan dari fungsi Lambda mengandung header dengan beberapa nilai atau berisi header yang sama beberapa kali, atau parameter permintaan (query) dengan beberapa nilai untuk kunci yang sama, Anda dapat mengaktifkan dukungan untuk sintaks header nilai ganda. Setelah Anda mengaktifkan header nilai ganda, header dan parameter query ditukarkan antara load balancer dan fungsi Lambda menggunakan array, bukan string. Jika Anda tidak mengaktifkan sintaks header nilai ganda dan header atau parameter query memiliki nilai ganda, load balancer menggunakan nilai terakhir yang diterima.

### Daftar Isi

- [Permintaan dengan header nilai ganda](#)
- [Respon dengan header nilai ganda](#)

- [Aktifkan header nilai ganda](#)

## Permintaan dengan header nilai ganda

Nama-nama bidang yang digunakan untuk header dan parameter string query berbeda tergantung apakah Anda mengaktifkan nilai ganda header untuk kelompok target.

Contoh permintaan berikut memiliki dua parameter query dengan tombol yang sama:

```
http://www.example.com?&myKey=val1&myKey=val2
```

Dengan format default, load balancer menggunakan nilai terakhir yang dikirim oleh klien dan mengirimkan sebuah peristiwa yang mencakup parameter string query menggunakan `queryStringParameters`. Sebagai contoh:

```
"queryStringParameters": { "myKey": "val2"},
```

Jika Anda mengaktifkan header nilai ganda, load balancer menggunakan kedua nilai kunci yang dikirim oleh klien dan mengirimkan sebuah peristiwa yang mencakup parameter string query menggunakan `multiValueQueryStringParameters`. Sebagai contoh:

```
"multiValueQueryStringParameters": { "myKey": ["val1", "val2"] },
```

Demikian pula, anggaplah bahwa klien mengirimkan permintaan dengan dua cookie di header:

```
"cookie": "name1=value1",  
"cookie": "name2=value2",
```

Dengan format default, load balancer menggunakan cookie terakhir yang dikirim oleh klien dan mengirimkan peristiwa yang mencakup header menggunakan `headers`. Sebagai contoh:

```
"headers": {  
  "cookie": "name2=value2",  
  ...  
},
```

Jika Anda mengaktifkan header nilai ganda, load balancer menggunakan kedua cookie yang dikirim oleh klien dan mengirimkan peristiwa yang mencakup header menggunakan `multiValueHeaders`. Sebagai contoh:

```
"multiValueHeaders": {
  "cookie": ["name1=value1", "name2=value2"],
  ...
},
```

Jika parameter permintaan dikodekan URL, maka load balancer tidak membaca kodenya. Anda harus memecahkan kode mereka dalam fungsi Lambda Anda.

## Respons dengan header nilai ganda

Nama-nama bidang yang digunakan untuk header berbeda tergantung pada apakah Anda mengaktifkan header nilai ganda untuk kelompok target. Anda harus menggunakan `multiValueHeaders` jika Anda telah mengaktifkan header nilai ganda dan `headers` sebaliknya.

Dengan format default, Anda dapat menentukan cookie tunggal:

```
{
  "headers": {
    "Set-cookie": "cookie-name=cookie-value;Domain=myweb.com;Secure;HttpOnly",
    "Content-Type": "application/json"
  },
}
```

Jika Anda mengaktifkan header nilai ganda, Anda harus menentukan beberapa cookie sebagai berikut:

```
{
  "multiValueHeaders": {
    "Set-cookie": ["cookie-name=cookie-
value;Domain=myweb.com;Secure;HttpOnly", "cookie-name=cookie-value;Expires=May 8,
2019"],
    "Content-Type": ["application/json"]
  },
}
```

Penyeimbang beban mungkin mengirim header ke klien dalam urutan yang berbeda dari urutan yang ditentukan dalam muatan respons Lambda. Oleh karena itu, jangan mengandalkan header yang dikembalikan dalam urutan tertentu.

## Aktifkan header nilai ganda

Anda dapat mengaktifkan atau menonaktifkan header nilai ganda untuk kelompok target dengan jenis `targetLambda`.

### Console

Untuk mengaktifkan header multi-nilai

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Aktifkan header Multi nilai.
6. Pilih Simpan perubahan.

### AWS CLI

Untuk mengaktifkan header multi-nilai

Gunakan `modify-target-group-attributes` perintah dengan `lambda.multi_value_headers.enabled` atribut.

```
aws elbv2 modify-target-group-attributes \  
  --target-group-arn target-group-arn \  
  --attributes "Key=lambda.multi_value_headers.enabled,Value=true"
```

### CloudFormation

Untuk mengaktifkan header multi-nilai

Perbarui `AWS::ElasticLoadBalancingV2::TargetGroup` sumber daya untuk menyertakan `lambda.multi_value_headers.enabled` atribut.

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      TargetType: lambda
```

```
Tags:
  - Key: 'department'
    Value: '123'
Targets:
  - Id: !Ref myLambdaFunction
TargetGroupAttributes:
  - Key: "lambda.multi_value_headers.enabled"
    Value: "true"
```

## Aktifkan pemeriksaan kesehatan

Secara default, pemeriksaan kesehatan dinonaktifkan untuk kelompok target jenis `lambda`. Anda dapat mengaktifkan pemeriksaan kesehatan untuk menerapkan DNS failover dengan Amazon Route 53. Fungsi `lambda` dapat memeriksa kesehatan layanan downstream sebelum menanggapi permintaan pemeriksaan kesehatan. Jika respons dari fungsi `lambda` menunjukkan kegagalan pemeriksaan kesehatan, kegagalan tersebut diteruskan ke Route 53. Anda dapat mengonfigurasi Route 53 agar gagal ke tumpukan aplikasi cadangan.

Anda dikenakan biaya untuk pemeriksaan kesehatan begitu juga untuk setiap panggilan fungsi `lambda`.

Berikut ini adalah format acara pemeriksaan kesehatan yang dikirim ke fungsi `lambda` Anda. Untuk memeriksa apakah suatu peristiwa adalah event pemeriksaan kesehatan, periksa nilai bidang `agent` pengguna. Agen pengguna untuk pemeriksaan kesehatan adalah `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
}
```

```
}
```

## Console

Untuk mengaktifkan pemeriksaan kesehatan untuk kelompok lambda sasaran

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Pemeriksaan kondisi, pilih Edit.
5. Untuk Pemeriksaan Kesehatan Pilih Aktifkan.
6. (Opsional) Perbarui pengaturan pemeriksaan kesehatan sesuai kebutuhan.
7. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaktifkan pemeriksaan kesehatan untuk kelompok lambda sasaran

Gunakan perintah [modify-target-group](#).

```
aws elbv2 modify-target-group \  
  --target-group-arn target-group-arn \  
  --health-check-enabled
```

## CloudFormation

Untuk mengaktifkan pemeriksaan kesehatan untuk kelompok lambda sasaran

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya.

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      TargetType: lambda  
      HealthCheckEnabled: true  
      Tags:  
        - Key: 'department'  
          Value: '123'
```

```
Targets:
  - Id: !Ref myLambdaFunction
```

## Daftarkan fungsi Lambda

Anda dapat mendaftarkan fungsi Lambda tunggal dengan masing-masing grup target. Untuk mengganti fungsi Lambda, kami sarankan Anda membuat grup target baru, mendaftarkan fungsi baru dengan grup target baru, dan memperbarui aturan listener untuk menggunakan grup target baru.

### Console

Untuk mendaftarkan fungsi Lambda

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Target, jika tidak ada fungsi Lambda yang terdaftar, pilih Daftarkan target.
5. Pilih fungsi Lambda atau masukkan ARN-nya.
6. Pilih Pendaftaran.

### AWS CLI

Untuk mendaftarkan fungsi Lambda

Gunakan perintah [register-target](#).

```
aws elbv2 register-targets \
  --target-group-arn target-group-arn \
  --targets Id=lambda-function-arn
```

### CloudFormation

Untuk mendaftarkan fungsi Lambda

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya.

```
Resources:
  myTargetGroup:
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
```

```
Properties:
  Name: my-target-group
  TargetType: lambda
  Tags:
    - Key: 'department'
      Value: '123'
  Targets:
    - Id: !Ref myLambdaFunction
```

## Deregistrasi fungsi Lambda

Jika Anda tidak perlu lagi mengirim lalu lintas ke fungsi Lambda Anda, Anda dapat membatalkan pendaftarannya. Setelah Anda membatalkan pendaftaran fungsi Lambda, permintaan dalam penerbangan gagal dengan galat HTTP 5XX.

Untuk mengganti fungsi Lambda, kami sarankan Anda membuat grup target baru, mendaftarkan fungsi baru dengan grup target baru, dan memperbarui aturan listener untuk menggunakan grup target baru.

### Console

Untuk membatalkan pendaftaran fungsi Lambda

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan laman detailnya.
4. Pada tab Target, pilih target dan pilih Deregister.
5. Ketika konfirmasi diminta, pilih Batalkan Pendaftaran.

### AWS CLI

Untuk membatalkan pendaftaran fungsi Lambda

Gunakan perintah [Target deregister](#).

```
aws elbv2 deregister-targets \  
  --target-group-arn target-group-arn \  
  --targets Id=lambda-function-arn
```

## Tag untuk kelompok target Application Load Balancer Anda

Tag membantu Anda mengategorikan grup target Auto dengan berbagai cara, misalnya, berdasarkan tujuan, pemilik, atau lingkungan.

Anda dapat menambahkan beberapa tag ke setiap grup Auto Scaling. Tombol tag harus unik untuk setiap kelompok target. Jika Anda menambahkan tag dengan kunci yang sudah terkait dengan grup target, maka akan memperbarui nilai tag tersebut.

Setelah selesai dengan tag, Anda dapat menghapusnya.

### Pembatasan

- Jumlah maksimum tanda per sumber daya—50
- Panjang kunci maksimum – 127 karakter Unicode
- Panjang nilai maksimum – 255 karakter Unicode
- Kunci dan nilai tanda peka huruf besar dan kecil. Karakter yang diperbolehkan adalah: huruf, spasi, dan angka yang dapat mewakili dalam UTF-8, serta karakter berikut: + - = . \_ : / @. \_:/@. Jangan gunakan spasi terkemuka atau paling belakang.
- Jangan gunakan `aws :` awalan dalam nama atau nilai tag Anda karena itu dicadangkan untuk AWS digunakan. Anda tidak dapat mengedit atau menghapus nama atau nilai tag dengan awalan ini. Tag dengan awalan ini tidak dihitung terhadap tag Anda per batas sumber daya.

### Console

Untuk mengelola tag untuk grup target

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. Pilih nama grup opsi untuk menampilkan halaman detailnya.
4. Pada tab Tag, pilih Kelola tag dan lakukan satu atau beberapa hal berikut:
  - a. Untuk memperbarui tag, masukkan nilai baru untuk Kunci dan Nilai.
  - b. Untuk menambahkan tag, pilih Tambahkan Tag dan masukkan nilai untuk Kunci dan Nilai
  - c. Untuk menghapus sebuah tag, pilih Remove di samping tag yang akan dihapus.

## 5. Pilih Simpan perubahan.

### AWS CLI

Untuk menambahkan tag

Gunakan perintah [add-tag](#). Contoh berikut menambahkan dua tag.

```
aws elbv2 add-tags \  
  --resource-arns target-group-arn \  
  --tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

Untuk menghapus tag

Gunakan perintah [remove-tag](#). Contoh berikut menghapus tag dengan kunci yang ditentukan.

```
aws elbv2 remove-tags \  
  --resource-arns target-group-arn \  
  --tag-keys project department
```

### CloudFormation

Untuk menambahkan tag

Perbarui [AWS::ElasticLoadBalancingV2::TargetGroup](#) sumber daya untuk menyertakan Tags properti.

```
Resources:  
  myTargetGroup:  
    Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'  
    Properties:  
      Name: my-target-group  
      Protocol: HTTP  
      Port: 80  
      TargetType: ip  
      VpcId: !Ref myVPC  
      Tags:  
        - Key: 'project'  
          Value: 'lima'  
        - Key: 'department'  
          Value: 'digital-media'
```

# Menghapus grup target Application Load Balancer

Anda dapat menghapus kelompok target jika tidak direferensikan oleh tindakan lanjut dari aturan pendengar. Menghapus kelompok target tidak mempengaruhi target terdaftar dengan kelompok target. Jika Anda tidak lagi membutuhkan instance EC2 terdaftar, Anda dapat menghentikan atau menghapusnya.

## Console

Untuk menghapus grup target

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Pada panel navigasi, di bawah Penyeimbang Beban, pilih Grup Target.
3. Pilih grup target dan pilih Tindakan, Hapus.
4. Pilih Hapus.

## AWS CLI

Untuk menghapus grup target

Gunakan perintah [delete-target-group](#).

```
aws elbv2 delete-target-group \  
  --target-group-arn target-group-arn
```

# Memantau Application Load Balancer Anda

Anda dapat menggunakan fitur berikut untuk memantau penyeimbang beban, menganalisis pola lalu lintas, dan memecahkan masalah dengan penyeimbang beban dan target Anda.

## CloudWatch metrik

Anda dapat menggunakan Amazon CloudWatch untuk mengambil statistik tentang titik data untuk penyeimbang beban dan target sebagai kumpulan data deret waktu yang diurutkan, yang dikenal sebagai metrik. Anda dapat menggunakan metrik ini untuk memverifikasi bahwa sistem Anda bekerja sesuai harapan. Untuk informasi selengkapnya, lihat [CloudWatch metrik untuk Application Load Balancer](#).

## Log akses

Anda dapat menggunakan log akses untuk mengambil informasi mendetail tentang permintaan yang dibuat ke penyeimbang beban Anda dan menyimpannya sebagai berkas log di Amazon S3. Anda dapat menggunakan log akses ini untuk menganalisis pola lalu lintas dan memecahkan masalah dengan target Anda. Untuk informasi selengkapnya, lihat [Log akses untuk Application Load Balancer Anda](#).

## Log koneksi

Anda dapat menggunakan log koneksi untuk menangkap atribut tentang permintaan yang dikirim ke penyeimbang beban, dan menyimpannya sebagai file log di Amazon S3. Anda dapat menggunakan log koneksi ini untuk menentukan alamat IP klien dan port, informasi sertifikat klien, hasil koneksi, dan cipher TLS yang digunakan. Log koneksi ini kemudian dapat digunakan untuk meninjau pola permintaan, dan tren lainnya. Untuk informasi selengkapnya, lihat [Log koneksi untuk Application Load Balancer](#).

## Log pemeriksaan kesehatan

Anda dapat menggunakan log pemeriksaan kesehatan untuk menangkap informasi terperinci tentang pemeriksaan kesehatan yang dilakukan ke target terdaftar untuk penyeimbang beban Anda dan menyimpannya sebagai file log di Amazon S3. Anda dapat menggunakan log pemeriksaan kesehatan ini untuk memecahkan masalah dengan target Anda. Untuk informasi selengkapnya, lihat [Log pemeriksaan kesehatan](#).

## Pelacakan permintaan

Anda dapat menggunakan pelacakan permintaan untuk melacak permintaan HTTP. Penyeimbang beban menambahkan header dengan pengidentifikasi jejak untuk setiap permintaan yang

diterimanya. Untuk informasi selengkapnya, lihat [Pelacakan permintaan untuk Application Load Balancer Anda](#).

### CloudTrail log

Anda dapat menggunakan AWS CloudTrail untuk menangkap informasi terperinci tentang panggilan yang dilakukan ke Elastic Load Balancing API dan menyimpannya sebagai file log di Amazon S3. Anda dapat menggunakan CloudTrail log ini untuk menentukan panggilan mana yang dilakukan, alamat IP sumber dari mana panggilan itu berasal, siapa yang melakukan panggilan, kapan panggilan dilakukan, dan sebagainya. Untuk informasi selengkapnya, lihat [Log panggilan API untuk Elastic Load Balancing menggunakan](#) CloudTrail

## CloudWatch metrik untuk Application Load Balancer

Elastic Load Balancing menerbitkan titik data ke Amazon CloudWatch untuk penyeimbang beban dan target Anda. CloudWatch memungkinkan Anda untuk mengambil statistik tentang titik-titik data tersebut sebagai kumpulan data deret waktu yang diurutkan, yang dikenal sebagai metrik. Anggap metrik sebagai variabel untuk memantau dan titik data sebagai nilai variabel tersebut dari waktu ke waktu. Misalnya, Anda dapat memantau jumlah total target sehat untuk penyeimbang beban selama periode waktu tertentu. Setiap titik data memiliki stempel waktu terkait dan unit pengukuran opsional.

Anda dapat menggunakan metrik untuk memverifikasi bahwa sistem Anda bekerja sesuai harapan. Misalnya, Anda dapat membuat CloudWatch alarm untuk memantau metrik tertentu dan memulai tindakan (seperti mengirim pemberitahuan ke alamat email) jika metrik berada di luar rentang yang Anda anggap dapat diterima.

Elastic Load Balancing melaporkan metrik CloudWatch hanya ketika permintaan mengalir melalui penyeimbang beban. Jika ada permintaan yang mengalir melalui penyeimbang beban, Elastic Load Balancing mengukur dan mengirimkan metriknya dalam interval 60 detik. Jika tidak ada permintaan yang mengalir melalui penyeimbang beban atau tidak ada data untuk metrik, metrik tidak dilaporkan.

Metrik untuk Application Load Balancer mengecualikan permintaan pemeriksaan kesehatan.

Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).

### Daftar Isi

- [Metrik Application Load Balancer](#)
- [Dimensi metrik untuk Application Load Balancer](#)
- [Statistik untuk metrik Application Load Balancer](#)

- [Lihat CloudWatch metrik untuk penyeimbang beban](#)

## Metrik Application Load Balancer

- [Penyeimbang beban](#)
- [LCUs](#)
- [Target](#)
- [Kesehatan kelompok sasaran](#)
- [Fungsi Lambda](#)
- [Otentikasi pengguna](#)
- [Pengoptimal Target](#)

Namespace AWS/ApplicationELB menyertakan metrik berikut untuk penyeimbang beban.

Metrik	Deskripsi
ActiveConnectionCount	<p>Jumlah total koneksi TCP bersamaan yang aktif dari klien ke penyeimbang beban dan dari penyeimbang beban ke target.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>
BYoIPUtilPercentage	<p>Persentase penggunaan dari kolam IP.</p> <p>Kriteria pelaporan: BYo IP diaktifkan pada penyeimbang beban.</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Average.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer , TargetGroup</li> </ul>

Metrik	Deskripsi
ClientTLSNegotiationErrorCount	<ul style="list-style-type: none"> <li>• LoadBalancer , TargetGroup , AvailabilityZone</li> </ul> <p>Jumlah koneksi TLS yang dimulai oleh klien yang tidak membuat sesi dengan penyeimbang beban karena kesalahan TLS. Kemungkinan penyebabnya termasuk ketidakcocokan cipher atau protokol atau klien gagal memverifikasi sertifikat server dan menutup koneksi.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>
DesyncMitigationMode_NonCompliant_Request_Count	<p>Jumlah permintaan yang tidak sesuai dengan RFC 7230.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>

Metrik	Deskripsi
<b>DroppedInvalidHeaderRequestCount</b>	<p>Jumlah permintaan di mana penyeimbang beban menghapus header HTTP dengan bidang header yang tidak valid sebelum perutean permintaan. Penyeimbang beban menghapus header ini hanya jika atribut <code>routing.http.drop_invalid_header_fields.enabled</code> diatur ke <code>true</code>.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Semua</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• <code>AvailabilityZone</code> , <code>LoadBalancer</code></li></ul>
<b>ForwardedInvalidHeaderRequestCount</b>	<p>Jumlah permintaan yang dirutekan oleh penyeimbang beban yang memiliki header HTTP dengan bidang header yang tidak valid. Penyeimbang beban meneruskan permintaan dengan header ini hanya jika atribut <code>routing.http.drop_invalid_header_fields.enabled</code> diatur ke <code>false</code>.</p> <p>Reporting criteria: Selalu dilaporkan</p> <p>Statistics: Semua</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• <code>AvailabilityZone</code> , <code>LoadBalancer</code></li></ul>

Metrik	Deskripsi
GprcRequestCount	<p>Jumlah permintaan gRPC diproses secara berulang-ulang IPv4 dan IPv6</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum. Minimum, Maximum, dan Average semua kembali 1.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer , TargetGroup</li> <li>• AvailabilityZone , LoadBalancer , TargetGroup</li> <li>• TargetGroup</li> <li>• AvailabilityZone , TargetGroup</li> </ul>
HTTP_Fixed_Response_Count	<p>Jumlah tindakan respons tetap yang berhasil.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>
HTTP_Redirect_Count	<p>Jumlah tindakan pengalihan yang berhasil.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>

Metrik	Deskripsi
HTTP_Redirect_Url_Limit_Exceeded_Count	<p>Jumlah tindakan pengalihan yang tidak dapat diselesaikan karena URL di header lokasi respons lebih besar dari 8K.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>
HTTPCode_ELB_3XX_Count	<p>Jumlah kode pengalihan HTTP 3XX yang berasal dari penyeimbang beban. Jumlah ini tidak termasuk kode respons yang dihasilkan oleh target.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>

Metrik	Deskripsi
HTTPCode_ELB_4XX_Count	<p>Jumlah kode kesalahan klien HTTP 4XX yang berasal dari penyeimbang beban. Jumlah ini tidak termasuk kode respons yang dihasilkan oleh target.</p> <p>Kesalahan klien dihasilkan saat permintaan salah format atau tidak lengkap. Permintaan ini tidak diterima oleh target, selain jika penyeimbang beban mengembalikan <a href="#">kode kesalahan HTTP 460</a>. Jumlah ini tidak termasuk kode respons apa pun yang dihasilkan oleh target.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum, Minimum, Maximum, dan Average semua kembali 1.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>
HTTPCode_ELB_5XX_Count	<p>Jumlah kode kesalahan server HTTP 5XX yang berasal dari penyeimbang beban. Jumlah ini tidak termasuk kode respons apa pun yang dihasilkan oleh target.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum, Minimum, Maximum, dan Average semua kembali 1.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>

Metrik	Deskripsi
HTTPCode_ELB_500_Count	<p>Jumlah kode kesalahan HTTP 500 yang berasal dari penyeimbang beban.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>
HTTPCode_ELB_502_Count	<p>Jumlah kode kesalahan HTTP 502 yang berasal dari penyeimbang beban.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>
HTTPCode_ELB_503_Count	<p>Jumlah kode kesalahan HTTP 503 yang berasal dari penyeimbang beban.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>

Metrik	Deskripsi
HTTPCode_ELB_504_Count	<p>Jumlah kode kesalahan HTTP 504 yang berasal dari penyeimbang beban.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>
IPv6ProcessedBytes	<p>Jumlah total byte yang diproses oleh load balancer over. IPv6 Hitungan ini termasuk dalam ProcessedBytes .</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>
IPv6RequestCount	<p>Jumlah IPv6 permintaan yang diterima oleh penyeimbang beban.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum. Minimum, Maximum, dan Average semua kembali 1.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>

Metrik	Deskripsi
LowReputationPacketsDropped	<p>Jumlah paket turun dari sumber berbahaya yang diketahui. Metrik ini dicatat ketika permintaan diblokir oleh perlindungan S tingkat sumber daya DDo.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>
LowReputationRequestsDenied	<p>Jumlah permintaan HTTP ditolak dengan respons HTTP 403. Metrik ini dicatat ketika permintaan diblokir oleh perlindungan S tingkat sumber daya DDo.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>
NewConnectionCount	<p>Jumlah total koneksi TCP baru yang dibuat dari klien ke penyeimbang beban dan dari penyeimbang beban ke target.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>

Metrik	Deskripsi
NonStickyRequestCount	<p>Jumlah permintaan di mana penyeimbang beban memilih target baru karena tidak dapat menggunakan sesi lekat yang ada. Misalnya, permintaan adalah permintaan pertama dari klien baru dan tidak ada cookie lekat yang disajikan, cookie lekat disajikan tetapi tidak menentukan target yang terdaftar dengan grup target ini, cookie lekat salah format atau kedaluwarsa, atau kesalahan internal mencegah penyeimbang beban membaca cookie lekat.</p> <p>Reporting criteria: Kelekatan diaktifkan pada grup target.</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>
ProcessedBytes	<p>Jumlah total byte yang diproses oleh penyeimbang beban over IPv4 dan IPv6 (header HTTP dan payload HTTP). Jumlah ini mencakup lalu lintas ke dan dari klien dan fungsi Lambda, lalu lintas melalui koneksi Websocket, dan lalu lintas dari Penyedia Identitas (iDP) jika otentikasi pengguna diaktifkan.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• LoadBalancer</li><li>• AvailabilityZone , LoadBalancer</li></ul>

Metrik	Deskripsi
RejectedConnectionCount	<p>Jumlah koneksi yang ditolak karena penyeimbang beban telah mencapai jumlah koneksi maksimumnya.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>
RequestCount	<p>Jumlah permintaan yang diproses berulang-ulang IPv4 dan IPv6. Metrik ini hanya bertambah untuk permintaan di mana simpul penyeimbang beban dapat memilih target. Permintaan yang ditolak sebelum target dipilih tidak tercermin dalam metrik ini.</p> <p>Kriteria pelaporan: Dilaporkan jika ada target terdaftar.</p> <p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• LoadBalancer , AvailabilityZone</li> <li>• LoadBalancer , TargetGroup</li> <li>• LoadBalancer , AvailabilityZone , TargetGroup</li> </ul>

Metrik	Deskripsi
RuleEvaluations	<p>Jumlah aturan yang dievaluasi oleh penyeimbang beban saat memproses permintaan. Aturan default tidak dihitung. 10 evaluasi aturan gratis per permintaan termasuk dalam hitungan ini.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> </ul>

AWS/ApplicationELBNamespace mencakup metrik berikut untuk unit kapasitas penyeimbang beban (LCU).

Metrik	Deskripsi
ConsumedLCUs	<p>Jumlah unit kapasitas penyeimbang beban (LCU) yang digunakan oleh penyeimbang beban Anda. Anda membayar untuk jumlah LCUs yang Anda gunakan per jam. Ketika reservasi LCU aktif, LCUs Consumed akan melaporkan 0 jika penggunaan di bawah kapasitas cadangan, dan akan melaporkan nilai di atas 0 jika penggunaan melebihi yang dipesan LCUs. Untuk informasi selengkapnya, lihat <a href="#">Harga Elastic Load Balancing</a>.</p> <p>Reporting criteria: Selalu dilaporkan</p> <p>Statistics: Semua</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> </ul>
PeakLCUs	<p>Jumlah maksimum unit kapasitas penyeimbang beban (LCU) yang digunakan oleh penyeimbang beban Anda pada titik waktu tertentu. Hanya berlaku saat menggunakan Reservasi LCU.</p>

Metrik	Deskripsi
	<p>Kriteria pelaporan: Selalu</p> <p>Statistik: Statistik yang paling berguna adalah Sum dan Max.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> </ul>
ReservedLCUs	<p>Metrik penagihan yang melaporkan kapasitas cadangan per menit. Total Reserved LCUs selama periode apa pun adalah jumlah yang akan dikenakan biaya untuk LCUs Anda. Misalnya, jika 500 LCUs dicadangkan selama satu jam, metrik per menit akan menjadi 8,33. LCUs Untuk informasi selengkapnya, lihat <a href="#">Pantau reservasi</a>.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Semua</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> </ul>

Namespace AWS/ApplicationELB menyertakan metrik berikut untuk target.

Metrik	Deskripsi
AnomalousHostCount	<p>Jumlah host yang terdeteksi dengan anomali.</p> <p>Reporting criteria: Selalu dilaporkan</p> <p>Statistik: Satu-satunya statistik yang berarti adalah Minimum dan Maximum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• TargetGroup , LoadBalancer</li> <li>• TargetGroup , AvailabilityZone , LoadBalancer</li> </ul>

Metrik	Deskripsi
HealthyHostCount	<p>Jumlah target yang dianggap sehat.</p> <p>Kriteria pelaporan: Dilaporkan jika ada target terdaftar.</p> <p>Statistics: Statistik yang paling berguna adalah Average, Minimum, dan Maximum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer , TargetGroup</li> <li>• LoadBalancer , AvailabilityZone , TargetGroup</li> </ul>
HTTPCode_Target_2XX_Count , HTTPCode_Target_3XX_Count , HTTPCode_Target_4XX_Count , HTTPCode_Target_5XX_Count	<p>Jumlah kode respons HTTP yang dihasilkan oleh target. Jumlah ini tidak termasuk kode respons apa pun yang dihasilkan oleh penyeimbang beban.</p> <p>Kriteria pelaporan: Dilaporkan jika ada target terdaftar.</p> <p>Statistics: Statistik yang paling berguna adalah Sum, Minimum, Maximum, dan Average semua kembali 1.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> <li>• TargetGroup , LoadBalancer</li> <li>• TargetGroup , AvailabilityZone , LoadBalancer</li> </ul>

Metrik	Deskripsi
MitigatedHostCount	<p>Jumlah target di bawah mitigasi.</p> <p>Reporting criteria: Selalu dilaporkan</p> <p>Statistics: Statistik yang paling berguna adalah Average, Minimum, dan Maximum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• TargetGroup , LoadBalancer</li> <li>• TargetGroup , AvailabilityZone , LoadBalancer</li> </ul>
RequestCountPerTarget	<p>Jumlah permintaan rata-rata per target, dalam kelompok target. Anda harus menentukan grup target menggunakan dimensi TargetGroup . Metrik ini tidak berlaku jika targetnya adalah fungsi Lambda.</p> <p>Hitungan ini menggunakan jumlah total permintaan yang diterima oleh kelompok sasaran, dibagi dengan jumlah target sehat dalam kelompok sasaran. Jika tidak ada target sehat dalam kelompok sasaran, itu dibagi dengan jumlah total target yang terdaftar.</p> <p>Reporting criteria: Selalu dilaporkan</p> <p>Statistics: Satu-satunya statistik yang valid adalah Sum. Statistik ini mewakili rata-rata bukan jumlah.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• TargetGroup</li> <li>• TargetGroup , AvailabilityZone</li> <li>• LoadBalancer , TargetGroup</li> <li>• LoadBalancer , AvailabilityZone , TargetGroup</li> </ul>

Metrik	Deskripsi
TargetConnectionErrorCount	<p>Jumlah koneksi yang tidak berhasil dibuat antara penyeimbang beban dan target. Metrik ini tidak berlaku jika targetnya adalah fungsi Lambda. Metrik ini tidak bertambah untuk koneksi pemeriksaan kesehatan yang gagal.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> <li>• TargetGroup , LoadBalancer</li> <li>• TargetGroup , AvailabilityZone , LoadBalancer</li> </ul>
TargetResponseTime	<p>Waktu berlalu, dalam hitungan detik, setelah permintaan meninggalkan penyeimbang beban hingga target mulai mengirim header respons. Ini setara dengan bidang <code>target_processing_time</code> di log akses.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Average dan pNN.NN (persentil).</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> <li>• TargetGroup , LoadBalancer</li> <li>• TargetGroup , AvailabilityZone , LoadBalancer</li> </ul>

Metrik	Deskripsi
TargetTLSNegotiationErrorCount	<p>Jumlah koneksi TLS yang dimulai oleh penyeimbang beban yang tidak membuat sesi dengan target. Kemungkinan penyebabnya termasuk ketidakcocokan cipher atau protokol. Metrik ini tidak berlaku jika targetnya adalah fungsi Lambda.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> <li>• TargetGroup , LoadBalancer</li> <li>• TargetGroup , AvailabilityZone , LoadBalancer</li> </ul>
UnHealthyHostCount	<p>Jumlah target yang dianggap tidak sehat.</p> <p>Ketika Anda membatalkan pendaftaran target, ini berkurang HealthyHostCount tetapi tidak meningkat. Unhealthy HostCount</p> <p>Kriteria pelaporan: Dilaporkan jika ada target terdaftar.</p> <p>Statistics: Statistik yang paling berguna adalah Average, Minimum, dan Maximum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer , TargetGroup</li> <li>• LoadBalancer , AvailabilityZone , TargetGroup</li> </ul>

Metrik	Deskripsi
ZonalShiftedHostCount	<p>Jumlah target yang dianggap dinonaktifkan karena pergeseran zona.</p> <p>Kriteria pelaporan: Dilaporkan ketika ada nilai</p> <p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer , TargetGroup .</li> <li>• AvailabilityZone , LoadBalancer , TargetGroup .</li> </ul>

AWS/ApplicationELBNamespace menyertakan metrik berikut untuk kesehatan grup target. Untuk informasi selengkapnya, lihat [the section called “Kesehatan kelompok sasaran”](#).

Metrik	Deskripsi
HealthyStateDNS	<p>Jumlah zona yang memenuhi persyaratan status sehat DNS.</p> <p>Statistics: Statistik yang paling berguna adalah Max.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer , TargetGroup</li> <li>• AvailabilityZone , LoadBalancer , TargetGroup</li> </ul>
HealthyStateRouting	<p>Jumlah zona yang memenuhi persyaratan keadaan sehat perutean.</p> <p>Statistics: Statistik yang paling berguna adalah Max.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer , TargetGroup</li> <li>• AvailabilityZone , LoadBalancer , TargetGroup</li> </ul>
UnhealthyRoutingRequestCount	<p>Jumlah permintaan yang dirutekan menggunakan tindakan failover routing (gagal terbuka).</p>

Metrik	Deskripsi
	<p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer , TargetGroup</li> <li>• AvailabilityZone , LoadBalancer , TargetGroup</li> </ul>
UnhealthyStateDNS	<p>Jumlah zona yang tidak memenuhi persyaratan keadaan sehat DNS dan karenanya ditandai tidak sehat di DNS.</p> <p>Statistics: Statistik yang paling berguna adalah Min.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer , TargetGroup</li> <li>• AvailabilityZone , LoadBalancer , TargetGroup</li> </ul>
UnhealthyStateRouting	<p>Jumlah zona yang tidak memenuhi persyaratan perutean kondisi sehat, dan oleh karena itu penyeimbang beban mendistribusikan lalu lintas ke semua target di zona tersebut, termasuk target yang tidak sehat.</p> <p>Statistics: Statistik yang paling berguna adalah Min.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer , TargetGroup</li> <li>• AvailabilityZone , LoadBalancer , TargetGroup</li> </ul>

Namespace AWS/ApplicationELB menyertakan metrik berikut untuk fungsi Lambda yang terdaftar sebagai target.

Metrik	Deskripsi
LambdaInternalError	Jumlah permintaan untuk fungsi Lambda yang gagal karena masalah internal pada penyeimbang beban atau AWS Lambda. Untuk

Metrik	Deskripsi
	<p>mendapatkan kode alasan kesalahan, periksa bidang <code>error_reason</code> dari log akses.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• TargetGroup</li> <li>• TargetGroup , LoadBalancer</li> </ul>
<p>LambdaTargetProcessedBytes</p>	<p>Jumlah total byte yang diproses oleh penyeimbang beban untuk permintaan ke dan respons dari fungsi Lambda.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> </ul>
<p>LambdaUserError</p>	<p>Jumlah permintaan untuk fungsi Lambda yang gagal karena masalah dengan fungsi Lambda. Misalnya penyeimbang beban tidak memiliki izin untuk mengaktifkan fungsi, penyeimbang beban menerima JSON dari fungsi yang salah format atau kehilangan bidang yang wajib diisi, atau ukuran isi permintaan atau respons melebihi ukuran maksimum 1 MB. Untuk mendapatkan kode alasan kesalahan, periksa bidang <code>error_reason</code> dari log akses.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• TargetGroup</li> <li>• TargetGroup , LoadBalancer</li> </ul>

Namespace `AWS/ApplicationELB` menyertakan metrik berikut untuk autentikasi pengguna.

Metrik	Deskripsi
<code>ELBAuthError</code>	<p>Jumlah autentikasi pengguna yang tidak dapat diselesaikan karena tindakan autentikasi salah dikonfigurasi, penyeimbang beban tidak dapat membuat koneksi dengan IdP, atau penyeimbang beban tidak dapat menyelesaikan alur autentikasi karena kesalahan internal. Untuk mendapatkan kode alasan kesalahan, periksa bidang <code>error_reason</code> dari log akses.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• <code>LoadBalancer</code></li> <li>• <code>AvailabilityZone</code> , <code>LoadBalancer</code></li> </ul>
<code>ELBAuthFailure</code>	<p>Jumlah autentikasi pengguna yang tidak dapat diselesaikan karena IdP menolak akses ke pengguna atau kode otorisasi digunakan lebih dari sekali. Untuk mendapatkan kode alasan kesalahan, periksa bidang <code>error_reason</code> dari log akses.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• <code>LoadBalancer</code></li> <li>• <code>AvailabilityZone</code> , <code>LoadBalancer</code></li> </ul>
<code>ELBAuthLatency</code>	<p>Waktu berlalu dalam hitungan milidetik untuk membuat kueri IdP untuk token ID dan info pengguna. Jika satu atau beberapa operasi ini gagal, inilah saatnya untuk gagal.</p> <p>Reporting criteria: Ada nilai bukan nol</p>

Metrik	Deskripsi
	<p>Statistics: Semua statistik bermakna.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>
ELBAuthRefreshTokenSuccess	<p>Frekuensi penyeimbang beban berhasil merefresh klaim pengguna menggunakan token refresh yang diberikan oleh IdP.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>
ELBAuthSuccess	<p>Jumlah tindakan autentikasi yang berhasil. Metrik ini bertambah di akhir alur kerja autentikasi setelah penyeimbang beban mengambil klaim pengguna dari IdP.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Statistik yang paling berguna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>

Metrik	Deskripsi
ELBAuthUserClaimsSizeExceeded	<p>Frekuensi IdP yang dikonfigurasi mengembalikan klaim pengguna yang ukurannya melebihi 11K byte.</p> <p>Reporting criteria: Ada nilai bukan nol</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>

AWS/ApplicationELBNamespace menyertakan metrik berikut untuk pengoptimal target.

Metrik	Deskripsi
TargetControlRequestCount	<p>Jumlah permintaan yang diteruskan oleh ALB ke agen.</p> <p>Kriteria pelaporan: Pengoptimal target diaktifkan pada grup target dan ada nilai bukan nol.</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>
TargetControlRequestRejectCount	<p>Jumlah permintaan yang ditolak oleh ALB karena tidak ada target yang siap menerima permintaan. Metrik ini menunjukkan uptick saat TargetControlWorkQueueLength nol.</p> <p>Kriteria pelaporan: Pengoptimal target diaktifkan pada grup target dan ada nilai bukan nol.</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p>

Metrik	Deskripsi
	<p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>
<p>TargetControlActiveChannelCount</p>	<p>Jumlah saluran kontrol aktif antara ALB dan agen. Untuk penyeimbangan beban, ini harus sama dengan jumlah agen. Angka yang lebih rendah dari yang diharapkan menunjukkan bahwa agen tidak dikonfigurasi dengan benar atau tidak tersedia.</p> <p>Kriteria pelaporan: Pengoptimal target diaktifkan pada grup target dan ada nilai bukan nol.</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>
<p>TargetControlNewChannelCount</p>	<p>Jumlah saluran kontrol baru yang dibuat antara ALB dan agen. Anda akan melihat peningkatan dalam metrik ini ketika target baru dengan agen yang diinstal berhasil ditambahkan ke grup target.</p> <p>Kriteria pelaporan: Pengoptimal target diaktifkan pada grup target dan ada nilai bukan nol.</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>

Metrik	Deskripsi
<code>TargetControlChannelErrorCount</code>	<p>Jumlah saluran kontrol antara ALB dan agen yang gagal membuat atau mengalami kesalahan yang tidak terduga. Kesalahan saluran kontrol akan mengakibatkan agen (dan target) tidak menerima lalu lintas aplikasi apa pun.</p> <p>Kriteria pelaporan: Pengoptimal target diaktifkan pada grup target dan ada nilai bukan nol.</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• <code>LoadBalancer</code></li><li>• <code>AvailabilityZone</code> , <code>LoadBalancer</code></li></ul>
<code>TargetControlWorkQueueLength</code>	<p>Jumlah sinyal yang diterima oleh ALB dari agen yang meminta permintaan.</p> <p>Data ini berasal dari snapshot yang diambil pada interval 1 menit. Perubahan sub-menit tidak ditangkap.</p> <p>Kriteria pelaporan: Pengoptimal target diaktifkan pada grup target dan ada nilai bukan nol.</p> <p>Statistics: Satu-satunya statistik yang bermakna adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"><li>• <code>LoadBalancer</code></li><li>• <code>AvailabilityZone</code> , <code>LoadBalancer</code></li></ul>

Metrik	Deskripsi
TargetControlProcessedBytes	<p>Jumlah byte yang diproses oleh ALB untuk lalu lintas ke grup target yang mengaktifkan pengoptimal target.</p> <p>Kriteria pelaporan: Pengoptimal target diaktifkan pada grup target dan ada nilai bukan nol.</p> <p>Statistik: Statistik yang paling berarti adalah Sum.</p> <p>Dimensi</p> <ul style="list-style-type: none"> <li>• LoadBalancer</li> <li>• AvailabilityZone , LoadBalancer</li> </ul>

## Dimensi metrik untuk Application Load Balancer

Untuk memfilter metrik untuk Application Load Balancer Anda, gunakan dimensi berikut.

Dimensi	Deskripsi
AvailabilityZone	Memfilter data metrik berdasarkan Availability Zone.
LoadBalancer	Memfilter data metrik berdasarkan penyeimbang beban. Tentukan penyeimbang beban sebagai berikut: app/ load-balancer-name /1234567890123456 (bagian akhir dari load balancer ARN).
TargetGroup	Memfilter data metrik berdasarkan grup target. Tentukan kelompok target sebagai berikut: targetgroup/ target-group-name/1234567890123456 (bagian akhir dari kelompok target ARN).

## Statistik untuk metrik Application Load Balancer

CloudWatch menyediakan statistik berdasarkan titik data metrik yang diterbitkan oleh Elastic Load Balancing. Statistik adalah agregasi data metrik selama periode waktu tertentu. Saat Anda meminta statistik, aliran data yang dikembalikan diidentifikasi oleh nama metrik dan dimensi. Dimensi adalah

pasangan nama-nilai yang secara unik mengidentifikasi metrik. Misalnya, Anda dapat meminta statistik untuk semua instans EC2 yang sehat di belakang penyeimbang beban yang diluncurkan di Availability Zone tertentu.

Statistik `Minimum` dan `Maximum` mencerminkan nilai minimum dan maksimum titik data yang dilaporkan oleh simpul penyeimbang beban oleh individu di setiap jendela pengambilan sampel. Misalnya, ada 2 node load balancer yang membentuk Application Load Balancer. Satu simpul memiliki `HealthyHostCount` dengan `Minimum` 2, `Maximum` 10, dan `Average` 6, sedangkan simpul lainnya memiliki `HealthyHostCount` dengan `Minimum` 1, `Maximum` 5, dan `Average` 3. Oleh karena itu, penyeimbang beban memiliki `Minimum` 1, `Maximum` 10, dan `Average` sekitar 4.

Kami menyarankan Anda memantau bukan nol `UnHealthyHostCount` dalam `Minimum` statistik, dan alarm pada nilai bukan nol untuk lebih dari satu titik data. Menggunakan `Minimum` will mendeteksi kapan target dianggap tidak sehat oleh setiap node dan Availability Zone dari load balancer Anda. Mengkhawatirkan `Average` atau `Maximum` berguna jika Anda ingin diberitahu tentang potensi masalah, dan kami menyarankan pelanggan meninjau metrik ini dan menyelidiki kejadian bukan nol. Mengurangi kegagalan secara otomatis dapat dilakukan dengan mengikuti praktik terbaik menggunakan pemeriksaan kesehatan penyeimbang beban di Amazon EC2 Auto Scaling, atau Amazon Elastic Container Service (Amazon ECS).

Statistik `Sum` adalah nilai agregat di semua simpul penyeimbang beban. Karena metrik menyertakan beberapa laporan per periode, `Sum` hanya berlaku untuk metrik yang diagregasikan di semua simpul penyeimbang beban.

Statistik `SampleCount` adalah jumlah sampel yang diukur. Karena metrik dikumpulkan berdasarkan interval dan peristiwa pengambilan sampel, statistik ini biasanya tidak berguna. Misalnya dengan `HealthyHostCount`, `SampleCount` didasarkan pada jumlah sampel yang dilaporkan setiap simpul penyeimbang beban, bukan jumlah host yang sehat.

Persentil menunjukkan posisi relatif suatu nilai dalam set data. Anda dapat menentukan persentil apa pun, menggunakan hingga dua tempat desimal (misalnya, hal 95.45). Misalnya, persentil ke-95 berarti bahwa 95 persen data berada di bawah nilai ini dan 5 persen di atas. Persentil sering kali digunakan untuk mengisolasi anomali. Misalnya, anggaplah aplikasi melayani sebagian besar permintaan dari cache dalam 1-2 ms, tetapi dalam 100-200 ms jika cache kosong. Maksimumnya mencerminkan kasus paling lambat, sekitar 200 ms. Rata-ratanya tidak menunjukkan distribusi data. Persentil memberikan tampilan performa aplikasi yang lebih bermakna. Dengan menggunakan persentil ke-99 sebagai pemicu Auto Scaling atau CloudWatch alarm, Anda dapat menargetkan bahwa tidak lebih dari 1 persen permintaan membutuhkan waktu lebih dari 2 ms untuk diproses.

## Lihat CloudWatch metrik untuk penyeimbang beban

Anda dapat melihat CloudWatch metrik untuk penyeimbang beban menggunakan konsol Amazon EC2. Metrik ini ditampilkan sebagai grafik pemantauan. Grafik pemantauan menunjukkan titik data jika penyeimbang beban aktif dan menerima permintaan.

Atau, Anda dapat melihat metrik untuk penyeimbang beban menggunakan konsol CloudWatch

Untuk melihat metrik menggunakan konsol

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Untuk melihat metrik yang difilter oleh grup target, lakukan hal berikut:
  - a. Di panel navigasi, pilih Target Groups.
  - b. Pilih grup target Anda, lalu pilih tab Monitoring.
  - c. (Opsional) Untuk memfilter hasil berdasarkan waktu, pilih rentang waktu dari Showing data for.
  - d. Untuk mendapatkan tampilan yang lebih besar dari satu metrik, pilih grafiknya.
3. Untuk melihat metrik yang difilter oleh penyeimbang beban, lakukan hal berikut:
  - a. Di panel navigasi, pilih Load Balancers.
  - b. Pilih penyeimbang beban Anda, lalu pilih tab Monitoring.
  - c. (Opsional) Untuk memfilter hasil berdasarkan waktu, pilih rentang waktu dari Showing data for.
  - d. Untuk mendapatkan tampilan yang lebih besar dari satu metrik, pilih grafiknya.

Untuk melihat metrik menggunakan konsol CloudWatch

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Di panel navigasi, pilih Metrik.
3. Pilih namespace ApplicationELB.
4. (Opsional) Untuk melihat metrik di semua dimensi, masukkan namanya di kolom pencarian.
5. (Opsional) Untuk memfilter metrik berdasarkan dimensi, pilih salah satu hal berikut:

- Untuk hanya menampilkan metrik yang dilaporkan untuk penyeimbang beban Anda, pilih Per AppELB Metrics. Untuk melihat metrik untuk satu penyeimbang beban, masukkan namanya di kolom pencarian.
- Untuk hanya menampilkan metrik yang dilaporkan untuk grup target Anda, pilih Per AppELB, per TG Metrics. Untuk melihat metrik untuk satu grup target, masukkan namanya di kolom pencarian.
- Untuk hanya menampilkan metrik yang dilaporkan untuk penyeimbang beban Anda berdasarkan Availability Zone, pilih Per AppELB, per AZ Metrics. Untuk melihat metrik untuk satu penyeimbang beban, masukkan namanya di kolom pencarian. Untuk melihat metrik untuk satu Availability Zone, masukkan namanya di kolom pencarian.
- Untuk hanya menampilkan metrik yang dilaporkan untuk penyeimbang beban Anda berdasarkan Availability Zone dan grup target, pilih Per AppELB, per AZ, per TG Metrics. Untuk melihat metrik untuk satu penyeimbang beban, masukkan namanya di kolom pencarian. Untuk melihat metrik untuk satu grup target, masukkan namanya di kolom pencarian. Untuk melihat metrik untuk satu Availability Zone, masukkan namanya di kolom pencarian.

Untuk melihat metrik menggunakan AWS CLI

Gunakan perintah [list-metrics](#) berikut untuk mencantumkan metrik yang tersedia:

```
aws cloudwatch list-metrics --namespace AWS/ApplicationELB
```

Untuk mendapatkan statistik untuk metrik menggunakan AWS CLI

Gunakan [get-metric-statistics](#) perintah berikut dapatkan statistik untuk metrik dan dimensi yang ditentukan. CloudWatch memperlakukan setiap kombinasi dimensi yang unik sebagai metrik terpisah. Anda tidak dapat mengambil statistik menggunakan kombinasi dimensi yang diterbitkan secara khusus. Anda harus menentukan dimensi yang sama yang digunakan saat metrik dibuat.

```
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
```

Berikut ini adalah contoh output:

```
{
  "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"
}
```

## Log akses untuk Application Load Balancer Anda

Elastic Load Balancing memberikan log akses yang mengambil informasi mendetail tentang permintaan yang dikirim ke penyeimbang beban Anda. Setiap log berisi informasi, seperti waktu permintaan diterima, alamat IP klien, latensi, jalur permintaan, dan respons server. Anda dapat menggunakan log akses ini untuk menganalisis pola lalu lintas dan memecahkan masalah.

Access logs adalah fitur opsional Elastic Load Balancing yang dinonaktifkan secara default. Setelah mengaktifkan log akses untuk penyeimbang beban, Elastic Load Balancing menangkap log dan menyimpannya di bucket Amazon S3 yang Anda tentukan sebagai file terkompresi. Anda dapat menonaktifkan log akses kapan saja.

Anda dikenakan biaya penyimpanan untuk Amazon S3, tetapi tidak dikenakan biaya untuk bandwidth yang digunakan oleh Elastic Load Balancing untuk mengirim berkas log ke Amazon S3. Untuk informasi selengkapnya tentang biaya penyimpanan, lihat [harga Amazon S3](#).

### Daftar Isi

- [Berkas log akses](#)
- [Entri log akses](#)
- [Contoh Entri log](#)
- [Konfigurasi pemberitahuan pengiriman log](#)

- [Memproses berkas log akses](#)
- [Aktifkan log akses untuk Application Load Balancer](#)
- [Nonaktifkan log akses untuk Application Load Balancer](#)

## Berkas log akses

Elastic Load Balancing menerbitkan berkas log untuk setiap simpul penyeimbang beban setiap 5 menit. Pengiriman log pada akhirnya konsisten. Penyeimbang beban dapat mengirimkan beberapa log untuk periode yang sama. Hal ini biasanya terjadi jika situs memiliki lalu lintas tinggi.

Nama file log akses menggunakan format berikut:

```
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

Nama bucket S3 Anda.

prefix

(Opsional) Awalan (hierarki logis) untuk bucket. Awalan yang Anda tentukan tidak boleh menyertakan string AWSLogs. Untuk informasi selengkapnya, lihat [Mengatur objek menggunakan awalan](#).

AWSLogs

Kami menambahkan bagian dari nama file dimulai dengan AWSLogs setelah nama bucket dan awalan opsional yang Anda tentukan.

aws-account-id

ID AWS akun pemilik.

region

Wilayah untuk penyeimbang beban dan bucket S3 Anda.

yyyy/mm/dd

Tanggal pengiriman log.

## load-balancer-id

ID sumber daya penyeimbang beban. Jika ID sumber daya berisi garis miring (/) apa pun, mereka akan diganti dengan titik (.).

## akhir zaman

Tanggal dan waktu interval pengelogan berakhir. Misalnya, waktu akhir 20140215T2340Z berisi entri untuk permintaan yang dibuat antara 23:35 dan 23:40 dalam waktu UTC atau Zulu.

## alamat ip

Alamat IP simpul penyeimbang beban yang menangani permintaan. Untuk penyeimbang beban internal, ini adalah alamat IP privat.

## string acak

String acak yang dihasilkan sistem.

Berikut ini adalah contoh nama file log dengan awalan:

```
s3://amzn-s3-demo-logging-bucket/logging-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
```

Berikut ini adalah contoh nama file log tanpa awalan:

```
s3://amzn-s3-demo-logging-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
```

Anda dapat menyimpan file log dalam bucket selama yang diinginkan, tetapi Anda juga dapat menentukan aturan siklus hidup Amazon S3 untuk mengarsipkan atau menghapus file log secara otomatis. Untuk informasi selengkapnya, lihat [Manajemen siklus hidup objek](#) di Panduan Pengguna Amazon S3.

## Entri log akses

Permintaan log Elastic Load Balancing dikirim ke penyeimbang beban, termasuk permintaan yang tidak pernah sampai ke target. Misalnya, jika klien mengirimkan permintaan yang salah format atau tidak ada target sehat untuk merespons permintaan, permintaan tersebut tetap dicatat.

Setiap entri log berisi rincian permintaan tunggal (atau koneksi dalam kasus WebSockets) yang dibuat ke penyeimbang beban. Untuk WebSockets, entri ditulis hanya setelah koneksi ditutup. Jika koneksi yang ditingkatkan tidak dapat dibuat, entrinya sama dengan permintaan HTTP atau HTTPS.

### Important

Permintaan log Elastic Load Balancing berdasarkan upaya terbaik. Sebaiknya gunakan log akses untuk memahami sifat permintaan, bukan sebagai penghitungan lengkap semua permintaan.

## Daftar Isi

- [Sintaksis](#)
- [Tindakan yang diambil](#)
- [Alasan klasifikasi](#)
- [Kode alasan kesalahan](#)
- [Ubah kode status](#)

## Sintaksis

Tabel berikut menjelaskan bidang entri log akses, secara berurutan. Semua bidang dibatasi oleh spasi. Saat kami menambahkan bidang baru, kami menambahkannya ke akhir entri log. Saat kami bersiap untuk merilis bidang baru, Anda mungkin melihat tambahan “-” sebelum bidang dirilis. Pastikan Anda mengonfigurasi penguraian log untuk berhenti setelah bidang terdokumentasi terakhir, dan perbarui penguraian log setelah kami merilis bidang baru.

Bidang (posisi)	Deskripsi
jenis (1)	Jenis permintaan atau koneksi. Nilai yang mungkin adalah sebagai berikut (abaikan nilai lainnya): <ul style="list-style-type: none"><li>• <code>http</code> — HTTP</li><li>• <code>https</code> — HTTP melalui TLS</li><li>• <code>h2</code> — HTTP/2 melalui TLS</li><li>• <code>grpc</code> — gRPC melalui TLS</li><li>• <code>ws</code> — WebSockets</li></ul>

Bidang (posisi)	Deskripsi
	<ul style="list-style-type: none"> <li>• <code>wss</code>— WebSockets lebih dari TLS</li> </ul>
waktu (2)	Waktu saat penyeimbang beban menghasilkan respons terhadap klien, dalam format ISO 8601. Sebab WebSockets, ini adalah waktu ketika koneksi ditutup.
elb (3)	ID sumber daya penyeimbang beban. Jika Anda mengurai entri log akses, perhatikan bahwa sumber daya IDs dapat berisi garis miring maju (/).
klien: port (4)	Alamat IP dan port dari klien yang meminta. Jika ada proxy di depan penyeimbang beban, bidang ini berisi alamat IP proxy.
target:port (5)	<p>Alamat IP dan port target yang diproses permintaan ini.</p> <p>Jika klien tidak mengirimkan permintaan penuh, penyeimbang beban tidak dapat mengirimkan permintaan ke target, dan nilai ini diatur ke -.</p> <p>Jika target adalah fungsi Lambda, nilai ini diatur ke -.</p> <p>Jika permintaan diblokir oleh AWS WAF, nilai ini diatur ke -.</p>
request_processing_time (6)	<p>Total waktu yang berlalu (dalam detik, dengan presisi milidetik) sejak penyeimbang beban menerima permintaan hingga saat mengirimkan permintaan ke target.</p> <p>Nilai ini diatur ke -1 jika penyeimbang beban tidak dapat mengirimkan permintaan ke target. Hal ini dapat terjadi jika target menutup koneksi sebelum batas waktu idle atau jika klien mengirimkan permintaan yang salah format.</p> <p>Nilai ini juga dapat diatur ke -1 jika koneksi TCP tidak dapat dibuat dengan target sebelum mencapai batas waktu koneksi TCP 10 detik.</p> <p>Jika AWS WAF diaktifkan untuk Application Load Balancer Anda atau jenis target adalah fungsi Lambda, waktu yang dibutuhkan klien untuk mengirim data yang diperlukan untuk permintaan POST dihitung.</p> <p><code>request_processing_time</code></p>

Bidang (posisi)	Deskripsi
target_processing_time (7)	<p>Total waktu yang berlalu (dalam detik, dengan presisi milidetik) sejak penyeimbang beban mengirimkan permintaan ke target hingga target mulai mengirimkan header respons.</p> <p>Nilai ini diatur ke -1 jika penyeimbang beban tidak dapat mengirimkan permintaan ke target. Hal ini dapat terjadi jika target menutup koneksi sebelum batas waktu idle atau jika klien mengirimkan permintaan yang salah format.</p> <p>Nilai ini juga dapat diatur ke -1 jika target terdaftar tidak merespons sebelum batas waktu idle.</p> <p>Jika tidak AWS WAF diaktifkan untuk Application Load Balancer Anda, waktu yang dibutuhkan klien untuk mengirim data yang diperlukan untuk permintaan POST dihitung. target_processing_time</p>
response_processing_time (8)	<p>Total waktu yang berlalu (dalam detik, dengan presisi milidetik) sejak penyeimbang beban menerima header respons dari target hingga mulai mengirimkan respons ke klien. Ini termasuk waktu antrean di penyeimbang beban dan waktu akuisisi koneksi dari penyeimbang beban ke klien.</p> <p>Nilai ini disetel ke -1 jika penyeimbang beban tidak menerima respons dari target. Hal ini dapat terjadi jika target menutup koneksi sebelum batas waktu idle atau jika klien mengirimkan permintaan yang salah format.</p>
elb_status_code (9)	<p>Kode status respons yang dihasilkan oleh penyeimbang beban, aturan respons tetap, atau kode respons AWS WAF khusus untuk tindakan Blokir.</p>
target_status_code (10)	<p>Kode status respons dari target. Nilai ini dicatat hanya jika koneksi dibuat ke target dan target mengirimkan respon. Jika tidak, nilainya diatur ke -.</p>
received_bytes (11)	<p>Ukuran permintaan dalam byte, diterima dari klien (peminta). Untuk permintaan HTTP, ini termasuk header. Untuk WebSockets, ini adalah jumlah total byte yang diterima dari klien pada koneksi.</p>

Bidang (posisi)	Deskripsi
sent_byte (12)	<p>Ukuran respons dalam byte, dikirim ke klien (peminta). Untuk permintaan HTTP, ini termasuk header respon dan body. Untuk WebSockets, ini adalah jumlah total byte yang dikirim ke klien pada koneksi.</p> <p>Header TCP dan payload jabat tangan TLS tidak termasuk dalam sent_bytes. Oleh karena itu sent_bytes tidak akan cocok DataTransfer-Out-Bytes AWS Cost Explorer.</p>
“request_line” (13)	<p>Baris permintaan dari klien, diapit dalam tanda kutip ganda dan dicatat menggunakan format berikut: metode HTTP + protocol: //host:port/uri + versi HTTP. Penyeimbang beban mempertahankan URL yang dikirim oleh klien, sebagaimana adanya, saat mencatat URI permintaan. Ini tidak mengatur jenis konten untuk berkas log akses. Saat Anda memproses bidang ini, pertimbangkan bagaimana klien mengirim URL.</p>
“user_agent” (14)	<p>String Agen-Pengguna yang mengidentifikasi klien yang memulai permintaan, diapit dalam tanda kutip ganda. String-nya terdiri dari satu atau beberapa pengidentifikasi produk, produk[/versi]. Jika string lebih panjang dari 8 KB, string akan terpotong.</p>
ssl_cipher (15)	<p>[Listener HTTPS] Cipher SSL. Nilai ini diatur ke - jika listener bukan listener HTTPS.</p>
ssl_protokol (16)	<p>[Listener HTTPS] Protokol SSL. Nilai ini diatur ke - jika listener bukan listener HTTPS.</p>
target_group_arn (17)	<p>Amazon Resource Name (ARN) dari grup target.</p>
“trace_id” (18)	<p>Isi dari header X-Amzn-Trace-Id, diapit dalam tanda kutip ganda.</p>
“domain_name” (19)	<p>[Listener HTTPS] Domain SNI yang diberikan oleh klien selama handshake TLS, diapit dalam tanda kutip ganda. Nilai ini diatur ke - jika klien tidak mendukung SNI atau domain tidak cocok dengan sertifikat dan sertifikat default disajikan kepada klien.</p>

Bidang (posisi)	Deskripsi
"pilihan_cert_arn" (20)	[Listener HTTPS] ARN sertifikat yang disajikan kepada klien, diapit dalam tanda kutip ganda. Nilai ini diatur ke <code>session-reused</code> jika sesi digunakan kembali. Nilai ini diatur ke <code>-</code> jika listener bukan listener HTTPS.
matched_rule_priority (21)	Nilai prioritas aturan yang cocok dengan permintaan. Jika aturan cocok, ini adalah nilai dari 1 hingga 50.000. Jika tidak ada aturan yang cocok dan tindakan default telah diambil, nilai ini diatur ke 0. Jika terjadi kesalahan selama evaluasi aturan, nilainya diatur ke -1. Untuk kesalahan lainnya, nilainya diatur ke <code>-</code> .
request_creation_time (22)	Waktu saat penyeimbang beban menerima permintaan dari klien, dalam format ISO 8601.
"actions_execute" (23)	Tindakan yang diambil saat memproses permintaan, diapit dalam tanda kutip ganda. Nilai ini adalah daftar yang dipisahkan koma yang dapat menyertakan nilai yang dijelaskan dalam <a href="#">Tindakan yang diambil</a> . Jika tidak ada tindakan yang diambil, seperti untuk permintaan yang salah format, nilai ini diatur ke <code>-</code> .
"redirect_url" (24)	URL target pengalihan untuk header lokasi respons HTTP, diapit dalam tanda kutip ganda. Jika tidak ada tindakan pengalihan yang diambil, nilai ini diatur ke <code>-</code> .
"error_reason" (25)	Kode alasan kesalahan, diapit dalam tanda kutip ganda. Jika permintaan gagal, ini adalah salah satu kode kesalahan yang dijelaskan dalam <a href="#">Kode alasan kesalahan</a> . Jika tindakan yang diambil tidak menyertakan tindakan autentikasi atau target bukan fungsi Lambda, nilai ini diatur ke <code>-</code> .

Bidang (posisi)	Deskripsi
"target:port_list" (26)	<p>Daftar alamat IP dan port yang dipisahkan spasi untuk target yang memproses permintaan ini, diapit dalam tanda kutip ganda. Saat ini, daftar ini dapat berisi satu item dan cocok dengan bidang target:port.</p> <p>Jika klien tidak mengirimkan permintaan penuh, penyeimbang beban tidak dapat mengirimkan permintaan ke target, dan nilai ini diatur ke -.</p> <p>Jika target adalah fungsi Lambda, nilai ini diatur ke -.</p> <p>Jika permintaan diblokir oleh AWS WAF, nilai ini diatur ke -.</p>
"target_status_code_list" (27)	<p>Daftar kode status yang dipisahkan spasi dari respons target, diapit dalam tanda kutip ganda. Saat ini, daftar ini dapat berisi satu item dan cocok dengan bidang target_status_code.</p> <p>Nilai ini dicatat hanya jika koneksi dibuat ke target dan target mengirimkan respon. Jika tidak, nilainya diatur ke -.</p>
"klasifikasi" (28)	<p>Klasifikasi untuk mitigasi desync, diapit dalam tanda kutip ganda. Jika permintaan tidak sesuai dengan RFC 7230, nilai yang mungkin adalah Dapat diterima, Ambigu, dan Parah.</p> <p>Jika permintaan sesuai dengan RFC 7230, nilai ini diatur ke -.</p>
"klasifikasi_alasan" (29)	<p>Kode alasan klasifikasi, diapit dalam tanda kutip ganda. Jika permintaan tidak sesuai dengan RFC 7230, ini adalah salah satu kode klasifikasi yang dijelaskan dalam <a href="#">Alasan klasifikasi</a>. Jika permintaan sesuai dengan RFC 7230, nilai ini diatur ke -.</p>
conn_trace_id (30)	<p>ID ketertelusuran koneksi adalah ID buram unik yang digunakan untuk mengidentifikasi setiap koneksi. Setelah koneksi dibuat dengan klien, permintaan berikutnya dari klien ini akan berisi ID ini di entri log akses masing-masing. ID ini bertindak sebagai kunci asing untuk membuat tautan antara koneksi dan log akses.</p>

Bidang (posisi)	Deskripsi
“transfor med_host” (31)	Header host setelah dimodifikasi oleh transformasi penulisan ulang header host. Jika salah satu dari berikut ini benar, nilai ini diatur ke -. <ul style="list-style-type: none"> <li>• Tidak ada transformasi yang diterapkan</li> <li>• Transformasi gagal</li> <li>• Transformasi berhasil dengan tidak ada perubahan pada header host</li> <li>• Tidak ada header host asli (misalnya, permintaan HTTP/1.0)</li> </ul>
“transformed_uri” (32)	URI setelah dimodifikasi oleh transformasi penulisan ulang URL. Jika salah satu dari berikut ini benar, nilai ini diatur ke -. <ul style="list-style-type: none"> <li>• Tidak ada transformasi yang diterapkan</li> <li>• Transformasi gagal</li> <li>• Transformasi berhasil dengan tidak ada perubahan pada URI</li> </ul>
“request_transform _status” (33)	Status transformasi penulisan ulang. Jika tidak ada transformasi penulisan ulang yang diterapkan, nilai ini diatur ke -. Jika tidak, nilai ini adalah salah satu nilai status yang dijelaskan dalam <a href="#">the section called “Ubah kode status”</a> .

## Tindakan yang diambil

Penyeimbang beban menyimpan tindakan yang diperlukan di bidang `actions_executed` dari log akses.

- `authenticate` — Penyeimbang beban memvalidasi sesi, mengautentikasi pengguna, dan menambahkan informasi pengguna ke header permintaan, seperti yang ditentukan oleh konfigurasi aturan.
- `fixed-response` — Penyeimbang beban mengeluarkan respons tetap, seperti yang ditentukan oleh konfigurasi aturan.
- `forward` — Penyeimbang beban meneruskan permintaan ke target, seperti yang ditentukan oleh konfigurasi aturan.
- `redirect` — Penyeimbang beban mengalihkan permintaan ke URL lain, seperti yang ditentukan oleh konfigurasi aturan.

- `rewrite`— Penyeimbang beban menulis ulang URL permintaan, seperti yang ditentukan oleh konfigurasi aturan.
- `waf` — Penyeimbang beban meneruskan permintaan ke AWS WAF untuk menentukan apakah permintaan harus diteruskan ke target. Jika ini adalah tindakan terakhir, AWS WAF ditentukan bahwa permintaan harus ditolak. Secara default, permintaan yang ditolak oleh AWS WAF akan dicatat sebagai “403” di bidang `e1b_status_code`. Ketika AWS WAF dikonfigurasi untuk menolak permintaan dengan Kode Respons Kustom, `e1b_status_code` bidang akan mencerminkan kode respons yang dikonfigurasi.
- `waf-failed`— Penyeimbang beban berusaha meneruskan permintaan ke AWS WAF, tetapi proses ini gagal.

## Alasan klasifikasi

Jika permintaan tidak sesuai dengan RFC 7230, penyeimbang beban menyimpan salah satu kode berikut di bidang `classification_reason` dari log akses. Untuk informasi selengkapnya, lihat [Mode mitigasi desync](#).

Kode	Deskripsi	Klasifikasi
<code>AmbiguousUri</code>	URI Permintaan berisi karakter kontrol.	Ambigu
<code>BadContentLength</code>	Header Content-Length berisi nilai yang tidak dapat diuraikan atau bukan angka yang valid.	Parah
<code>BadHeader</code>	Header berisi karakter null atau carriage return.	Parah
<code>BadTransferEncoding</code>	Header Transfer-Encoding berisi nilai yang buruk.	Parah
<code>BadUri</code>	URI permintaan berisi karakter null atau carriage return.	Parah
<code>BadMethod</code>	Metode permintaannya salah format.	Parah
<code>BadVersion</code>	Versi permintaannya salah format.	Parah
<code>BothTeClpPresent</code>	Permintaan berisi header Transfer-Encoding dan header Content-Length.	Ambigu

Kode	Deskripsi	Klasifikasi
DuplicateContentLength	Ada beberapa header Content-Length dengan nilai yang sama.	Ambigu
EmptyHeader	Header kosong atau ada garis dengan hanya spasi.	Ambigu
GetHeadZeroContentLength	Ada header Content-Length dengan nilai 0 untuk permintaan GET atau HEAD.	Dapat diterima
MultipleContentLength	Ada beberapa header Content-Length dengan nilai yang berbeda.	Parah
MultipleTransferEncodingChunked	Ada beberapa Transfer-Encoding: chunked header.	Parah
NonCompliantHeader	Header berisi karakter non-ASCII atau kontrol.	Dapat diterima
NonCompliantVersion	Versi permintaan berisi nilai yang buruk.	Dapat diterima
SpaceInUri	URI permintaan berisi spasi yang bukan URL yang dikodekan.	Dapat diterima
SuspiciousHeader	Ada header yang dapat dinormalisasi ke Transfer-Encoding atau Content-Length menggunakan teknik normalisasi teks yang umum.	Ambigu
SuspiciousTeClPresent	Permintaan berisi header Transfer-Encoding dan header Content-Length, dengan setidaknya salah satu dari mereka mencurigakan.	Parah

Kode	Deskripsi	Klasifikasi
UndefinedContentLengthSemantics	Ada header Content-Length yang ditentukan untuk permintaan GET atau HEAD.	Ambigu
UndefinedTransferEncodingSemantics	Ada header Transfer-Encoding yang ditentukan untuk permintaan GET atau HEAD.	Ambigu

## Kode alasan kesalahan

Jika penyeimbang beban tidak dapat menyelesaikan tindakan autentikasi, penyeimbang beban menyimpan salah satu kode alasan berikut di bidang `error_reason` dari log akses. Penyeimbang beban juga menambah metrik yang sesuai CloudWatch . Untuk informasi selengkapnya, lihat [Mengautentikasi pengguna menggunakan Application Load Balancer](#).

Kode	Deskripsi	Metrik
AuthInvalidCookie	Cookie autentikasi tidak valid.	ELBAuthFailure
AuthInvalidGrantError	Kode pemberian otorisasi dari titik akhir token tidak valid.	ELBAuthFailure
AuthInvalidIdToken	Token ID tidak valid.	ELBAuthFailure
AuthInvalidStateParam	Parameter status tidak valid.	ELBAuthFailure
AuthInvalidTokenResponse	Respons dari titik akhir token tidak valid.	ELBAuthFailure

Kode	Deskripsi	Metrik
AuthInvalidUserInfoResponse	Respons dari titik akhir info pengguna tidak valid.	ELBAuthFailure
AuthMissingCodeParam	Respons autentikasi dari titik akhir otorisasi kehilangan parameter kueri bernama 'kode'.	ELBAuthFailure
AuthMissingHostHeader	Respons autentikasi dari titik akhir otorisasi kehilangan bidang header host.	ELBAuthError
AuthMissingStateParam	Respons autentikasi dari titik akhir otorisasi kehilangan parameter kueri bernama 'status'.	ELBAuthFailure
AuthTokenEpRequestFailed	Ada respons kesalahan (non-2xx) dari titik akhir token.	ELBAuthError
AuthTokenEpRequestTimeout	Penyeimbang beban tidak dapat berkomunikasi dengan titik akhir token, atau titik akhir token tidak merespons dalam 5 detik.	ELBAuthError
AuthUnhandledException	Penyeimbang beban mengalami pengecualian yang tidak tertangani.	ELBAuthError
AuthUserInfoEpRequestFailed	Ada respons kesalahan (non-2xx) dari titik akhir info pengguna IdP.	ELBAuthError
AuthUserInfoEpRequestTimeout	Penyeimbang beban tidak dapat berkomunikasi dengan titik akhir info pengguna IDP, atau titik akhir info pengguna tidak merespons dalam 5 detik.	ELBAuthError
AuthUserInfoResponseSizeExceeded	Ukuran klaim yang dikembalikan oleh IdP melebihi 11K byte.	ELBAuthUserClaimsSizeExceeded

Jika penyeimbang beban tidak dapat menyelesaikan tindakan validasi jwt, penyeimbang beban menyimpan salah satu kode alasan berikut di bidang `error_reason` pada log akses. Penyeimbang beban juga menambah metrik yang sesuai CloudWatch . Untuk informasi selengkapnya, lihat [Verifikasi JWTs menggunakan Application Load Balancer](#).

Kode	Deskripsi	Metrik
JWTHeaderNotPresent	Permintaan tidak mengandung header Otorisasi.	JWTValidationFailureCount
JWTRequestFormatInvalid	Token dalam permintaan salah bentuk atau bagian wajib hilang (header, payload, atau tanda tangan), Header tidak mengandung awalan "Pembawa", Header berisi jenis autentikasi yang berbeda seperti "Dasar ", Header otorisasi ada tetapi token tidak ada, jika ada beberapa token yang ada dalam permintaan	JWTValidationFailureCount
JWKSRequestTimeout	Penyeimbang beban tidak dapat berkomunikasi dengan titik akhir JWKS, atau titik akhir JWKS tidak merespons dalam 5 detik.	JWTValidationFailureCount
JWKSResponseSizeExceeded	Ukuran respons yang dikembalikan oleh titik akhir JWKS melebihi 150KB atau jumlah kunci yang dikembalikan oleh titik akhir JWKS melebihi 10.	JWTValidationFailureCount
JWKSRequestFailed	Ada respons kesalahan (non-2XX) dari titik akhir JWKS.	JWTValidationFailureCount
JWKSResponseInvalid	Respons JWKS memiliki satu atau lebih masalah berikut: Format non-JSON, Karakter tidak valid, Format JWKS tidak valid, atribut JWKS Missing/invalid wajib, Kunci publik memiliki algoritma yang tidak didukung, kunci	JWTValidationFailureCount

Kode	Deskripsi	Metrik
	publik tidak dapat dikonversi ke kunci decoding, ukuran kunci publik tidak 2K.	
JWTSignatureValidationError	Gagal memvalidasi tanda tangan token karena alasan apa pun termasuk tanda tangan tidak cocok, Token ditandatangani dengan algoritme yang tidak didukung, KID dalam token tidak ada di titik akhir JWKS.	JWTValidationFailureCount
JWTClaimNotPresent	JWT dalam permintaan klien tidak mengandung klaim yang diperlukan untuk validasi	JWTValidationFailureCount
JWTClaimFormatInvalid	Format nilai klaim di JWT tidak sesuai dengan format yang ditentukan dalam konfigurasi	JWTValidationFailureCount
JWTClaimValueInvalid	Nilai klaim di JWT tidak valid.	JWTValidationFailureCount
JWTValidationInternalError	Penyeimbang beban mengalami kesalahan tak terduga saat memvalidasi JWT dalam permintaan klien.	JWTValidationFailureCount

Jika permintaan ke grup target tertimbang gagal, penyeimbang beban menyimpan salah satu kode kesalahan berikut di bidang `error_reason` dari log akses.

Kode	Deskripsi
AWSALBTGCookieInvalid	AWSALBTG Cookie, yang digunakan dengan kelompok sasaran tertimbang, tidak valid. Misalnya, penyeimbang beban mengembalikan kesalahan ini saat nilai cookie dikodekan ke URL.

Kode	Deskripsi
WeightedTargetGroupsUnhandledException	Penyeimbang beban mengalami pengecualian yang tidak tertangani.

Jika permintaan untuk fungsi Lambda gagal, penyeimbang beban menyimpan salah satu kode alasan berikut di bidang `error_reason` dari log akses. Penyeimbang beban juga menambah metrik yang sesuai CloudWatch . Untuk informasi selengkapnya, lihat tindakan [Pemanggilan](#) Lambda.

Kode	Deskripsi	Metrik
LambdaAccessDenied	Penyeimbang beban tidak memiliki izin untuk memanggil fungsi Lambda.	LambdaUserError
LambdaBadRequest	Pemanggilan Lambda gagal karena header atau isi permintaan klien tidak hanya berisi karakter UTF-8.	LambdaUserError
LambdaConnectionError	Penyeimbang beban tidak dapat terhubung ke Lambda.	LambdaInternalError
LambdaConnectionTimeout	Upaya untuk terhubung ke Lambda habis.	LambdaInternalError
LambdaEC2AccessDeniedException	Amazon EC2 menolak akses ke Lambda selama inisialisasi fungsi.	LambdaUserError
LambdaEC2ThrottledException	Amazon EC2 melambatkan Lambda selama inisialisasi fungsi.	LambdaUserError
LambdaEC2UnexpectedException	Amazon EC2 mengalami pengecualian yang tidak terduga selama inisialisasi fungsi.	LambdaUserError

Kode	Deskripsi	Metrik
LambdaENILimitReachedException	Lambda tidak dapat membuat antarmuka jaringan di VPC yang ditentukan dalam konfigurasi fungsi Lambda karena batas untuk antarmuka jaringan terlampaui.	LambdaUserError
LambdaInvalidResponse	Respons dari fungsi Lambda adalah salah format atau kehilangan bidang yang wajib diisi.	LambdaUserError
LambdaInvalidRuntimeException	Versi waktu aktif Lambda yang ditentukan tidak didukung.	LambdaUserError
LambdaInvalidSecurityGroupIDException	ID grup keamanan yang ditentukan dalam konfigurasi fungsi Lambda tidak valid.	LambdaUserError
LambdaInvalidSubnetIDException	ID subnet yang ditentukan dalam konfigurasi fungsi Lambda tidak valid.	LambdaUserError
LambdaInvalidZipFileException	Lambda tidak dapat membuka file zip fungsi yang ditentukan.	LambdaUserError
LambdaKMSAccessDeniedException	Lambda tidak dapat mendekripsi variabel lingkungan karena akses ke kunci KMS ditolak. Periksa izin KMS fungsi Lambda.	LambdaUserError
LambdaKMSDisabledException	Lambda tidak dapat mendekripsi variabel lingkungan karena kunci KMS yang ditentukan dinonaktifkan. Periksa pengaturan kunci KMS fungsi Lambda.	LambdaUserError

Kode	Deskripsi	Metrik
LambdaKMSInvalidStateException	Lambda tidak dapat mendekripsi variabel lingkungan karena status kunci KMS tidak valid. Periksa pengaturan kunci KMS fungsi Lambda.	LambdaUserError
LambdaKMSNotFoundException	Lambda tidak dapat mendekripsi variabel lingkungan karena kunci KMS tidak ditemukan. Periksa pengaturan kunci KMS fungsi Lambda.	LambdaUserError
LambdaRequestTooLarge	Ukuran isi permintaan melebihi 1 MB.	LambdaUserError
LambdaResourceNotFound	Fungsi Lambda tidak dapat ditemukan.	LambdaUserError
LambdaResponseTooLarge	Ukuran respons melebihi 1 MB.	LambdaUserError
LambdaServiceException	Lambda mengalami kesalahan internal.	LambdaInternalError
LambdaSubnetIPAddressLimitReachedException	Lambda tidak dapat menyiapkan akses VPC untuk fungsi Lambda karena satu atau beberapa subnet tidak memiliki alamat IP yang tersedia.	LambdaUserError
LambdaThrottling	Fungsi Lambda dilambatkan karena ada terlalu banyak permintaan.	LambdaUserError
LambdaUnhandled	Fungsi Lambda mengalami pengecualian yang tidak tertangani.	LambdaUserError
LambdaUnhandledException	Penyeimbang beban mengalami pengecualian yang tidak tertangani.	LambdaInternalError

Kode	Deskripsi	Metrik
LambdaWebsocketNotSupported	WebSockets tidak didukung dengan Lambda.	LambdaUserError

Jika penyeimbang beban mengalami kesalahan saat meneruskan permintaan ke AWS WAF, ia menyimpan salah satu kode kesalahan berikut di bidang `error_reason` dari log akses.

Kode	Deskripsi
WAFConnectionError	Penyeimbang beban tidak dapat terhubung ke AWS WAF.
WAFConnectionTimeout	Koneksi ke AWS WAF timeed out.
WAFResponseReadTimeout	Permintaan untuk AWS WAF kehabisan waktu.
WAFServiceError	AWS WAF mengembalikan kesalahan 5XX.
WAFUnhandledException	Penyeimbang beban mengalami pengecualian yang tidak tertangani.

## Ubah kode status

Kode	Deskripsi
TransformBufferTooSmall	Transformasi penulisan ulang gagal karena hasilnya melebihi ukuran buffer internal. Cobalah untuk membuat ekspresi reguler kurang kompleks.
TransformCompileError	Kompilasi ekspresi reguler gagal.
TransformCompileTooBig	Ekspresi reguler yang dikompilasi terlalu besar. Cobalah untuk membuat ekspresi reguler kurang kompleks.

Kode	Deskripsi
TransformInvalidHost	Transformasi penulisan ulang header host gagal karena host yang dihasilkan tidak valid.
TransformInvalidPath	Transformasi penulisan ulang URL gagal karena jalur yang dihasilkan tidak valid.
TransformRegexSyntaxError	Ekspresi reguler berisi kesalahan sintaks.
TransformReplaceError	Penggantian transformasi gagal.
TransformSuccess	Transformasi penulisan ulang berhasil diselesaikan.

## Contoh Entri log

Berikut ini adalah contoh entri log. Perhatikan bahwa contoh teks muncul di beberapa baris hanya untuk membuatnya lebih mudah dibaca.

### Contoh Entri HTTP

Berikut ini adalah contoh entri log untuk listener HTTP (port 80 ke 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" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

### Contoh Entri HTTPS

Berikut ini adalah contoh entri log untuk listener HTTPS (port 443 ke 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_1234abcd5678ef90 "m.example.com" "-" "TransformSuccess"
```

## Contoh Entri HTTP/2

Berikut ini adalah contoh entri log untuk pengaliran HTTP/2.

```
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" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

## Contoh WebSockets Entri

Berikut ini adalah contoh entri log untuk WebSockets koneksi.

```
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" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

## Contoh Entri Aman WebSockets

Berikut ini adalah contoh entri log untuk WebSockets koneksi aman.

```
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" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

## Contoh Entri untuk Fungsi Lambda

Berikut ini adalah contoh entri log untuk permintaan ke fungsi Lambda yang berhasil:

```
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" "-" "-" "-" "-" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

Berikut ini adalah contoh entri log untuk permintaan ke fungsi Lambda yang gagal:

```
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" "-" "-" "-" "-"
TID_1234abcd5678ef90 "-" "-" "-"
```

## Konfigurasi pemberitahuan pengiriman log

Untuk menerima pemberitahuan saat Elastic Load Balancing mengirimkan log ke bucket S3 Anda, gunakan Pemberitahuan Acara Amazon S3. Elastic Load Balancing menggunakan [PutObject](#), [CreateMultipartUpload](#), dan [POST Object](#) untuk mengirimkan log ke Amazon S3. Untuk memastikan bahwa Anda menerima semua pemberitahuan pengiriman log, sertakan semua peristiwa pembuatan objek ini dalam konfigurasi Anda.

Untuk informasi selengkapnya, lihat [Pemberitahuan Acara Amazon S3](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

## Memproses berkas log akses

Berkas log akses terkompresi. Jika mengunduh file-nya, Anda harus membatalkan kompresinya untuk melihat informasi.

Jika ada banyak permintaan di situs web Anda, penyeimbang beban Anda dapat menghasilkan berkas log dengan gigabyte data. Anda mungkin tidak dapat memproses data dalam jumlah besar menggunakan line-by-line pemrosesan. Oleh karena itu, Anda mungkin harus menggunakan alat analisis yang memberikan solusi pemrosesan paralel. Misalnya, Anda dapat menggunakan alat analisis berikut untuk menganalisis dan memproses log akses:

- Amazon Athena adalah layanan kueri interaktif yang memudahkan untuk menganalisis data di Amazon S3 menggunakan SQL standar. Untuk informasi selengkapnya, lihat [Membuat kueri log Application Load Balancer](#) di Panduan Pengguna Amazon Athena.
- [Loggly](#)
- [Splunk](#)
- [Logika sumo](#)

## Aktifkan log akses untuk Application Load Balancer

Saat mengaktifkan log akses untuk penyeimbang beban, Anda harus menentukan nama bucket S3 tempat penyeimbang beban akan menyimpan log. Bucket harus memiliki kebijakan bucket yang memberikan izin Elastic Load Balancing untuk menulis ke bucket.

### Tugas

- [Langkah 1: Buat ember S3](#)
- [Langkah 2: Lampirkan kebijakan ke bucket S3 Anda](#)
- [Langkah 3: Konfigurasi log akses](#)
- [Langkah 4: Verifikasi izin bucket](#)
- [Pemecahan masalah](#)

### Langkah 1: Buat ember S3

Saat mengaktifkan log akses, Anda harus menentukan bucket S3 untuk log akses. Anda dapat menggunakan bucket yang sudah ada, atau membuat bucket khusus untuk log akses. Bucket harus memenuhi persyaratan berikut.

## Persyaratan

- Bucket harus ditempatkan di Wilayah yang sama dengan penyeimbang beban. Bucket dan load balancer dapat dimiliki oleh akun yang berbeda.
- Satu-satunya opsi enkripsi sisi server yang didukung adalah kunci yang dikelola Amazon S3 (SSE-S3). Untuk informasi selengkapnya, lihat [kunci enkripsi terkelola Amazon S3 \(SSE-S3\)](#).

Untuk membuat bucket S3 menggunakan konsol Amazon S3

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih Buat bucket.
3. Pada halaman Create bucket, lakukan hal berikut:
  - a. Untuk Bucket name, masukkan nama untuk bucket Anda. Nama ini harus unik di semua nama bucket yang ada di Amazon S3. Di beberapa Wilayah, mungkin ada pembatasan tambahan pada nama bucket. Untuk informasi selengkapnya, lihat [Pembatasan dan batasan bucket](#) di Panduan Pengguna Amazon S3.
  - b. Untuk AWS Region, pilih Wilayah tempat Anda membuat penyeimbang beban.
  - c. Untuk enkripsi Default, pilih kunci yang dikelola Amazon S3 (SSE-S3).
  - d. Pilih Buat bucket.

## Langkah 2: Lampirkan kebijakan ke bucket S3 Anda

Bucket S3 Anda harus memiliki kebijakan bucket yang memberikan izin Elastic Load Balancing untuk menulis log akses ke bucket. Kebijakan bucket adalah kumpulan pernyataan JSON yang ditulis dalam bahasa kebijakan akses untuk menentukan izin akses untuk bucket Anda. Setiap pernyataan mencakup informasi tentang satu izin dan berisi serangkaian elemen.

Jika Anda menggunakan bucket yang sudah memiliki kebijakan terlampir, Anda dapat menambahkan pernyataan untuk log akses Elastic Load Balancing ke kebijakan. Jika Anda melakukannya, sebaiknya Anda mengevaluasi kumpulan izin yang dihasilkan untuk memastikan bahwa izin tersebut sesuai untuk pengguna yang memerlukan akses ke bucket untuk log akses.

### Kebijakan bucket

Kebijakan ini memberikan izin ke layanan pengiriman log.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
    }
  ]
}
```

Untuk `Resource`, masukkan ARN lokasi untuk log akses, menggunakan format yang ditunjukkan dalam kebijakan contoh. Selalu sertakan ID akun dengan penyeimbang beban di jalur sumber daya bucket S3 ARN. Ini memastikan bahwa hanya penyeimbang beban dari akun tertentu yang dapat menulis log akses ke bucket S3.

[ARN yang Anda tentukan tergantung pada apakah Anda berencana untuk menyertakan awalan saat Anda mengaktifkan log akses di langkah 3.](#)

Contoh S3 bucket ARN dengan awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket` dan awalannya adalah `logging-prefix`

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

AWS GovCloud (US)— Contoh berikut menggunakan sintaks ARN untuk AWS GovCloud (US) Regions

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Contoh S3 bucket ARN tanpa awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket`. Tidak ada bagian awalan di ember S3 ARN.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

## AWS GovCloud (US)— Contoh berikut menggunakan sintaks ARN untuk. AWS GovCloud (US) Regions

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

### Kebijakan bucket lama

Sebelumnya, untuk Wilayah yang tersedia sebelum Agustus 2022, kami mewajibkan kebijakan yang memberikan izin ke akun Elastic Load Balancing yang khusus untuk Wilayah. Kebijakan lama ini masih didukung, namun sebaiknya Anda menggantinya dengan kebijakan yang lebih baru di atas. Jika Anda lebih suka tetap menggunakan kebijakan lama, yang tidak ditampilkan di sini, Anda bisa.

Sebagai referensi, berikut adalah akun Elastic Load Balancing yang akan ditentukan `Principal` dalam kebijakan lama. IDs Perhatikan bahwa Wilayah yang tidak ada dalam daftar ini tidak mendukung kebijakan lama.

- AS Timur (Virginia N.) — 127311923021
- AS Timur (Ohio) — 033677994240
- AS Barat (California N.) — 027434742980
- AS Barat (Oregon) — 797873946194
- Afrika (Cape Town) — 098369216593
- Asia Pasifik (Hong Kong) — 754344448648
- Asia Pasifik (Jakarta) - 589379963580
- Asia Pasifik (Mumbai) — 718504428378
- Asia Pasifik (Osaka) — 383597477331
- Asia Pasifik (Seoul) — 600734575887
- Asia Pasifik (Singapura) — 114774131450
- Asia Pasifik (Sydney) — 783225319266
- Asia Pasifik (Tokyo) — 582318560864
- Kanada (Tengah) — 985666609251
- Eropa (Frankfurt am Main) — 054676820928
- Eropa (Irlandia) — 156460612806
- Eropa (London) — 652711504416
- Eropa (Milan) — 635631232127

- Eropa (Paris) — 009996457667
- Eropa (Stockholm) — 897822967062
- Timur Tengah (Bahrain) — 076674570225
- Amerika Selatan (São Paulo) — 507241528517
- AWS GovCloud (AS-Timur) — 190560391635
- AWS GovCloud (AS-Barat) — 048591011584

## Zona Outposts

Kebijakan berikut memberikan izin ke layanan pengiriman log yang ditentukan. Gunakan kebijakan ini untuk menyeimbangkan beban di Zona Outposts.

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": "logdelivery.elb.amazonaws.com"
  },
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*",
  "Condition": {
    "StringEquals": {
      "s3:x-amz-acl": "bucket-owner-full-control"
    }
  }
}
```

Untuk `Resource`, masukkan ARN lokasi untuk log akses, menggunakan format yang ditunjukkan dalam kebijakan contoh. Selalu sertakan ID akun dengan penyeimbang beban di jalur sumber daya bucket S3 ARN. Ini memastikan bahwa hanya penyeimbang beban dari akun tertentu yang dapat menulis log akses ke bucket S3.

[ARN bucket S3 yang Anda tentukan bergantung pada apakah Anda berencana untuk menyertakan awalan saat Anda mengaktifkan log akses di langkah 3.](#)

Contoh S3 bucket ARN dengan awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket` dan awalnya adalah `logging-prefix`

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

## Contoh S3 bucket ARN tanpa awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket`. Tidak ada bagian awalan di ember S3 ARN.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

## Praktik terbaik keamanan

- Gunakan jalur sumber daya lengkap, termasuk bagian ID akun dari bucket S3 ARN. Jangan gunakan wildcard (\*) di bagian ID akun ARN bucket S3.

```
"Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
```

- Gunakan `aws:SourceArn` untuk memastikan bahwa hanya penyeimbang beban dari Wilayah dan akun tertentu yang dapat menggunakan bucket Anda.

```
"Condition": {
  "ArnLike": {
    "aws:SourceArn":
      "arn:aws:elasticloadbalancing:region:123456789012:loadbalancer/*"
  }
}
```

- Gunakan `aws:SourceOrgId` dengan `aws:SourceArn` untuk memastikan bahwa hanya penyeimbang beban dari organisasi tertentu yang dapat menggunakan bucket Anda.

```
"Condition": {
  "StringEquals": {
    "aws:SourceOrgId": "o-1234567890"
  },
  "ArnLike": {
    "aws:SourceArn": "arn:aws:elasticloadbalancing:*:*:loadbalancer/*"
  }
}
```

- Jika Anda memiliki Deny pernyataan untuk mencegah akses ke prinsipal layanan kecuali yang diizinkan secara eksplisit, pastikan untuk menambahkan `logdelivery.elasticloadbalancing.amazonaws.com` ke daftar prinsip layanan yang diizinkan. Misalnya, jika Anda menggunakan `aws:PrincipalServiceNamesList` kondisi tersebut, tambahkan `logdelivery.elasticloadbalancing.amazonaws.com` sebagai berikut:

```
{
  "Effect": "Deny",
  "Principal": "*",
  "Condition": {
    "StringNotEqualsIfExists": {
      "aws:PrincipalServiceNamesList": [
        "logdelivery.elasticloadbalancing.amazonaws.com",
        "service.amazonaws.com"
      ]
    }
  }
}
```

Jika Anda menggunakan `NotPrincipal` elemen, tambahkan `logdelivery.elasticloadbalancing.amazonaws.com` sebagai berikut. Perhatikan bahwa kami menyarankan Anda menggunakan kunci `aws:PrincipalServiceName` or `aws:PrincipalServiceNamesList` kondisi untuk secara eksplisit mengizinkan prinsip layanan alih-alih menggunakan elemen `NotPrincipal`. Untuk informasi selengkapnya, lihat [NotPrincipal](#).

```
{
  "Effect": "Deny",
  "NotPrincipal": {
    "Service": [
      "logdelivery.elasticloadbalancing.amazonaws.com",
      "service.amazonaws.com"
    ]
  }
},
```

Setelah membuat kebijakan bucket, gunakan antarmuka Amazon S3, seperti konsol Amazon S3 AWS CLI atau perintah, untuk melampirkan kebijakan bucket ke bucket S3.

## Console

Untuk melampirkan kebijakan bucket Anda ke bucket S3

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih nama bucket untuk membuka halaman detailnya.
3. Pilih Izin lalu pilih Kebijakan Bucket, Edit.

4. Perbarui kebijakan bucket untuk memberikan izin yang diperlukan.
5. Pilih Simpan perubahan.

## AWS CLI

Untuk melampirkan kebijakan bucket Anda ke bucket S3

Gunakan perintah [put-bucket-policy](#). Dalam contoh ini, kebijakan bucket disimpan ke file.json yang ditentukan.

```
aws s3api put-bucket-policy \  
  --bucket amzn-s3-demo-bucket \  
  --policy file://access-log-policy.json
```

## Langkah 3: Konfigurasi log akses

Gunakan prosedur berikut untuk mengonfigurasi log akses untuk menangkap informasi permintaan dan mengirimkan file log ke bucket S3 Anda.

### Persyaratan

Bucket harus memenuhi persyaratan yang dijelaskan pada [langkah 1](#), dan Anda harus melampirkan kebijakan bucket seperti yang dijelaskan pada [langkah 2](#). Jika Anda menyertakan awalan, itu tidak boleh menyertakan string "AWSLogs".

Untuk mengelola bucket S3 untuk log akses Anda

Pastikan untuk menonaktifkan log akses sebelum menghapus bucket yang dikonfigurasi untuk log akses. Jika tidak, jika ada bucket baru dengan nama yang sama dan kebijakan bucket yang diperlukan tetapi dibuat dalam bucket Akun AWS yang tidak Anda miliki, Elastic Load Balancing dapat menulis log akses untuk penyeimbang beban Anda ke bucket baru ini.

### Console

Untuk mengaktifkan log akses

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Di panel navigasi, pilih Load Balancers.

3. Pilih nama penyeimbang beban Anda untuk membuka halaman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Untuk Monitoring, aktifkan Access logs.
6. Untuk URI S3, masukkan URI S3 untuk file log Anda. URI yang Anda tentukan bergantung pada apakah Anda menggunakan awalan.
  - URI dengan awalan: `s3:///amzn-s3-demo-logging-bucketlogging-prefix`
  - URI tanpa awalan: `s3://amzn-s3-demo-logging-bucket`
7. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaktifkan log akses

Gunakan [modify-load-balancer-attributes](#) perintah dengan atribut terkait.

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes \  
    Key=access_logs.s3.enabled,Value=true \  
    Key=access_logs.s3.bucket,Value=amzn-s3-demo-logging-bucket \  
    Key=access_logs.s3.prefix,Value=logging-prefix
```

## CloudFormation

Untuk mengaktifkan log akses

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya untuk menyertakan atribut terkait.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:  
      Name: my-alb  
      Type: application  
      Scheme: internal  
      Subnets:  
        - !Ref subnet-AZ1  
        - !Ref subnet-AZ2
```

```
SecurityGroups:
  - !Ref mySecurityGroup
LoadBalancerAttributes:
  - Key: "access_logs.s3.enabled"
    Value: "true"
  - Key: "access_logs.s3.bucket"
    Value: "amzn-s3-demo-logging-bucket"
  - Key: "access_logs.s3.prefix"
    Value: "logging-prefix"
```

## Langkah 4: Verifikasi izin bucket

Setelah log akses diaktifkan untuk penyeimbang beban Anda, Elastic Load Balancing memvalidasi bucket S3 dan membuat file pengujian untuk memastikan bahwa kebijakan bucket menentukan izin yang diperlukan. Anda dapat menggunakan konsol Amazon S3 untuk memverifikasi bahwa file uji dibuat. File uji bukan berkas log akses yang sebenarnya; file tersebut tidak berisi contoh catatan.

Untuk memverifikasi file pengujian telah dibuat di bucket Anda menggunakan konsol Amazon S3

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih nama bucket yang Anda tentukan untuk log akses.
3. Arahkan ke file pengujian, `ELBAccessLogTestFile`. Lokasi tergantung pada apakah Anda menggunakan awalan.
  - Lokasi dengan awalan: `amzn-s3-demo-logging-bucket//logging-prefix/AWSLogs/123456789012ELBAccessLogTestFile`
  - Lokasi tanpa awalan: `amzn-s3-demo-logging-bucket//AWSLogs/123456789012ELBAccessLogTestFile`

## Pemecahan masalah

Jika Anda menerima kesalahan akses ditolak, berikut ini adalah kemungkinan penyebabnya:

- Kebijakan bucket tidak memberikan izin Elastic Load Balancing untuk menulis log akses ke bucket. Verifikasi bahwa Anda menggunakan kebijakan bucket yang benar untuk Wilayah tersebut. Verifikasi bahwa ARN sumber daya menggunakan nama bucket yang sama dengan yang Anda tentukan saat mengaktifkan log akses. Verifikasi bahwa ARN sumber daya tidak menyertakan awalan jika Anda tidak menentukan awalan saat Anda mengaktifkan log akses.

- Bucket menggunakan opsi enkripsi sisi server yang tidak didukung. Bucket harus menggunakan kunci yang dikelola Amazon S3 (SSE-S3).

## Nonaktifkan log akses untuk Application Load Balancer

Anda dapat menonaktifkan log akses untuk penyeimbang beban Anda kapan saja. Setelah Anda menonaktifkan log akses, log akses Anda tetap berada di bucket S3 hingga Anda menghapusnya. Untuk informasi selengkapnya, lihat [Membuat, mengonfigurasi, dan bekerja dengan bucket S3 di Panduan Pengguna Amazon S3](#).

### Console

Untuk menonaktifkan log akses

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Di panel navigasi, pilih Load Balancers.
3. Pilih nama penyeimbang beban Anda untuk membuka halaman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Untuk Monitoring, matikan log Access.
6. Pilih Simpan perubahan.

### AWS CLI

Untuk menonaktifkan log akses

Gunakan perintah [modify-load-balancer-attributes](#).

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes Key=access_logs.s3.enabled,Value=false
```

## Log koneksi untuk Application Load Balancer

Elastic Load Balancing menyediakan log koneksi yang menangkap informasi terperinci tentang permintaan yang dikirim ke penyeimbang beban Anda. Setiap log berisi informasi seperti alamat IP dan port klien, port pendengar, sandi dan protokol TLS yang digunakan, latensi jabatan tangan

TLS, status koneksi, dan detail sertifikat klien. Anda dapat menggunakan log koneksi ini untuk menganalisis pola permintaan dan memecahkan masalah.

Log koneksi adalah fitur opsional Elastic Load Balancing yang dinonaktifkan secara default. Setelah mengaktifkan log koneksi untuk penyeimbang beban, Elastic Load Balancing menangkap log dan menyimpannya di bucket Amazon S3 yang Anda tentukan, sebagai file terkompresi. Anda dapat menonaktifkan log koneksi kapan saja.

Anda dikenakan biaya penyimpanan untuk Amazon S3, tetapi tidak dikenakan biaya untuk bandwidth yang digunakan oleh Elastic Load Balancing untuk mengirim berkas log ke Amazon S3. Untuk informasi selengkapnya tentang biaya penyimpanan, lihat [harga Amazon S3](#).

## Daftar Isi

- [File log koneksi](#)
- [Entri log koneksi](#)
- [Contoh Entri log](#)
- [Memproses file log koneksi](#)
- [Aktifkan log koneksi untuk Application Load Balancer](#)
- [Nonaktifkan log koneksi untuk Application Load Balancer](#)

## File log koneksi

Elastic Load Balancing menerbitkan berkas log untuk setiap simpul penyeimbang beban setiap 5 menit. Pengiriman log pada akhirnya konsisten. Penyeimbang beban dapat mengirimkan beberapa log untuk periode yang sama. Hal ini biasanya terjadi jika situs memiliki lalu lintas tinggi.

Nama file log koneksi menggunakan format berikut:

```
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

Nama bucket S3 Anda.

## prefix

(Opsional) Awalan (hierarki logis) untuk bucket. Awalan yang Anda tentukan tidak boleh menyertakan string `AWSLogs`. Untuk informasi selengkapnya, lihat [Mengatur objek menggunakan awalan](#).

## AWSLogs

Kami menambahkan bagian dari nama file dimulai dengan `AWSLogs` setelah nama bucket dan awalan opsional yang Anda tentukan.

## aws-account-id

ID AWS akun pemilik.

## region

Wilayah untuk penyeimbang beban dan bucket S3 Anda.

## yyyy/mm/dd

Tanggal pengiriman log.

## load-balancer-id

ID sumber daya penyeimbang beban. Jika ID sumber daya berisi garis miring (`/`) apa pun, mereka akan diganti dengan titik (`.`).

## akhir zaman

Tanggal dan waktu interval pengelogan berakhir. Misalnya, waktu akhir `20140215T2340Z` berisi entri untuk permintaan yang dibuat antara 23:35 dan 23:40 dalam waktu UTC atau Zulu.

## alamat ip

Alamat IP simpul penyeimbang beban yang menangani permintaan. Untuk penyeimbang beban internal, ini adalah alamat IP privat.

## string acak

String acak yang dihasilkan sistem.

Berikut ini adalah contoh nama file log dengan awalan:

```
s3://amzn-s3-demo-logging-bucket/logging-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
```

Berikut ini adalah contoh nama file log tanpa awalan:

```
s3://amzn-s3-demo-logging-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
```

Anda dapat menyimpan file log dalam bucket selama yang diinginkan, tetapi Anda juga dapat menentukan aturan siklus hidup Amazon S3 untuk mengarsipkan atau menghapus file log secara otomatis. Untuk informasi selengkapnya, lihat [Manajemen siklus hidup objek](#) di Panduan Pengguna Amazon S3.

## Entri log koneksi

Setiap upaya koneksi memiliki entri dalam file log koneksi. Bagaimana permintaan klien dikirim ditentukan oleh koneksi yang persisten, atau tidak persisten. Koneksi nonpersistent memiliki satu permintaan, yang menciptakan satu entri di log akses dan log koneksi. Koneksi persisten memiliki beberapa permintaan, yang membuat beberapa entri di log akses dan satu entri di log koneksi.

### Daftar Isi

- [Sintaksis](#)
- [Kode alasan kesalahan](#)

## Sintaksis

Tabel berikut menjelaskan bidang entri log koneksi, secara berurutan. Semua bidang dibatasi oleh spasi. Saat kami menambahkan bidang baru, kami menambahkannya ke akhir entri log. Saat kami bersiap untuk merilis bidang baru, Anda mungkin melihat tambahan “-” sebelum bidang dirilis. Pastikan Anda mengonfigurasi penguraian log untuk berhenti setelah bidang terdokumentasi terakhir, dan perbarui penguraian log setelah kami merilis bidang baru.

Bidang (posisi)	Deskripsi
stempel waktu (1)	Waktu, dalam format ISO 8601, ketika penyeimbang beban berhasil dibuat atau gagal membuat koneksi.

Bidang (posisi)	Deskripsi
client_ip (2)	Alamat IP dari klien yang meminta.
client_port (3)	Port klien yang meminta.
listener_port (4)	Port pendengar penyeimbang beban menerima permintaan klien.
tls_protocol (5)	[HTTPS listener] SSL/TLS Protokol yang digunakan selama jabat tangan. Bidang ini diatur - untuk SSL/TLS non-permintaan.
tls_cipher (6)	[HTTPS listener] SSL/TLS Protokol yang digunakan selama jabat tangan. Bidang ini diatur - untuk SSL/TLS non-permintaan.
tls_handshake_latency (7)	[HTTPS listener] Total waktu dalam hitungan detik, dengan presisi milidetik, berlalu saat membuat jabat tangan yang sukses. Bidang ini diatur ke - kapan: <ul style="list-style-type: none"> <li>• Permintaan yang masuk bukan SSL/TLS permintaan.</li> <li>• Jabat tangan tidak berhasil dibuat.</li> </ul>
leaf_client_certificate_subject (8)	[HTTPS listener] Nama subjek dari sertifikat klien daun. Bidang ini diatur ke - kapan: <ul style="list-style-type: none"> <li>• Permintaan yang masuk bukan SSL/TLS permintaan.</li> <li>• Pendengar penyeimbang beban tidak dikonfigurasi dengan mTL diaktifkan.</li> <li>• Server tidak dapat sertifikat klien daun. load/parse</li> </ul>
leaf_client_certificate_validity (9)	[HTTPS listener] Validitas, dengan not-before dan not-after dalam format ISO 8601, dari sertifikat klien daun. Bidang ini diatur ke - kapan: <ul style="list-style-type: none"> <li>• Permintaan yang masuk bukan SSL/TLS permintaan.</li> <li>• Pendengar penyeimbang beban tidak dikonfigurasi dengan mTL diaktifkan.</li> <li>• Server tidak dapat sertifikat klien daun. load/parse</li> </ul>

Bidang (posisi)	Deskripsi
leaf_client_cert_serial_number (10)	<p>[HTTPS listener] Nomor seri sertifikat klien daun. Bidang ini diatur ke - kapan:</p> <ul style="list-style-type: none"> <li>• Permintaan yang masuk bukan SSL/TLS permintaan.</li> <li>• Pendengar penyeimbang beban tidak dikonfigurasi dengan mTL diaktifkan.</li> <li>• Server tidak dapat sertifikat klien daun. load/parse</li> </ul>
tls_verify_status (11)	<p>[HTTPS listener] Status permintaan koneksi. Nilai ini adalah Success jika koneksi berhasil dibuat. Pada koneksi yang gagal nilainya adalahFailed:\$error_code .</p>
conn_trace_id (12)	<p>ID ketertelusuran koneksi adalah ID buram unik yang digunakan untuk mengidentifikasi setiap koneksi. Setelah koneksi dibuat dengan klien, permintaan berikutnya dari klien ini berisi ID ini di entri log akses masing-masing. ID ini bertindak sebagai kunci asing untuk membuat tautan antara koneksi dan log akses.</p>
tls_keyexchange (13)	<p>[HTTPS listener] Pertukaran kunci yang digunakan selama jabat tangan untuk TLS atau PQ-TLS. Bidang ini diatur - untuk SSL/TLS non-permintaan.</p>

## Kode alasan kesalahan

Jika penyeimbang beban tidak dapat membuat koneksi, penyeimbang beban menyimpan salah satu kode alasan berikut di log koneksi.

Kode	Deskripsi
ClientCertificateMaxChainDepthExceeded	Kedalaman rantai sertifikat klien maksimum telah terlampaui

Kode	Deskripsi	
ClientCertificateMaximumSizeExceeded	Ukuran sertifikat klien maksimum telah terlampaui	
ClientCertificateRevoked	Sertifikat klien telah dicabut oleh CA	
ClientCertificateProcessingError	Kesalahan pemrosesan CRL	
ClientCertificateUntrusted	Sertifikat klien tidak dipercaya	
ClientCertificateNotYetValid	Sertifikat klien belum valid	
ClientCertificateExpired	Sertifikat klien kedaluwarsa	
ClientCertificateTypeUnsupported	Jenis sertifikat klien tidak didukung	
ClientCertificateInvalid	Sertifikat klien tidak valid	
ClientCertificatePurposeInvalid	Tujuan sertifikat klien tidak valid	
ClientCertificateRejected	Sertifikat klien ditolak oleh validasi server kustom	
UnmappedConnectionError	Kesalahan koneksi runtime yang tidak dipetakan	

## Contoh Entri log

Berikut ini adalah contoh entri log koneksi. Perhatikan bahwa contoh teks muncul di beberapa baris hanya untuk membuatnya lebih mudah dibaca.

Berikut ini adalah contoh entri log untuk koneksi yang berhasil dengan pendengar HTTPS dengan modus verifikasi TLS timbal balik diaktifkan pada port 443.

```
2023-10-04T17:05:15.514108Z 203.0.113.1 36280 443 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256
4.036
"CN=amazondomains.com,0=endEntity,L=Seattle,ST=Washington,C=US"
NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z
FEF257372D5C14D4 Success TID_3180a73013c8ca4bac2f731159d4b0fe
```

Berikut ini adalah contoh entri log untuk koneksi yang gagal dengan pendengar HTTPS dengan modus verifikasi TLS timbal balik diaktifkan pada port 443.

```
2023-10-04T17:05:15.514108Z 203.0.113.1 36280 443 TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256
-
"CN=amazondomains.com,0=endEntity,L=Seattle,ST=Washington,C=US"
NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z
FEF257372D5C14D4 Failed:ClientCertUntrusted TID_1c71a68d70587445ad5127ff8b2687d7
```

## Memproses file log koneksi

File log koneksi dikompresi. Jika Anda membuka file menggunakan konsol Amazon S3, file tersebut tidak terkompresi dan informasinya ditampilkan. Jika mengunduh file-nya, Anda harus membatalkan kompresinya untuk melihat informasi.

Jika ada banyak permintaan di situs web Anda, penyeimbang beban Anda dapat menghasilkan berkas log dengan gigabyte data. Anda mungkin tidak dapat memproses data dalam jumlah besar menggunakan line-by-line pemrosesan. Oleh karena itu, Anda mungkin harus menggunakan alat analisis yang memberikan solusi pemrosesan paralel. Misalnya, Anda dapat menggunakan alat analisis berikut untuk menganalisis dan memproses log koneksi:

- Amazon Athena adalah layanan kueri interaktif yang memudahkan untuk menganalisis data di Amazon S3 menggunakan SQL standar.
- [Loggly](#)
- [Splunk](#)

- [Logika sumo](#)

## Aktifkan log koneksi untuk Application Load Balancer

Saat mengaktifkan log koneksi untuk penyeimbang beban, Anda harus menentukan nama bucket S3 tempat penyeimbang beban akan menyimpan log. Bucket harus memiliki kebijakan bucket yang memberikan izin Elastic Load Balancing untuk menulis ke bucket.

### Tugas

- [Langkah 1: Buat ember S3](#)
- [Langkah 2: Lampirkan kebijakan ke bucket S3 Anda](#)
- [Langkah 3: Konfigurasi log koneksi](#)
- [Langkah 4: Verifikasi izin bucket](#)
- [Pemecahan masalah](#)

### Langkah 1: Buat ember S3

Saat Anda mengaktifkan log koneksi, Anda harus menentukan bucket S3 untuk log koneksi. Anda dapat menggunakan bucket yang sudah ada, atau membuat bucket khusus untuk log koneksi. Bucket harus memenuhi persyaratan berikut.

### Persyaratan

- Bucket harus ditempatkan di Wilayah yang sama dengan penyeimbang beban. Bucket dan load balancer dapat dimiliki oleh akun yang berbeda.
- Satu-satunya opsi enkripsi sisi server yang didukung adalah kunci yang dikelola Amazon S3 (SSE-S3). Untuk informasi selengkapnya, lihat [kunci enkripsi terkelola Amazon S3 \(SSE-S3\)](#).

Untuk membuat bucket S3 menggunakan konsol Amazon S3

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih Buat bucket.
3. Pada halaman Create bucket, lakukan hal berikut:
  - a. Untuk Bucket name, masukkan nama untuk bucket Anda. Nama ini harus unik di semua nama bucket yang ada di Amazon S3. Di beberapa Wilayah, mungkin ada pembatasan

tambahan pada nama bucket. Untuk informasi selengkapnya, lihat [Pembatasan dan batasan bucket](#) di Panduan Pengguna Amazon S3.

- b. Untuk AWS Region, pilih Wilayah tempat Anda membuat penyeimbang beban.
- c. Untuk enkripsi Default, pilih kunci yang dikelola Amazon S3 (SSE-S3).
- d. Pilih Buat bucket.

## Langkah 2: Lampirkan kebijakan ke bucket S3 Anda

Bucket S3 Anda harus memiliki kebijakan bucket yang memberikan izin Elastic Load Balancing untuk menulis log koneksi ke bucket. Kebijakan bucket adalah kumpulan pernyataan JSON yang ditulis dalam bahasa kebijakan akses untuk menentukan izin akses untuk bucket Anda. Setiap pernyataan mencakup informasi tentang satu izin dan berisi serangkaian elemen.

Jika Anda menggunakan bucket yang sudah memiliki kebijakan terlampir, Anda dapat menambahkan pernyataan untuk log koneksi Elastic Load Balancing ke kebijakan. Jika Anda melakukannya, sebaiknya Anda mengevaluasi kumpulan izin yang dihasilkan untuk memastikan bahwa izin tersebut sesuai untuk pengguna yang memerlukan akses ke bucket untuk log koneksi.

### Kebijakan bucket

Kebijakan ini memberikan izin ke layanan pengiriman log yang ditentukan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
    }
  ]
}
```

Untuk `Resource`, masukkan ARN lokasi untuk log akses, menggunakan format yang ditunjukkan dalam kebijakan contoh. Selalu sertakan ID akun akun dengan penyeimbang beban di jalur sumber

daya bucket S3 ARN. Ini memastikan bahwa hanya penyeimbang beban dari akun tertentu yang dapat menulis log akses ke bucket S3.

[ARN yang Anda tentukan tergantung pada apakah Anda berencana untuk menyertakan awalan saat Anda mengaktifkan log akses di langkah 3.](#)

Contoh S3 bucket ARN dengan awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket` dan awalannya adalah `logging-prefix`

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

AWS GovCloud (US)— Contoh berikut menggunakan sintaks ARN untuk AWS GovCloud (US) Regions

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Contoh S3 bucket ARN tanpa awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket`. Tidak ada bagian awalan di ember S3 ARN.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

AWS GovCloud (US)— Contoh berikut menggunakan sintaks ARN untuk AWS GovCloud (US) Regions

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

Kebijakan bucket lama

Sebelumnya, untuk Wilayah yang tersedia sebelum Agustus 2022, kami mewajibkan kebijakan yang memberikan izin ke akun Elastic Load Balancing yang khusus untuk Wilayah. Kebijakan lama ini masih didukung, namun sebaiknya Anda menggantinya dengan kebijakan yang lebih baru di atas. Jika Anda lebih suka tetap menggunakan kebijakan lama, yang tidak ditampilkan di sini, Anda bisa.

Sebagai referensi, berikut adalah akun Elastic Load Balancing yang akan ditentukan `Principal` dalam kebijakan lama. IDs Perhatikan bahwa Wilayah yang tidak ada dalam daftar ini tidak mendukung kebijakan lama.

- AS Timur (Virginia N.) — 127311923021

- AS Timur (Ohio) — 033677994240
- AS Barat (California N.) — 027434742980
- AS Barat (Oregon) — 797873946194
- Afrika (Cape Town) — 098369216593
- Asia Pasifik (Hong Kong) — 754344448648
- Asia Pasifik (Jakarta) - 589379963580
- Asia Pasifik (Mumbai) — 718504428378
- Asia Pasifik (Osaka) — 383597477331
- Asia Pasifik (Seoul) — 600734575887
- Asia Pasifik (Singapura) — 114774131450
- Asia Pasifik (Sydney) — 783225319266
- Asia Pasifik (Tokyo) — 582318560864
- Kanada (Tengah) — 985666609251
- Eropa (Frankfurt am Main) — 054676820928
- Eropa (Irlandia) — 156460612806
- Eropa (London) — 652711504416
- Eropa (Milan) — 635631232127
- Eropa (Paris) — 009996457667
- Eropa (Stockholm) — 897822967062
- Timur Tengah (Bahrain) — 076674570225
- Amerika Selatan (São Paulo) — 507241528517
- AWS GovCloud (AS-Timur) — 190560391635
- AWS GovCloud (AS-Barat) — 048591011584

## Zona Outposts

Kebijakan berikut memberikan izin ke layanan pengiriman log yang ditentukan. Gunakan kebijakan ini untuk menyeimbangkan beban di Zona Outposts.

```
{
  "Effect": "Allow",
  "Principal": {
```

```

    "Service": "logdelivery.elb.amazonaws.com"
  },
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
  "Condition": {
    "StringEquals": {
      "s3:x-amz-acl": "bucket-owner-full-control"
    }
  }
}

```

Untuk `Resource`, masukkan ARN lokasi untuk log akses. Selalu sertakan ID akun dengan penyeimbang beban di jalur sumber daya bucket S3 ARN. Ini memastikan bahwa hanya penyeimbang beban dari akun tertentu yang dapat menulis log akses ke bucket S3.

[ARN yang Anda tentukan tergantung pada apakah Anda berencana untuk menyertakan awalan saat Anda mengaktifkan log akses di langkah 3.](#)

Contoh S3 bucket ARN dengan awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket` dan awalnya adalah `logging-prefix`

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Contoh S3 bucket ARN tanpa awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket`. Tidak ada bagian awalan di ember S3 ARN.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

Praktik terbaik keamanan

Untuk meningkatkan keamanan, gunakan bucket ARNs S3 yang tepat.

- Gunakan jalur sumber daya lengkap, bukan hanya ARN bucket S3.
- Sertakan bagian ID akun dari bucket S3 ARN.
- Jangan gunakan wildcard (\*) di bagian ID akun ARN bucket S3.

Setelah membuat kebijakan bucket, gunakan antarmuka Amazon S3, seperti konsol Amazon S3 AWS CLI atau perintah, untuk melampirkan kebijakan bucket ke bucket S3.

## Console

Untuk melampirkan kebijakan bucket Anda ke bucket S3

1. Buka konsol Amazon S3 di. <https://console.aws.amazon.com/s3/>
2. Pilih nama bucket untuk membuka halaman detailnya.
3. Pilih Izin lalu pilih Kebijakan Bucket, Edit.
4. Perbarui kebijakan bucket untuk memberikan izin yang diperlukan.
5. Pilih Simpan perubahan.

## AWS CLI

Untuk melampirkan kebijakan bucket Anda ke bucket S3

Gunakan perintah [put-bucket-policy](#). Dalam contoh ini, kebijakan bucket disimpan ke file.json yang ditentukan.

```
aws s3api put-bucket-policy \  
  --bucket amzn-s3-demo-bucket \  
  --policy file://access-log-policy.json
```

## Langkah 3: Konfigurasi log koneksi

Gunakan prosedur berikut untuk mengonfigurasi log koneksi untuk menangkap dan mengirimkan file log ke bucket S3 Anda.

### Persyaratan

Bucket harus memenuhi persyaratan yang dijelaskan pada [langkah 1](#), dan Anda harus melampirkan kebijakan bucket seperti yang dijelaskan pada [langkah 2](#). Jika Anda menentukan awalan, itu tidak harus menyertakan string "AWSLogs".

Untuk mengelola bucket S3 untuk log koneksi Anda

Pastikan untuk menonaktifkan log koneksi sebelum Anda menghapus bucket yang Anda konfigurasi untuk log koneksi. Jika tidak, jika ada bucket baru dengan nama yang sama dan kebijakan bucket yang diperlukan tetapi dibuat dalam bucket Akun AWS yang tidak Anda miliki, Elastic Load Balancing dapat menulis log koneksi untuk penyeimbang beban Anda ke bucket baru ini.

## Console

Untuk mengaktifkan log koneksi

1. Buka konsol Amazon EC2 di. <https://console.aws.amazon.com/ec2/>
2. Di panel navigasi, pilih Load Balancers.
3. Pilih nama penyeimbang beban Anda untuk membuka halaman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Untuk Pemantauan, nyalakan log Koneksi.
6. Untuk URI S3, masukkan URI S3 untuk file log Anda. URI yang Anda tentukan bergantung pada apakah Anda menggunakan awalan.
  - URI dengan awalan: `s3://bucket-name/prefix`
  - URI tanpa awalan: `s3://bucket-name`
7. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaktifkan log koneksi

Gunakan [modify-load-balancer-attributes](#) perintah dengan atribut terkait.

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes \  
    Key=connection_logs.s3.enabled,Value=true \  
    Key=connection_logs.s3.bucket,Value=amzn-s3-demo-logging-bucket \  
    Key=connection_logs.s3.prefix,Value=logging-prefix
```

## CloudFormation

Untuk mengaktifkan log koneksi

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya untuk menyertakan atribut terkait.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
```

```
Properties:
  Name: my-alb
  Type: application
  Scheme: internal
  Subnets:
    - !Ref subnet-AZ1
    - !Ref subnet-AZ2
  SecurityGroups:
    - !Ref mySecurityGroup
  LoadBalancerAttributes:
    - Key: "connection_logs.s3.enabled"
      Value: "true"
    - Key: "connection_logs.s3.bucket"
      Value: "amzn-s3-demo-logging-bucket"
    - Key: "connection_logs.s3.prefix"
      Value: "logging-prefix"
```

## Langkah 4: Verifikasi izin bucket

Setelah log koneksi diaktifkan untuk penyeimbang beban Anda, Elastic Load Balancing memvalidasi bucket S3 dan membuat file pengujian untuk memastikan bahwa kebijakan bucket menentukan izin yang diperlukan. Anda dapat menggunakan konsol Amazon S3 untuk memverifikasi bahwa file uji dibuat. File pengujian bukan file log koneksi yang sebenarnya; itu tidak berisi catatan contoh.

Untuk memverifikasi bahwa Elastic Load Balancing membuat file uji di bucket S3 Anda

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih nama bucket yang Anda tentukan untuk log koneksi.
3. Arahkan ke file pengujian, `ELBConnectionLogTestFile`. Lokasi tergantung pada apakah Anda menggunakan awalan.
  - Lokasi dengan awalan: `amzn-s3-demo-logging-bucket//prefix/AWSLogs/123456789012ELBConnectionLogTestFile`
  - Lokasi tanpa awalan: `amzn-s3-demo-logging-bucket//AWSLogs/123456789012ELBConnectionLogTestFile`

## Pemecahan masalah

Jika Anda menerima kesalahan akses ditolak, berikut ini adalah kemungkinan penyebabnya:

- Kebijakan bucket tidak memberikan izin Elastic Load Balancing untuk menulis log koneksi ke bucket. Verifikasi bahwa Anda menggunakan kebijakan bucket yang benar untuk Wilayah tersebut. Verifikasi bahwa ARN sumber daya menggunakan nama bucket yang sama dengan yang Anda tentukan saat mengaktifkan log koneksi. Verifikasi bahwa ARN sumber daya tidak menyertakan awalan jika Anda tidak menentukan awalan saat Anda mengaktifkan log koneksi.
- Bucket menggunakan opsi enkripsi sisi server yang tidak didukung. Bucket harus menggunakan kunci yang dikelola Amazon S3 (SSE-S3).

## Nonaktifkan log koneksi untuk Application Load Balancer

Anda dapat menonaktifkan log koneksi untuk penyeimbang beban Anda kapan saja. Setelah Anda menonaktifkan log koneksi, log koneksi Anda tetap berada di bucket S3 hingga Anda menghapusnya. Untuk informasi selengkapnya, lihat [Membuat, mengonfigurasi, dan bekerja dengan bucket](#) di Panduan Pengguna Amazon S3.

### Console

Untuk menonaktifkan log koneksi

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Di panel navigasi, pilih Load Balancers.
3. Pilih nama penyeimbang beban Anda untuk membuka halaman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Untuk Pemantauan, matikan log Koneksi.
6. Pilih Simpan perubahan.

### AWS CLI

Untuk menonaktifkan log koneksi

Gunakan perintah [modify-load-balancer-attributes](#).

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes Key=connection_logs.s3.enabled,Value=false
```

## Log pemeriksaan kesehatan

Elastic Load Balancing menyediakan log pemeriksaan kesehatan yang menangkap informasi terperinci tentang status pemeriksaan kesehatan target terdaftar Anda, termasuk alasan kegagalan saat pemeriksaan kesehatan gagal. Log pemeriksaan kesehatan didukung untuk instans EC2, alamat IP, dan target fungsi Lambda. Setiap entri log berisi informasi seperti jenis atau koneksi permintaan pemeriksaan kesehatan, stempel waktu, alamat target, ID grup target, status kesehatan, dan kode alasan. Anda dapat menggunakan log pemeriksaan kesehatan ini untuk menganalisis pola kesehatan target, memantau transisi kesehatan, dan memecahkan masalah.

Health check logs adalah fitur opsional yang dinonaktifkan secara default. Setelah mengaktifkan log pemeriksaan kesehatan untuk penyeimbang beban, Elastic Load Balancing menangkap log dan menyimpannya sebagai file terkompresi di bucket Amazon S3 yang Anda tentukan. Anda dapat menonaktifkan log pemeriksaan kesehatan kapan saja.

Anda dikenakan biaya penyimpanan untuk Amazon S3, tetapi tidak dikenakan biaya untuk bandwidth yang digunakan oleh Elastic Load Balancing untuk mengirim berkas log ke Amazon S3. Untuk informasi selengkapnya tentang biaya penyimpanan, lihat [harga Amazon S3](#).

### Daftar Isi

- [File log pemeriksaan kesehatan](#)
- [Entri log pemeriksaan kesehatan](#)
- [Contoh Entri log](#)
- [Konfigurasi pemberitahuan pengiriman log](#)
- [Memproses file log pemeriksaan kesehatan](#)
- [Aktifkan log pemeriksaan kesehatan untuk Application Load Balancer](#)
- [Nonaktifkan log pemeriksaan kesehatan untuk Application Load Balancer](#)

## File log pemeriksaan kesehatan

Elastic Load Balancing menerbitkan berkas log untuk setiap simpul penyeimbang beban setiap 5 menit. Penyeimbang beban dapat mengirimkan beberapa log untuk periode yang sama ketika sejumlah besar target dilampirkan ke penyeimbang beban atau interval pemeriksaan kesehatan kecil dikonfigurasi (misalnya, setiap 5 detik).

Nama file log pemeriksaan kesehatan menggunakan format berikut:

```
bucket[/prefix]/AWSLogs/aws-account-id/elasticloadbalancing/region/yyyy/mm/dd/  
health_check_log_aws-account-id_elasticloadbalancing_region_app.load-balancer-id_end-  
time_ip-address_random-string.log.gz
```

## bucket

Nama bucket S3 Anda.

## prefix

(Opsional) Awalan (hierarki logis) untuk bucket. Awalan yang Anda tentukan tidak boleh menyertakan string AWSLogs. Untuk informasi selengkapnya, lihat [Mengatur objek menggunakan awalan](#).

## AWSLogs

Kami menambahkan bagian dari nama file dimulai dengan AWSLogs setelah nama bucket dan awalan opsional yang Anda tentukan.

## aws-account-id

ID AWS akun pemilik.

## region

Wilayah untuk penyeimbang beban dan bucket S3 Anda.

## yyyy/mm/dd

Tanggal pengiriman log.

## load-balancer-id

ID sumber daya penyeimbang beban. Jika ID sumber daya berisi garis miring (/) apa pun, mereka akan diganti dengan titik (.).

## akhir zaman

Tanggal dan waktu interval pengelogan berakhir. Misalnya, waktu akhir 20140215T2340Z berisi entri untuk permintaan yang dibuat antara 23:35 dan 23:40 dalam waktu UTC atau Zulu.

## alamat ip

Alamat IP simpul penyeimbang beban yang menangani permintaan. Untuk penyeimbang beban internal, ini adalah alamat IP privat.

## string acak

String acak yang dihasilkan sistem.

Berikut ini adalah contoh nama file log dengan awalan:

```
s3://amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/elasticloadbalancing/us-east-2/2022/05/01/health_check_log_123456789012_elasticloadbalancing_us-east-2_app.my-loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Berikut ini adalah contoh nama file log tanpa awalan:

```
s3://amzn-s3-demo-logging-bucket/AWSLogs/123456789012/elasticloadbalancing/us-east-2/2022/05/01/health_check_log_123456789012_elasticloadbalancing_us-east-2_app.my-loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Anda dapat menyimpan file log dalam bucket selama yang diinginkan, tetapi Anda juga dapat menentukan aturan siklus hidup Amazon S3 untuk mengarsipkan atau menghapus file log secara otomatis. Untuk informasi selengkapnya, lihat [Manajemen siklus hidup objek](#) di Panduan Pengguna Amazon S3.

## Entri log pemeriksaan kesehatan

Log Elastic Load Balancing menargetkan hasil pemeriksaan kesehatan termasuk alasan kegagalan untuk semua target yang terdaftar dari penyeimbang beban tersebut. Setiap entri log berisi rincian hasil pemeriksaan kesehatan tunggal yang dibuat untuk target yang terdaftar.

### Daftar Isi

- [Sintaksis](#)
- [Kode alasan kesalahan](#)

### Sintaksis

Tabel berikut menjelaskan bidang entri log pemeriksaan kesehatan, secara berurutan. Semua bidang dibatasi oleh spasi. Saat kami menambahkan bidang baru, kami menambahkannya ke akhir entri log. Saat kami bersiap untuk merilis bidang baru, Anda mungkin melihat tambahan “-” sebelum bidang

dirilis. Pastikan Anda mengonfigurasi penguraian log untuk berhenti setelah bidang terdokumentasi terakhir, dan perbarui penguraian log setelah kami merilis bidang baru.

Bidang (posisi)	Deskripsi
jenis (1)	Jenis permintaan atau koneksi pemeriksaan kesehatan. Nilai yang mungkin adalah sebagai berikut (abaikan nilai lainnya): <ul style="list-style-type: none"> <li>• <code>http</code>-- HTTP</li> <li>• <code>https</code>- HTTP melalui TLS</li> <li>• <code>h2</code>- HTTP/2 melalui TLS</li> <li>• <code>grpc</code>- gRPC</li> <li>• <code>lambda</code>- Fungsi Lambda</li> </ul>
waktu (2)	Stempel waktu ketika pemeriksaan kesehatan dimulai pada target, dalam format ISO 8601.
latensi (3)	Total waktu berlalu (dalam hitungan detik) untuk menyelesaikan pemeriksaan kesehatan saat ini.
target_addr (4)	Alamat IP dan port target dalam format, IP: port. ARN Lambda jika targetnya adalah fungsi Lambda.
target_group_id (5)	Nama kelompok sasaran yang terkait dengan target.
Status (6)	Status pemeriksaan kesehatan. Nilai ini PASS jika pemeriksaan kesehatan berhasil. Pada pemeriksaan kesehatan yang gagal nilainya FAIL
status_code (7)	Kode respon diterima dari target untuk permintaan pemeriksaan kesehatan.
alasan_kode (8)	Alasan kegagalan jika pemeriksaan kesehatan gagal. Lihat <a href="#">Kode alasan kesalahan</a>

## Kode alasan kesalahan

Jika pemeriksaan kesehatan target gagal, penyeimbang beban akan mencatat salah satu kode alasan berikut di log pemeriksaan kesehatan.

Kode	Deskripsi
<code>RequestTimedOut</code>	Permintaan pemeriksaan kesehatan habis waktu sambil menunggu tanggapan
<code>ConnectionTimedOut</code>	Pemeriksaan kesehatan gagal karena upaya koneksi TCP habis
<code>ConnectionReset</code>	Pemeriksaan kesehatan gagal karena penyetelan ulang koneksi
<code>ResponseCodeMismatch</code>	Kode status HTTP dari respons target terhadap permintaan pemeriksaan kesehatan tidak cocok dengan kode status yang dikonfigurasi
<code>ResponseStringMismatch</code>	Badan respons yang dikembalikan oleh target tidak berisi string yang dikonfigurasi dalam konfigurasi pemeriksaan kesehatan grup target
<code>InternalError</code>	Kesalahan penyeimbang beban internal
<code>TargetError</code>	Target mengembalikan kode kesalahan 5xx sebagai tanggapan atas permintaan pemeriksaan kesehatan
<code>GRPCStatusHeaderEmpty</code>	Respons target GRPC memiliki header <code>grpc-status</code> tanpa nilai
<code>GRPCUnexpectedStatus</code>	Target GRPC merespons dengan status <code>grpc</code> yang tidak terduga

## Contoh Entri log

Berikut ini adalah contoh entri log pemeriksaan kesehatan. Perhatikan bahwa contoh teks muncul di beberapa baris hanya untuk membuatnya lebih mudah dibaca.

Berikut ini adalah contoh entri log untuk pemeriksaan kesehatan yang sukses.

```
http 2025-10-31T12:44:59.875678Z 0.019584011 172.31.20.97:80 HCLogsTestIPs PASS 200 -
```

Berikut ini adalah contoh entri log untuk pemeriksaan kesehatan yang gagal.

```
http 2025-10-31T12:44:58.901409Z 1.121980746 172.31.31.9:80 HCLogsTestIPs FAIL 502  
TargetError
```

## Konfigurasi pemberitahuan pengiriman log

Untuk menerima pemberitahuan saat Elastic Load Balancing mengirimkan log ke bucket S3 Anda, gunakan Pemberitahuan Acara Amazon S3. Elastic Load Balancing menggunakan [PutObject](#), [CreateMultipartUpload](#), dan [POST Object](#) untuk mengirimkan log ke Amazon S3. Untuk memastikan bahwa Anda menerima semua pemberitahuan pengiriman log, sertakan semua peristiwa pembuatan objek ini dalam konfigurasi Anda.

Untuk informasi selengkapnya, lihat [Pemberitahuan Acara Amazon S3](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

## Memproses file log pemeriksaan kesehatan

File log pemeriksaan kesehatan dikompresi. Jika mengunduh file-nya, Anda harus membatalkan kompresinya untuk melihat informasi.

Jika ada banyak permintaan di situs web Anda, penyeimbang beban Anda dapat menghasilkan berkas log dengan gigabyte data. Anda mungkin tidak dapat memproses data dalam jumlah besar menggunakan line-by-line pemrosesan. Oleh karena itu, Anda mungkin harus menggunakan alat analisis yang memberikan solusi pemrosesan paralel. Misalnya, Anda dapat menggunakan alat analisis berikut untuk menganalisis dan memproses log pemeriksaan kesehatan:

- Amazon Athena adalah layanan kueri interaktif yang memudahkan untuk menganalisis data di Amazon S3 menggunakan SQL standar.

- [Loggly](#)
- [Splunk](#)
- [Logika sumo](#)

## Aktifkan log pemeriksaan kesehatan untuk Application Load Balancer

Saat Anda mengaktifkan log pemeriksaan kesehatan untuk penyeimbang beban Anda, Anda harus menentukan nama bucket S3 tempat penyeimbang beban akan menyimpan log. Bucket harus memiliki kebijakan bucket yang memberikan izin Elastic Load Balancing untuk menulis ke bucket.

### Tugas

- [Langkah 1: Buat ember S3](#)
- [Langkah 2: Lampirkan kebijakan ke bucket S3 Anda](#)
- [Langkah 3: Konfigurasi log pemeriksaan kesehatan](#)
- [Langkah 4: Verifikasi izin bucket](#)
- [Pemecahan masalah](#)

### Langkah 1: Buat ember S3

Saat mengaktifkan log pemeriksaan kesehatan, Anda harus menentukan bucket S3 untuk log pemeriksaan kesehatan. Anda dapat menggunakan bucket yang sudah ada, atau membuat bucket khusus untuk log pemeriksaan kesehatan. Bucket harus memenuhi persyaratan berikut.

### Persyaratan

- Bucket harus ditempatkan di Wilayah yang sama dengan penyeimbang beban. Bucket dan load balancer dapat dimiliki oleh akun yang berbeda.
- Satu-satunya opsi enkripsi sisi server yang didukung adalah kunci yang dikelola Amazon S3 (SSE-S3). Untuk informasi selengkapnya, lihat [kunci enkripsi terkelola Amazon S3 \(SSE-S3\)](#).

Untuk membuat bucket S3 menggunakan konsol Amazon S3

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih Buat bucket.
3. Pada halaman Create bucket, lakukan hal berikut:

- a. Untuk Bucket name, masukkan nama untuk bucket Anda. Nama ini harus unik di semua nama bucket yang ada di Amazon S3. Di beberapa Wilayah, mungkin ada pembatasan tambahan pada nama bucket. Untuk informasi selengkapnya, lihat [Pembatasan dan batasan bucket](#) di Panduan Pengguna Amazon S3.
- b. Untuk AWS Region, pilih Wilayah tempat Anda membuat penyeimbang beban.
- c. Untuk enkripsi Default, pilih kunci yang dikelola Amazon S3 (SSE-S3).
- d. Pilih Buat bucket.

## Langkah 2: Lampirkan kebijakan ke bucket S3 Anda

Bucket S3 Anda harus memiliki kebijakan bucket yang memberikan izin Elastic Load Balancing untuk menulis log pemeriksaan kesehatan ke bucket. Kebijakan bucket adalah kumpulan pernyataan JSON yang ditulis dalam bahasa kebijakan akses untuk menentukan izin akses untuk bucket Anda. Setiap pernyataan mencakup informasi tentang satu izin dan berisi serangkaian elemen.

Jika Anda menggunakan bucket yang sudah memiliki kebijakan terlampir, Anda dapat menambahkan pernyataan untuk log pemeriksaan kesehatan Elastic Load Balancing ke kebijakan. Jika Anda melakukannya, sebaiknya Anda mengevaluasi kumpulan izin yang dihasilkan untuk memastikan bahwa izin tersebut sesuai untuk pengguna yang memerlukan akses ke bucket untuk log pemeriksaan kesehatan.

### Kebijakan bucket

Kebijakan ini memberikan izin ke layanan pengiriman log yang ditentukan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
    }
  ]
}
```

Untuk `Resource`, masukkan ARN lokasi untuk log akses, menggunakan format yang ditunjukkan dalam kebijakan contoh. Selalu sertakan ID akun dengan penyeimbang beban di jalur sumber daya bucket S3 ARN. Ini memastikan bahwa hanya penyeimbang beban dari akun tertentu yang dapat menulis log akses ke bucket S3.

[ARN yang Anda tentukan tergantung pada apakah Anda berencana untuk menyertakan awalan saat Anda mengaktifkan log akses di langkah 3.](#)

Contoh S3 bucket ARN dengan awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket` dan awalannya adalah `logging-prefix`

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

AWS GovCloud (US)— Contoh berikut menggunakan sintaks ARN untuk AWS GovCloud (US) Regions

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Contoh S3 bucket ARN tanpa awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket`. Tidak ada bagian awalan di ember S3 ARN.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

AWS GovCloud (US)— Contoh berikut menggunakan sintaks ARN untuk AWS GovCloud (US) Regions

```
arn:aws-us-gov:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

### Kebijakan bucket lama

Sebelumnya, untuk Wilayah yang tersedia sebelum Agustus 2022, kami mewajibkan kebijakan yang memberikan izin ke akun Elastic Load Balancing yang khusus untuk Wilayah. Kebijakan lama ini masih didukung, namun sebaiknya Anda menggantinya dengan kebijakan yang lebih baru di atas. Jika Anda lebih suka tetap menggunakan kebijakan lama, yang tidak ditampilkan di sini, Anda bisa.

Sebagai referensi, berikut adalah akun Elastic Load Balancing yang akan ditentukan `Principal` dalam kebijakan lama. IDs Perhatikan bahwa Wilayah yang tidak ada dalam daftar ini tidak mendukung kebijakan lama.

- AS Timur (Virginia N.) — 127311923021
- AS Timur (Ohio) — 033677994240
- AS Barat (California N.) — 027434742980
- AS Barat (Oregon) — 797873946194
- Afrika (Cape Town) — 098369216593
- Asia Pasifik (Hong Kong) — 754344448648
- Asia Pasifik (Jakarta) - 589379963580
- Asia Pasifik (Mumbai) — 718504428378
- Asia Pasifik (Osaka) — 383597477331
- Asia Pasifik (Seoul) — 600734575887
- Asia Pasifik (Singapura) — 114774131450
- Asia Pasifik (Sydney) — 783225319266
- Asia Pasifik (Tokyo) — 582318560864
- Kanada (Tengah) — 985666609251
- Eropa (Frankfurt am Main) — 054676820928
- Eropa (Irlandia) — 156460612806
- Eropa (London) — 652711504416
- Eropa (Milan) — 635631232127
- Eropa (Paris) — 009996457667
- Eropa (Stockholm) — 897822967062
- Timur Tengah (Bahrain) — 076674570225
- Amerika Selatan (São Paulo) — 507241528517
- AWS GovCloud (AS-Timur) — 190560391635
- AWS GovCloud (AS-Barat) — 048591011584

## Zona Outposts

Kebijakan berikut memberikan izin ke layanan pengiriman log yang ditentukan. Gunakan kebijakan ini untuk menyeimbangkan beban di Zona Outposts.

```
{  
  "Effect": "Allow",  
  "Principal": {
```

```

    "Service": "logdelivery.elb.amazonaws.com"
  },
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/AWSLogs/123456789012/*"
  "Condition": {
    "StringEquals": {
      "s3:x-amz-acl": "bucket-owner-full-control"
    }
  }
}

```

Untuk `Resource`, masukkan ARN lokasi untuk log akses. Selalu sertakan ID akun dengan penyeimbang beban di jalur sumber daya bucket S3 ARN. Ini memastikan bahwa hanya penyeimbang beban dari akun tertentu yang dapat menulis log akses ke bucket S3.

[ARN yang Anda tentukan tergantung pada apakah Anda berencana untuk menyertakan awalan saat Anda mengaktifkan log akses di langkah 3.](#)

Contoh S3 bucket ARN dengan awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket` dan awalnya adalah `logging-prefix`

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/logging-prefix/AWSLogs/123456789012/*
```

Contoh S3 bucket ARN tanpa awalan

Nama bucket S3 adalah `amzn-s3-demo-logging-bucket`. Tidak ada bagian awalan di ember S3 ARN.

```
arn:aws:s3:::amzn-s3-demo-logging-bucket/AWSLogs/123456789012/*
```

Praktik terbaik keamanan

Untuk meningkatkan keamanan, gunakan bucket ARNs S3 yang tepat.

- Gunakan jalur sumber daya lengkap, bukan hanya ARN bucket S3.
- Sertakan bagian ID akun dari bucket S3 ARN.
- Jangan gunakan wildcard (\*) di bagian ID akun ARN bucket S3.

Setelah membuat kebijakan bucket, gunakan antarmuka Amazon S3, seperti konsol Amazon S3 AWS CLI atau perintah, untuk melampirkan kebijakan bucket ke bucket S3.

## Console

Untuk melampirkan kebijakan bucket Anda ke bucket S3

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih nama bucket untuk membuka halaman detailnya.
3. Pilih Izin lalu pilih Kebijakan Bucket, Edit.
4. Perbarui kebijakan bucket untuk memberikan izin yang diperlukan.
5. Pilih Simpan perubahan.

## AWS CLI

Untuk melampirkan kebijakan bucket Anda ke bucket S3

Gunakan perintah [put-bucket-policy](#). Dalam contoh ini, kebijakan bucket disimpan ke file.json yang ditentukan.

```
aws s3api put-bucket-policy \  
  --bucket amzn-s3-demo-bucket \  
  --policy file://access-log-policy.json
```

## Langkah 3: Konfigurasi log pemeriksaan kesehatan

Gunakan prosedur berikut untuk mengonfigurasi log pemeriksaan kesehatan untuk menangkap dan mengirimkan file log ke bucket S3 Anda.

### Persyaratan

Bucket harus memenuhi persyaratan yang dijelaskan pada [langkah 1](#), dan Anda harus melampirkan kebijakan bucket seperti yang dijelaskan pada [langkah 2](#). Jika Anda menentukan awalan, itu tidak harus menyertakan string "AWSLogs".

Untuk mengelola bucket S3 untuk log pemeriksaan kesehatan Anda

Pastikan untuk menonaktifkan log pemeriksaan kesehatan sebelum menghapus bucket yang telah dikonfigurasi untuk log pemeriksaan kesehatan. Jika tidak, jika ada bucket baru dengan nama yang sama dan kebijakan bucket yang diperlukan tetapi dibuat dalam bucket Akun AWS yang tidak Anda miliki, Elastic Load Balancing dapat menulis log pemeriksaan kesehatan untuk penyeimbang beban Anda ke bucket baru ini.

## Console

Untuk mengaktifkan log pemeriksaan kesehatan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Di panel navigasi, pilih Load Balancers.
3. Pilih nama penyeimbang beban Anda untuk membuka halaman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Untuk Pemantauan, nyalakan log Pemeriksaan Kesehatan.
6. Untuk URI S3, masukkan URI S3 untuk file log Anda. URI yang Anda tentukan bergantung pada apakah Anda menggunakan awalan.
  - URI dengan awalan: `s3://bucket-name/prefix`
  - URI tanpa awalan: `s3://bucket-name`
7. Pilih Simpan perubahan.

## AWS CLI

Untuk mengaktifkan log pemeriksaan kesehatan

Gunakan [modify-load-balancer-attributes](#) perintah dengan atribut terkait.

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes \  
    Key=health_check_logs.s3.enabled,Value=true \  
    Key=health_check_logs.s3.bucket,Value=amzn-s3-demo-logging-bucket \  
    Key=health_check_logs.s3.prefix,Value=logging-prefix
```

## CloudFormation

Untuk mengaktifkan log pemeriksaan kesehatan

Perbarui [AWS::ElasticLoadBalancingV2::LoadBalancer](#) sumber daya untuk menyertakan atribut terkait.

```
Resources:  
  myLoadBalancer:  
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'  
    Properties:
```

```
Name: my-alb
Type: application
Scheme: internal
Subnets:
  - !Ref subnet-AZ1
  - !Ref subnet-AZ2
SecurityGroups:
  - !Ref mySecurityGroup
LoadBalancerAttributes:
  - Key: "health_check_logs.s3.enabled"
    Value: "true"
  - Key: "health_check_logs.s3.bucket"
    Value: "amzn-s3-demo-logging-bucket"
  - Key: "health_check_logs.s3.prefix"
    Value: "logging-prefix"
```

## Langkah 4: Verifikasi izin bucket

Setelah log pemeriksaan kesehatan diaktifkan untuk penyeimbang beban Anda, Elastic Load Balancing memvalidasi bucket S3 dan membuat file pengujian untuk memastikan bahwa kebijakan bucket menentukan izin yang diperlukan. Anda dapat menggunakan konsol Amazon S3 untuk memverifikasi bahwa file uji dibuat. File pengujian bukan file log pemeriksaan kesehatan yang sebenarnya; itu tidak berisi catatan contoh.

Untuk memverifikasi bahwa Elastic Load Balancing membuat file uji di bucket S3 Anda

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih nama bucket yang Anda tentukan untuk log pemeriksaan kesehatan.
3. Arahkan ke file pengujian, ELBHealthCheckLogTestFile. Lokasi tergantung pada apakah Anda menggunakan awalan.
  - Lokasi dengan awalan: *amzn-s3-demo-logging-bucket//prefix/*  
AWSLogs/*123456789012*ELBHealthCheckLogTestFile
  - Lokasi tanpa awalan: *amzn-s3-demo-logging-bucket//*  
AWSLogs/*123456789012*ELBHealthCheckLogTestFile

## Pemecahan masalah

Jika Anda menerima kesalahan akses ditolak, berikut ini adalah kemungkinan penyebabnya:

- Kebijakan bucket tidak memberikan izin Elastic Load Balancing untuk menulis log pemeriksaan kesehatan ke bucket. Verifikasi bahwa Anda menggunakan kebijakan bucket yang benar untuk Wilayah tersebut. Verifikasi bahwa ARN sumber daya menggunakan nama bucket yang sama dengan yang Anda tentukan saat mengaktifkan log pemeriksaan kesehatan. Verifikasi bahwa ARN sumber daya tidak menyertakan awalan jika Anda tidak menentukan awalan saat Anda mengaktifkan log pemeriksaan kesehatan.
- Bucket menggunakan opsi enkripsi sisi server yang tidak didukung. Bucket harus menggunakan kunci yang dikelola Amazon S3 (SSE-S3).

## Nonaktifkan log pemeriksaan kesehatan untuk Application Load Balancer

Anda dapat menonaktifkan log pemeriksaan kesehatan untuk penyeimbang beban Anda kapan saja. Setelah Anda menonaktifkan log pemeriksaan kesehatan, log pemeriksaan kesehatan Anda tetap berada di ember S3 hingga Anda menghapusnya. Untuk informasi selengkapnya, lihat [Membuat, mengonfigurasi, dan bekerja dengan bucket](#) di Panduan Pengguna Amazon S3.

### Console

Untuk menonaktifkan log pemeriksaan kesehatan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>
2. Di panel navigasi, pilih Load Balancers.
3. Pilih nama penyeimbang beban Anda untuk membuka halaman detailnya.
4. Pada tab Atribut, pilih Edit.
5. Untuk Pemantauan, matikan log cek Kesehatan.
6. Pilih Simpan perubahan.

### AWS CLI

Untuk menonaktifkan log pemeriksaan kesehatan

Gunakan perintah [modify-load-balancer-attributes](#).

```
aws elbv2 modify-load-balancer-attributes \  
  --load-balancer-arn load-balancer-arn \  
  --attributes Key=health_check_logs.s3.enabled,Value=false
```

# Pelacakan permintaan untuk Application Load Balancer Anda

Saat penyeimbang beban menerima permintaan dari klien, hal tersebut menambahkan atau memperbarui header X-Amz-Trace-Id sebelum mengirim permintaan ke target. Layanan atau aplikasi apa pun antara penyeimbang beban dan target juga dapat menambahkan atau memperbarui header ini.

Anda dapat menggunakan pelacakan permintaan untuk melacak permintaan HTTP dari klien ke target atau layanan lainnya. Jika Anda mengaktifkan log akses, isi header X-Amz-Trace-Id dicatat. Untuk informasi selengkapnya, lihat [Log akses untuk Application Load Balancer Anda](#).

## Sintaksis

Header X-Amz-Trace-Id berisi bidang dengan format berikut:

```
Field=version-time-id
```

### Bidang

Nama bidang. Nilai yang didukung adalah Root dan Self.

Aplikasi dapat menambahkan bidang arbitrer untuk tujuannya sendiri. Penyeimbang beban mempertahankan bidang ini, tetapi tidak menggunakannya.

### versi

Nomor versi. Nilai ini adalah 1.

### Waktu

Jangka waktu dalam detik. Nilai ini panjangnya 8 digit heksadesimal.

### id

Pengidentifikasi jejak. Nilai ini adalah 24 digit heksadesimal.

### Contoh

Jika header X-Amz-Trace-Id tidak ada pada permintaan masuk, penyeimbang beban menghasilkan header dengan bidang Root dan meneruskan permintaan. Misalnya:

```
X-Amzn-Trace-Id: Root=1-67891233-abcdef012345678912345678
```

Jika header `X-Amz-Trace-Id` ada dan memiliki bidang `Root`, penyeimbang beban menyisipkan bidang `Self` dan meneruskan permintaan. Misalnya:

```
X-Amzn-Trace-Id: Self=1-67891233-12456789abcdef012345678;Root=1-67891233-abcdef012345678912345678
```

Jika aplikasi menambahkan header dengan bidang `Root` dan bidang kustom, penyeimbang beban mempertahankan kedua bidang, menyisipkan bidang `Self`, dan meneruskan permintaan:

```
X-Amzn-Trace-Id: Self=1-67891233-12456789abcdef012345678;Root=1-67891233-abcdef012345678912345678;CalledFrom=app
```

Jika header `X-Amz-Trace-Id` ada dan memiliki bidang `Self`, penyeimbang beban memperbarui nilai bidang `Self`.

## Batasan

- Penyeimbang beban memperbarui header saat menerima permintaan masuk, bukan saat menerima respons.
- Jika header HTTP lebih besar dari 7 KB, penyeimbang beban menulis ulang header `X-Amz-Trace-Id` dengan bidang `Root`.
- Dengan WebSockets, Anda dapat melacak hanya sampai permintaan peningkatan berhasil.

# Memecahkan masalah Application Load Balancer

Informasi berikut dapat membantu Anda memecahkan masalah pada Application Load Balancer.

## Masalah

- [Target terdaftar tidak dalam layanan](#)
- [Klien tidak dapat menyambung ke Load Balancer yang menghadap internet](#)
- [Permintaan yang dikirim ke domain kustom tidak diterima oleh penyeimbang beban](#)
- [Permintaan HTTPS yang dikirim ke penyeimbang beban mengembalikan “NET: :ERR\\_CERT\\_COMMON\\_NAME\\_INVALID”](#)
- [Load balancer menunjukkan peningkatan waktu pemrosesan](#)
- [Load Balancer mengirimkan kode respon 000](#)
- [Load Balancer menghasilkan kesalahan HTTP](#)
- [Target menghasilkan kesalahan HTTP](#)
- [AWS Certificate Manager Sertifikat tidak tersedia untuk digunakan](#)
- [Header Multi-Line tidak didukung](#)
- [Memecahkan masalah target yang tidak sehat menggunakan peta sumber daya](#)
- [Memecahkan masalah pengoptimal target](#)

## Target terdaftar tidak dalam layanan

Jika target memakan waktu lebih lama dari yang diharapkan untuk masuk ke status InService, mungkin target akan gagal dalam pemeriksaan kesehatan. Target Anda tidak akan masuk dalam pelayanan sampai melewati satu pemeriksaan kesehatan. Untuk informasi selengkapnya, lihat [Pemeriksaan kondisi untuk grup target Penyeimbang Beban Aplikasi](#).

Verifikasi bahwa instans Anda gagal dalam pemeriksaan kesehatan dan kemudian periksa masalah berikut:

### Grup keamanan tidak mengizinkan lalu lintas

Grup keamanan yang terkait dengan instans harus mengizinkan lalu lintas dari Load Balancer menggunakan port pemeriksaan kesehatan dan protokol pemeriksaan kesehatan. Anda dapat menambahkan aturan ke grup keamanan instans untuk mengizinkan semua lalu lintas dari

grup keamanan Load Balancer. Selain itu, grup keamanan untuk Load Balancer Anda harus mengizinkan lalu lintas ke instance.

Daftar kontrol akses jaringan (ACL) tidak mengizinkan lalu lintas

ACL jaringan yang terkait dengan subnet untuk instans Anda harus memungkinkan lalu lintas masuk pada port pemeriksaan kesehatan dan lalu lintas keluar pada port fana (1024-65535). ACL jaringan yang terkait dengan subnet untuk node Load Balancer Anda harus memungkinkan lalu lintas masuk pada port fana dan lalu lintas keluar pada pemeriksaan kesehatan dan port fana.

Jalur ping tidak ada

Buat halaman target untuk pemeriksaan kesehatan dan tentukan jalurnya sebagai jalur ping.

Waktu koneksi habis

Pertama, verifikasi bahwa Anda dapat terhubung ke target langsung dari dalam jaringan menggunakan alamat IP pribadi target dan protokol pemeriksaan kesehatan. Jika Anda tidak dapat terhubung, periksa apakah instans terlalu banyak digunakan, dan tambahkan lebih banyak target ke grup target Anda jika terlalu sibuk untuk merespons. Jika Anda dapat terhubung, mungkin halaman target tidak merespons sebelum periode batas waktu pemeriksaan kesehatan. Pilih halaman target yang lebih sederhana untuk pemeriksaan kesehatan atau sesuaikan pengaturan pemeriksaan kesehatan.

Target tidak mengembalikan kode respon yang sukses

Secara default, kode sukses adalah 200, tetapi secara opsional Anda dapat menentukan kode keberhasilan tambahan ketika Anda mengkonfigurasi pemeriksaan kesehatan. Konfirmasikan kode sukses yang diharapkan Load Balancer dan bahwa aplikasi Anda telah dikonfigurasi untuk menjawab kode ini berhasil.

Kode respons target rusak atau ada kesalahan saat menghubungkan ke target

Verifikasi bahwa aplikasi Anda merespons permintaan pemeriksaan kesehatan Load Balancer. Beberapa aplikasi memerlukan konfigurasi tambahan untuk menanggapi pemeriksaan kesehatan, seperti konfigurasi host virtual untuk menanggapi header host HTTP yang dikirim oleh Load Balancer. Nilai header host berisi alamat IP pribadi target, diikuti oleh port pemeriksaan kesehatan saat tidak menggunakan port default. Jika target menggunakan port pemeriksaan kesehatan default, nilai header host hanya berisi alamat IP pribadi target. Misalnya, jika alamat IP pribadi target Anda `10.0.0.10` dan port pemeriksaan kesehatannya `8080`, header Host HTTP yang dikirim oleh penyeimbang beban dalam pemeriksaan kesehatan adalah `Host : 10.0.0.10:8080`. Jika alamat IP pribadi target Anda `10.0.0.10` dan port pemeriksaan

kesehatannya adalah 80 header Host HTTP yang dikirim oleh penyeimbang beban dalam pemeriksaan kesehatan adalah `Host: 10.0.0.10`. Konfigurasi host virtual untuk menanggapi host tersebut, atau konfigurasi default, mungkin diperlukan agar berhasil memeriksa kesehatan aplikasi Anda. Permintaan pemeriksaan kesehatan memiliki atribut berikut: `User-Agent` diatur ke `ELB-HealthChecker/2.0`, pemangkas garis untuk bidang pesan-header adalah urutan CRLF, dan header berakhir pada baris kosong pertama diikuti oleh CRLF.

## Klien tidak dapat menyambung ke Load Balancer yang menghadap internet

Jika Load Balancer tidak merespons permintaan, periksa masalah berikut ini:

Load Balancer yang menghadap internet Anda terpasang ke subnet pribadi

Anda harus menentukan subnet publik untuk Load Balancer Anda. Subnet publik memiliki rute ke Internet Gateway untuk cloud privat virtual (VPC) Anda.

Grup keamanan atau jaringan ACL tidak mengizinkan lalu lintas

Grup keamanan untuk penyeimbang beban dan jaringan apa pun ACLs untuk subnet penyeimbang beban harus memungkinkan lalu lintas masuk dari klien dan lalu lintas keluar ke klien di port pendengar.

## Permintaan yang dikirim ke domain kustom tidak diterima oleh penyeimbang beban

Jika penyeimbang beban tidak menerima permintaan yang dikirim ke domain kustom, periksa masalah berikut:

Nama domain kustom tidak diselesaikan ke alamat IP penyeimbang beban

- Konfirmasikan alamat IP apa yang diselesaikan oleh nama domain khusus untuk menggunakan antarmuka baris perintah.
  - Linux, macOS, atau Unix — Anda dapat menggunakan `dig` perintah di dalam Terminal.  
Mantan. `dig example.com`
  - Windows — Anda dapat menggunakan `nslookup` perintah dalam Command Prompt.  
Mantan. `nslookup example.com`

- Konfirmasikan alamat IP apa yang diselesaikan oleh nama DNS penyeimbang beban untuk menggunakan antarmuka baris perintah.
- Bandingkan hasil dari dua output. Alamat IP harus cocok.

Jika menggunakan Route 53 untuk meng-host domain kustom Anda, lihat [Domain saya tidak tersedia di internet](#) di Panduan Pengembang Amazon Route 53.

## Permintaan HTTPS yang dikirim ke penyeimbang beban mengembalikan “NET: :ERR\_CERT\_COMMON\_NAME\_INVALID”

Jika permintaan HTTPS diterima NET : :ERR\_CERT\_COMMON\_NAME\_INVALID dari penyeimbang beban, periksa kemungkinan penyebab berikut:

- Nama domain yang digunakan dalam permintaan HTTPS tidak cocok dengan nama alternatif yang ditentukan dalam sertifikat ACM terkait pendengar.
- Nama DNS default load balancers sedang digunakan. Nama DNS default tidak dapat digunakan untuk membuat permintaan HTTPS karena sertifikat publik tidak dapat diminta untuk \*.amazonaws.com domain.

## Load balancer menunjukkan peningkatan waktu pemrosesan

Penyeimbang beban menghitung waktu pemrosesan secara berbeda berdasarkan konfigurasi.

- Jika AWS WAF dikaitkan dengan Application Load Balancer Anda dan klien mengirimkan permintaan HTTP POST, waktu untuk mengirim data untuk permintaan POST tercermin dalam `request_processing_time` bidang di log akses penyeimbang beban. Perilaku ini diharapkan untuk permintaan HTTP POST.
- Jika AWS WAF tidak terkait dengan Application Load Balancer Anda dan klien mengirimkan permintaan HTTP POST, waktu untuk mengirim data untuk permintaan POST tercermin dalam `target_processing_time` bidang di log akses penyeimbang beban. Perilaku ini diharapkan untuk permintaan HTTP POST.

## Load Balancer mengirimkan kode respon 000

Dengan koneksi HTTP/2, jika jumlah permintaan yang dilayani melalui satu koneksi melebihi 10.000, penyeimbang beban mengirimkan bingkai GOAWAY dan menutup koneksi dengan TCP FIN.

## Load Balancer menghasilkan kesalahan HTTP

Kesalahan HTTP berikut dihasilkan oleh Load Balancer. Load Balancer mengirimkan kode HTTP untuk klien, menyimpan permintaan ke log akses, dan menambahkan metrik HTTPCode\_ELB\_4XX\_Count atau HTTPCode\_ELB\_5XX\_Count.

### Kesalahan

- [HTTP 400: Permintaan buruk](#)
- [HTTP 401: Tidak sah](#)
- [HTTP 403: Terlarang](#)
- [HTTP 405: Metode tidak diperbolehkan](#)
- [HTTP 408: Waktu habis permintaan](#)
- [HTTP 413: Muatan terlalu besar](#)
- [HTTP 414: URI terlalu panjang](#)
- [HTTP 460](#)
- [HTTP 463](#)
- [HTTP 464](#)
- [HTTP 500: Kesalahan peladen internal](#)
- [HTTP 501: Tidak diimplementasikan](#)
- [HTTP 502: Gateway buruk](#)
- [503 Layanan Tidak Tersedia](#)
- [HTTP 504: Waktu habis gateway](#)
- [HTTP 505: Versi tidak didukung](#)
- [HTTP 507: Penyimpanan Tidak Cukup](#)
- [HTTP 561: Tidak sah](#)
- [HTTP 562: Permintaan JWKS Gagal](#)

## HTTP 400: Permintaan buruk

Kemungkinan penyebab :

- Klien mengirim permintaan cacat yang tidak memenuhi spesifikasi HTTP.
- Header permintaan melebihi 16 K per baris permintaan, 16 K per header tunggal, atau 64 K untuk seluruh header permintaan.
- Klien menutup koneksi sebelum mengirim badan permintaan lengkap.

## HTTP 401: Tidak sah

Anda mengkonfigurasi aturan pendengar untuk mengautentikasi pengguna, tetapi salah satu dari yang berikut ini benar:

- Anda mengkonfigurasi `OnUnauthenticatedRequest` untuk menolak pengguna yang tidak terautentikasi atau IdP ditolak akses.
- Ukuran klaim yang dikembalikan oleh IdP melebihi ukuran maksimum yang didukung oleh Load Balancer.
- Klien mengirimkan permintaan HTTP/1.0 tanpa host header, dan Load Balancer tidak dapat menghasilkan URL pengalihan.
- Lingkup yang diminta tidak mengembalikan ID token.
- Anda tidak menyelesaikan proses login sebelum batas waktu login klien berakhir. Untuk informasi selengkapnya lihat, [batas waktu login Klien](#).
- Otentikasi JWT gagal karena salah satu alasan berikut:
  - Permintaan tidak memiliki header Otorisasi. (`JWTHeaderNotPresent`)
  - Format token dalam permintaan tidak valid. Ini dapat terjadi ketika:
    - Token salah bentuk atau tidak ada bagian wajib (header, payload, atau tanda tangan)
    - Header tidak memiliki awalan “Pembawa”
    - Header berisi jenis otentikasi yang berbeda (misalnya, “Dasar”)
    - Header otorisasi ada tetapi token tidak ada
    - Beberapa token hadir dalam permintaan (`JWTRequestFormatInvalid`)
  - Validasi tanda tangan token gagal. Ini dapat terjadi ketika:
    - Tanda tangan tidak cocok
    - Kunci publik tidak valid atau tidak dapat dikonversi ke kunci decoding

- Ukuran kunci publik tidak 2K
- Token ditandatangani dengan algoritma yang tidak didukung
- KID dalam token tidak ada di titik akhir JWKS () JWTSignature ValidationFailed
- JWT tidak memiliki klaim yang diperlukan untuk validasi. (JWTClaimNotPresent)
- Format nilai klaim di JWT tidak cocok dengan format konfigurasi yang ditentukan. (JWTClaimFormatInvalid)

## HTTP 403: Terlarang

Anda mengonfigurasi daftar kontrol akses AWS WAF web (web ACL) untuk memantau permintaan ke Application Load Balancer Anda dan memblokir permintaan.

## HTTP 405: Metode tidak diperbolehkan

Klien menggunakan metode TRACE, yang tidak didukung oleh Application Load Balancer.

## HTTP 408: Waktu habis permintaan

Klien tidak mengirim data sebelum periode waktu habis siaga kedaluwarsa. Mengirim TCP tetap-hidup tidak mencegah waktu habis ini. Kirim setidaknya 1 byte data sebelum setiap periode waktu habis siaga berlalu. Meningkatkan panjang periode waktu habis siaga sesuai kebutuhan.

## HTTP 413: Muatan terlalu besar

Kemungkinan penyebab:

- Target adalah fungsi Lambda dan isi permintaan melebihi 1 MB.
- Header permintaan melebihi 16 K per baris permintaan, 16 K per header tunggal, atau 64 K untuk seluruh header permintaan.

## HTTP 414: URI terlalu panjang

Permintaan URL atau parameter kueri string terlalu besar.

## HTTP 460

Load Balancer menerima permintaan dari klien, namun klien menutup koneksi dengan Load Balancer sebelum periode timeout siaga berlalu.

Periksa apakah periode waktu habis klien lebih besar daripada periode waktu habis siaga untuk Load Balancer. Pastikan bahwa target Anda memberikan respons ke klien sebelum periode waktu habis klien berlalu, atau meningkatkan periode waktu habis klien untuk mencocokkan batas waktu siaga Load Balancer, jika klien mendukung ini.

## HTTP 463

Load balancer menerima permintaan header X-Forwarded-For dengan terlalu banyak alamat IP. Batas atas untuk alamat IP adalah 30.

## HTTP 464

Load Balancer menerima protokol permintaan masuk yang tidak kompatibel dengan konfigurasi versi protokol grup target.

Kemungkinan penyebab :

- Protokol permintaan adalah HTTP/1.1, sedangkan versi protokol kelompok target adalah gRPC atau HTTP/2.
- Protokol permintaan adalah gRPC, sedangkan versi protokol kelompok target adalah HTTP/1.1.
- Protokol permintaan adalah HTTP/2 dan permintaan tidak POST, sementara versi protokol kelompok target adalah gRPC.

## HTTP 500: Kesalahan peladen internal

Kemungkinan penyebab:

- Anda mengkonfigurasi daftar kontrol akses AWS WAF web (web ACL) dan ada kesalahan dalam mengeksekusi aturan ACL web.
- Load Balancer tidak dapat berkomunikasi dengan IDP tanda akhir atau IDP pengguna info akhir.
  - Verifikasi bahwa DNS IDP dapat diselesaikan secara publik.
  - Verifikasi bahwa grup keamanan untuk penyeimbang beban Anda dan jaringan ACLs untuk VPC Anda mengizinkan akses keluar ke titik akhir ini.
  - Verifikasi bahwa VPC Anda memiliki akses internet. Jika Anda memiliki Load Balancer menghadap internal, gunakan gateway NAT untuk mengaktifkan akses internet.
- Klaim pengguna yang diterima dari IDP berukuran lebih besar dari 11KB.

- Titik akhir token iDP atau titik akhir info pengguna iDP membutuhkan waktu lebih dari 5 detik untuk merespons.
- Penyeimbang beban tidak dapat berkomunikasi dengan titik akhir JWKS, atau titik akhir JWKS tidak merespons dalam 5 detik.
- Ukuran respons yang dikembalikan oleh titik akhir JWKS melebihi 150KB atau jumlah kunci yang dikembalikan oleh titik akhir JWKS melebihi 10
- Grup target mengaktifkan pengoptimal target dan agen mengalami kesalahan yang tidak terduga. Lihat [the section called “Memecahkan masalah pengoptimal target”](#).

## HTTP 501: Tidak diimplementasikan

Kemungkinan penyebab:

- Load Balancer menerima header Transfer-Encoding nilai yang tidak didukung. Nilai-nilai yang didukung untuk Transfer-Encoding adalah chunked dan identity. Sebagai alternatif, Anda dapat menggunakan Header-Encoding-Konten.
- Permintaan websocket dirutekan ke grup target dengan pengoptimal target diaktifkan.

## HTTP 502: Gateway buruk

Kemungkinan penyebab :

- Load Balancer menerima TCP RST dari target saat mencoba membuat sambungan.
- Load Balancer menerima respons tak terduga dari target, seperti “ICMP tujuan tidak terjangkau (Host tidak terjangkau)”, ketika mencoba untuk membuat sambungan. Periksa apakah lalu lintas diperbolehkan dari subnet Load Balancer ke target pada port target.
- Target menutup koneksi dengan TCP RST atau TCP FIN sementara Load Balancer memiliki permintaan yang luar biasa ke target. Periksa apakah durasi tetap-menyala target lebih pendek dari nilai batas waktu siaga Load Balancer.
- Respon target rusak atau berisi header HTTP yang tidak valid.
- Header respons target melebihi 32 K untuk seluruh header respons.
- Periode penundaan deregistration berlalu untuk permintaan yang ditangani oleh target yang dibatalkan. Tingkatkan masa tunda sehingga operasi yang panjang bisa selesai.
- Target adalah fungsi Lambda dan isi respon melebihi 1 MB.

- Target adalah fungsi Lambda yang tidak merespon sebelum waktu habis yang dikonfigurasi tercapai.
- Target adalah fungsi Lambda yang mengembalikan kesalahan atau fungsi itu dicekik oleh layanan Lambda.
- Penyeimbang beban mengalami kesalahan jabat tangan SSL saat menghubungkan ke target.

Untuk informasi selengkapnya lihat [Bagaimana cara memecahkan masalah error Application Load Balancer HTTP 502 di Pusat Pengetahuan](#) Dukungan. AWS

## 503 Layanan Tidak Tersedia

Kemungkinan penyebab:

- Kelompok sasaran untuk penyeimbang beban tidak memiliki target terdaftar, atau semua target yang terdaftar berada dalam suatu unused keadaan.
- Permintaan dialihkan ke grup target dengan pengoptimal target diaktifkan, dan ditolak karena tidak ada target yang siap menerima permintaan. Lihat [the section called “Memecahkan masalah pengoptimal target”](#).

## HTTP 504: Waktu habis gateway

Kemungkinan penyebab :

- Load Balancer gagal untuk membuat sambungan ke target sebelum batas waktu sambungan berakhir (10 detik).
- Load Balancer membuat sambungan ke target tetapi target tidak merespons sebelum periode waktu habis siaga berlalu.
- ACL jaringan untuk subnet tidak memungkinkan lalu lintas dari target ke simpul Load Balancer pada port fana (1024-65535).
- Target mengembalikan header konten-panjang yang lebih besar dari isi entitas. Load Balancer kehabisan waktu menunggu byte hilang.
- Target adalah fungsi Lambda dan layanan Lambda tidak merespons sebelum batas waktu koneksi berakhir.
- Penyeimbang beban mengalami batas waktu jabat tangan SSL (10 detik) saat menghubungkan ke target.

## HTTP 505: Versi tidak didukung

Load Balancer menerima permintaan versi HTTP yang tak terduga. Misalnya, Load Balancer membuat koneksi HTTP/1 tetapi menerima permintaan HTTP/2.

## HTTP 507: Penyimpanan Tidak Cukup

URL redirect terlalu panjang.

## HTTP 561: Tidak sah

Anda mengkonfigurasi aturan pendengar untuk mengautentikasi pengguna, tetapi IdP mengembalikan kode galat saat mengautentikasi pengguna. Periksa log akses Anda untuk [kode alasan kesalahan](#) terkait.

## HTTP 562: Permintaan JWKS Gagal

Penyeimbang beban gagal menerima respons yang berhasil dan valid dari titik akhir JWKS (JSON Web Key Set). Respons yang berhasil harus memiliki kode status dalam kisaran 200-299, tetapi kode status yang berbeda diterima sebagai gantinya. Tanggapan yang valid seharusnya tidak memiliki masalah berikut:

- Format non-JSON
- Karakter tidak valid
- Format JWKS tidak valid
- Atribut JWKS wajib hilang/tidak valid
- Kunci publik memiliki algoritme yang tidak didukung
- kunci publik tidak dapat dikonversi ke kunci decoding
- ukuran kunci publik tidak 2K

## Target menghasilkan kesalahan HTTP

Load Balancer meneruskan respons HTTP yang valid dari target ke klien, termasuk kesalahan HTTP. Kesalahan HTTP yang dihasilkan oleh target dicatat dalam metrik `HTTPCode_Target_4XX_Count` dan `HTTPCode_Target_5XX_Count`.

## AWS Certificate Manager Sertifikat tidak tersedia untuk digunakan

Saat memutuskan untuk menggunakan pendengar HTTPS dengan Application Load Balancer AWS Certificate Manager, Anda harus memvalidasi kepemilikan domain sebelum menerbitkan sertifikat. Jika langkah ini terlewatkan selama penyiapan, sertifikat tetap dalam Pending Validation status, dan tidak tersedia untuk digunakan sampai divalidasi.

- Jika menggunakan validasi email, lihat [Validasi email](#) di AWS Certificate Manager Panduan Pengguna.
- Jika menggunakan validasi DNS, lihat [Validasi DNS](#) di Panduan Pengguna.AWS Certificate Manager

## Header Multi-Line tidak didukung

Application Load Balancers tidak mendukung header multi-line, termasuk header tipe media. message/http Ketika header multi-baris disediakan Application Load Balancer menambahkan karakter titik dua, : "", sebelum meneruskannya ke target.

## Memecahkan masalah target yang tidak sehat menggunakan peta sumber daya

Jika target Application Load Balancer gagal dalam pemeriksaan kesehatan, Anda dapat menggunakan peta sumber daya untuk menemukan target yang tidak sehat dan mengambil tindakan berdasarkan kode alasan kegagalan. Untuk informasi selengkapnya, lihat [Lihat peta sumber daya Application Load Balancer](#).

Peta sumber daya menyediakan dua tampilan: Ikhtisar, dan Peta Target Tidak Sehat. Ikhtisar dipilih secara default dan menampilkan semua sumber daya penyeimbang beban Anda. Memilih tampilan Peta Target Tidak Sehat hanya akan menampilkan target yang tidak sehat di setiap grup target yang terkait dengan Application Load Balancer.

### Note

Anda harus mengaktifkan Tampilkan detail sumber daya untuk melihat ringkasan pemeriksaan kesehatan dan pesan kesalahan untuk semua sumber daya yang berlaku dalam

peta sumber daya. Ketika tidak diaktifkan, Anda harus memilih setiap sumber daya untuk melihat detailnya.

Kolom Grup target menampilkan ringkasan target yang sehat dan tidak sehat untuk setiap kelompok sasaran. Ini dapat membantu menentukan apakah semua target gagal dalam pemeriksaan kesehatan, atau hanya target tertentu yang gagal. Jika semua target dalam kelompok sasaran gagal dalam pemeriksaan kesehatan, periksa konfigurasi kelompok sasaran. Pilih nama grup target untuk membuka halaman detailnya di tab baru.

Kolom TargetId menampilkan targetID dan status pemeriksaan kesehatan saat ini untuk setiap target. Ketika target tidak sehat, kode alasan kegagalan pemeriksaan kesehatan ditampilkan. Ketika satu target gagal dalam pemeriksaan kesehatan, verifikasi target memiliki sumber daya yang cukup dan konfirmasi bahwa aplikasi yang berjalan pada target tersedia. Pilih ID target untuk membuka halaman detailnya di tab baru.

Memilih Ekspor memberi Anda opsi untuk mengekspor tampilan saat ini dari peta sumber daya Application Load Balancer Anda sebagai PDF.

Verifikasi bahwa instans Anda gagal dalam pemeriksaan kesehatan dan kemudian berdasarkan pemeriksaan kode alasan kegagalan untuk masalah berikut:

- Tidak Sehat: Ketidakcocokan Respons HTTP
  - Verifikasi aplikasi yang berjalan pada target mengirimkan respons HTTP yang benar ke permintaan pemeriksaan kesehatan Application Load Balancer.
  - Atau, Anda dapat memperbarui permintaan pemeriksaan kesehatan Application Load Balancer agar sesuai dengan respons dari aplikasi yang berjalan pada target.
- Tidak sehat: Waktu permintaan habis
  - Verifikasi grup keamanan dan daftar kontrol akses jaringan (ACL) yang terkait dengan target Anda dan Application Load Balancer tidak memblokir konektivitas.
  - Pastikan target memiliki sumber daya yang cukup tersedia untuk menerima koneksi dari Application Load Balancer.
  - Verifikasi status aplikasi apa pun yang berjalan pada target.
  - Respons pemeriksaan kesehatan Application Load Balancer dapat dilihat di setiap log aplikasi target. Untuk informasi lebih lanjut, lihat [Health check kode alasan](#).
- Tidak sehat: FailedHealthChecks

- Verifikasi status aplikasi apa pun yang berjalan pada target.
- Verifikasi target mendengarkan lalu lintas di port pemeriksaan kesehatan.

#### Saat menggunakan pendengar HTTPS

Anda memilih kebijakan keamanan yang digunakan untuk koneksi front-end. Kebijakan keamanan yang digunakan untuk koneksi back-end dipilih secara otomatis berdasarkan kebijakan keamanan front-end yang digunakan. Jika salah satu pendengar Anda memiliki:

- Kebijakan TLS pasca-kuantum FIPS - Koneksi backend digunakan `ELBSecurityPolicy-TLS13-1-0-FIPS-PQ-2025-09`
- Kebijakan FIPS - Koneksi backend digunakan `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04`
- Kebijakan TLS pasca-kuantum - Koneksi backend digunakan `ELBSecurityPolicy-TLS13-1-0-PQ-2025-09`
- Kebijakan TLS 1.3 - Koneksi backend digunakan `ELBSecurityPolicy-TLS13-1-0-2021-06`
- Semua kebijakan TLS lainnya menggunakan koneksi backend `ELBSecurityPolicy-2016-08`

Untuk informasi selengkapnya, lihat [Kebijakan keamanan](#).

- Verifikasi target menyediakan sertifikat server dan kunci dalam format yang benar yang ditentukan oleh kebijakan keamanan.
- Verifikasi target mendukung satu atau lebih cipher yang cocok, dan protokol yang disediakan oleh Application Load Balancer untuk membuat jabat tangan TLS.

## Memecahkan masalah pengoptimal target

Untuk pemantauan mendetail, lihat [Metrik pengoptimal target](#)

### Kesalahan Konfigurasi

- `HTTPCode_ELB_502_Count`: Penyeimbang beban menerima TCP RST dari agen ketika mencoba membuat koneksi.

- `HTTPCode_ELB_504_Count`: Penyeimbang beban gagal membuat koneksi ke agen sebelum periode batas waktu idle berlalu.
- `HTTPCode_Target_5XX_Count`: Agen menerima TCP RST dari aplikasi target ketika mencoba untuk membuat koneksi. (Hanya berlaku jika aplikasi target itu sendiri tidak menghasilkan respons kesalahan ini.)

Untuk memperbaiki masalah ini, pastikan bahwa:

- Grup keamanan pada target dikonfigurasi dengan benar.
- Agen berjalan dengan konfigurasi yang diharapkan.
- Aplikasi target sedang berjalan dan mendengarkan pada `TARGET_CONTROL_DESTINATION_ADDRESS` yang dikonfigurasi di agen.

### Kesalahan Layanan Tidak Tersedia () `HTTPCode_ELB_503_Count`

Kesalahan HTTP 503 yang konsisten berarti bahwa tidak ada cukup target yang siap menerima permintaan dari ALB. `TargetControlRequestRejectCount` Metrik mewakili permintaan yang ditolak ini. `TargetControlWorkQueueLength` Metrik akan jatuh mendekati nilai nol. Untuk memperbaiki masalah ini, pertimbangkan:

- Meningkatkan jumlah target
- Menyetel variabel `TARGET_CONTROL_MAX_CONCURRENCY` pada agen ke nilai yang lebih besar.

### Kesalahan pemeriksaan kesehatan

- Jika port pemeriksaan kesehatan sama dengan `TARGET_CONTROL_DATA_ADDRESS`, maka permintaan pemeriksaan kesehatan dari ALB dikirim ke aplikasi target melalui agen. Jika pemeriksaan kesehatan gagal (karena HTTP 502 atau Timeout) lihat bagian Kesalahan Konfigurasi.

## Kuota untuk Application Load Balancer Anda

AWS Akun Anda memiliki kuota default, sebelumnya disebut sebagai batas, untuk setiap layanan. AWS Kecuali dinyatakan lain, setiap kuota bersifat khusus per Wilayah. Anda dapat meminta peningkatan untuk beberapa kuota, dan kuota lainnya tidak dapat ditingkatkan.

Untuk melihat kuota untuk Application Load Balancer Anda, buka [konsol Service Quotas](#). Di panel navigasi, pilih AWS layanan dan pilih Elastic Load Balancing. Anda juga dapat menggunakan perintah [describe-account-limits](#)(AWS CLI) untuk Elastic Load Balancing.

Untuk meminta penambahan kuota, lihat [Meminta penambahan kuota](#) di Panduan Pengguna Service Quotas. Jika kuota belum tersedia di Service Quotas, ajukan permintaan kenaikan kuota [layanan](#).

### Kuota

- [Penyeimbang beban](#)
- [Kelompok-kelompok target](#)
- [Aturan](#)
- [Penyimpanan kepercayaan](#)
- [Sertifikat](#)
- [Header HTTP](#)
- [Unit Kapasitas Load Balancer](#)

## Penyeimbang beban

AWS Akun Anda memiliki kuota berikut yang terkait dengan Application Load Balancers.

Nama	Default	Dapat disesuaikan
Application Load Balancer per Wilayah	50	<a href="#">Ya</a>
Sertifikat per Application Load Balancer (tidak termasuk sertifikat default)	25	<a href="#">Ya</a>
Listener per Application Load Balancer	50	<a href="#">Ya</a>

Nama	Default	Dapat disesuaikan
Grup Target per Tindakan per Application Load Balancer	5	Tidak
Grup Target per Application Load Balancer	100	Tidak
Target per Application Load Balancer	1.000	<a href="#">Ya</a>

## Kelompok-kelompok target

Kuota berikut adalah untuk kelompok sasaran.

Nama	Default	Dapat disesuaikan
Grup Target per Wilayah	3.000 *	<a href="#">Ya</a>
Target per Grup Target per Wilayah (contoh atau alamat IP)	1.000	<a href="#">Ya</a>
Target per Grup Target per Wilayah (fungsi Lambda)	1	Tidak
Load balancer per kelompok target	1	Tidak

\* Kuota ini dibagi oleh Application Load Balancers dan Network Load Balancer.

## Aturan

Kuota berikut adalah untuk aturan.

Nama	Default	Dapat disesuaikan
Aturan per Application Load Balancer (tidak termasuk aturan default)	100	<a href="#">Ya</a>
Nilai Syarat per Aturan	5	Tidak

Nama	Default	Dapat disesuaikan
Wildcard Syarat per Aturan	6	Tidak
Evaluasi kecocokan per aturan	5	Tidak

## Penyimpanan kepercayaan

Kuota berikut adalah untuk toko kepercayaan.

Nama	Default	Dapat disesuaikan
Toko kepercayaan per akun	20	<a href="#">Ya</a>
Jumlah pendengar yang menggunakan mTL dalam mode verifikasi, per penyeimbang beban.	2	Tidak

## Sertifikat

Kuota berikut berlaku untuk sertifikat, termasuk nama sertifikat CA iklan dan daftar pencabutan sertifikat.

Nama	Default	Dapat disesuaikan
Ukuran sertifikat CA	16 KB	Tidak
Sertifikat CA per toko kepercayaan	25	<a href="#">Ya</a>
Ukuran subjek sertifikat CA per toko kepercayaan	10.000	<a href="#">Ya</a>
Kedalaman rantai sertifikat maksimum	4	Tidak
Entri pencabutan per toko kepercayaan	500.000	<a href="#">Ya</a>
Ukuran file daftar pencabutan	50 MB	Tidak
Daftar pencabutan per toko kepercayaan	30	<a href="#">Ya</a>

Nama	Default	Dapat disesuaikan
Ukuran pesan TLS	64 K	Tidak

## Header HTTP

Berikut ini adalah batas ukuran untuk header HTTP.

Nama	Default	Dapat disesuaikan
Baris permintaan	16 K	Tidak
Header tunggal	16 K	Tidak
Seluruh header respons	32 K	Tidak
Seluruh header permintaan	64 K	Tidak

## Unit Kapasitas Load Balancer

Kuota berikut adalah untuk Load Balancer Capacity Units (LCU).

Nama	Default	Dapat disesuaikan
Unit Kapasitas Load Balancer Aplikasi Cadangan (LCUs) per Application Load Balancer	15.000	Ya
Unit Kapasitas Load Balancer Aplikasi Cadangan (LCU) per Wilayah	0	<a href="#">Ya</a>

# Riwayat dokumen untuk Application Load Balancer

Tabel berikut menjelaskan rilis untuk Application Load Balancer.

Perubahan	Deskripsi	Tanggal
<a href="#">Akses validasi token</a>	Rilis ini menambahkan dukungan untuk penyeimbang beban untuk memvalidasi JSON Web Tokens (JWT) yang disediakan oleh klien untuk komunikasi aman service-to-service (S2S) atau (M2M). machine-to-machine	November 21, 2025
<a href="#">Mengubah</a>	Rilis ini menambahkan dukungan untuk mengubah header host dan URLs untuk permintaan yang masuk sebelum penyeimbang beban merutekan lalu lintas ke target.	Oktober 15, 2025
<a href="#">Kebijakan bucket untuk log akses dan log koneksi</a>	Sebelum rilis ini, kebijakan bucket yang Anda gunakan bergantung pada apakah Wilayah tersedia sebelum atau setelah Agustus 2022. Dengan rilis ini, kebijakan bucket yang lebih baru didukung di semua Wilayah. Perhatikan bahwa kebijakan bucket lama masih didukung.	September 10, 2025
<a href="#">Modifikasi header HTTP</a>	Rilis ini menambahkan dukungan untuk modifikasi header HTTP untuk semua kode respons. Sebelumnya,	Februari 28, 2025

fitur ini terbatas pada kode respons 2xx dan 3xx.

### [Reservasi Unit Kapasitas](#)

Rilis ini menambahkan dukungan untuk menetapkan kapasitas minimum penyeimbang beban Anda.

November 20, 2024

### [Peta sumber daya](#)

Rilis ini menambahkan dukungan untuk melihat sumber daya penyeimbang beban dan hubungan Anda dalam format visual.

8 Maret 2024

### [Satu klik WAF](#)

Rilis ini menambahkan dukungan untuk mengonfigurasi perilaku penyeimbang beban Anda jika terintegrasi dengan satu klik. AWS WAF

Februari 6, 2024

### [TLS timbal balik](#)

Rilis ini menambahkan dukungan untuk otentikasi TLS timbal balik.

26 November 2023

### [Bobot Target Otomatis](#)

Rilis ini menambahkan dukungan untuk algoritma bobot target otomatis.

26 November 2023

### [Pengakhiran FIPS 140-3 TLS](#)

Rilis ini menambahkan kebijakan keamanan yang menggunakan modul kriptografi FIPS 140-3 saat mengakhiri koneksi TLS.

20 November 2023

### [Daftarkan target menggunakan IPv6](#)

Rilis ini menambahkan dukungan untuk mendaftarkan instance sebagai target saat ditangani oleh IPv6.

2 Oktober 2023

---

<a href="#">Kebijakan keamanan yang mendukung TLS 1.3</a>	Rilis ini menambahkan dukungan untuk kebijakan keamanan standar TLS 1.3.	22 Maret 2023
<a href="#">Pergeseran zona</a>	Rilis ini menambahkan dukungan untuk merutekan lalu lintas dari satu Availability Zone yang terganggu melalui integrasi dengan Amazon Application Recovery Controller (ARC).	28 November 2022
<a href="#">Matikan penyeimbangan beban lintas zona</a>	Rilis ini menambahkan dukungan untuk mematikan penyeimbangan beban lintas zona.	28 November 2022
<a href="#">Kesehatan kelompok sasaran</a>	Rilis ini menambahkan dukungan untuk mengonfigurasi jumlah minimum atau persentase target yang harus sehat, dan tindakan apa yang dilakukan penyeimbang beban ketika ambang batas tidak terpenuhi.	28 November 2022
<a href="#">Penyeimbangan beban lintas zona</a>	Rilis ini menambahkan dukungan untuk mengonfigurasi penyeimbangan beban lintas zona di tingkat grup target.	17 November 2022
<a href="#">IPv6 kelompok sasaran</a>	Rilis ini menambahkan dukungan untuk mengkonfigurasi grup IPv6 target untuk Application Load Balancers.	23 November 2021

---

<a href="#">IPv6 penyeimbang beban internal</a>	Rilis ini menambahkan dukungan untuk mengkonfigurasi grup IPv6 target untuk Application Load Balancers.	23 November 2021
<a href="#">AWS PrivateLink dan alamat IP statis</a>	Rilis ini menambahkan dukungan untuk menggunakan AWS PrivateLink dan mengekspos alamat IP statis dengan meneruskan lalu lintas langsung dari Network Load Balancers ke Application Load Balancers.	27 September 2021
<a href="#">Pelestarian port klien</a>	Rilis ini menambahkan atribut untuk mempertahankan port sumber yang digunakan klien untuk terhubung ke penyeimbang beban.	29 Juli 2021
<a href="#">Header TLS</a>	Rilis ini menambahkan atribut untuk menunjukkan bahwa header TLS, yang berisi informasi tentang versi TLS yang dinegosiasikan dan cipher suite, ditambahkan ke permintaan klien sebelum mengirimnya ke target.	21 Juli 2021
<a href="#">Sertifikat ACM tambahan</a>	Rilis ini mendukung sertifikat RSA dengan panjang kunci 2048, 3072, dan 4096-bit, dan semua sertifikat ECDSA.	14 Juli 2021

---

<a href="#">Kelengkapan berbasis aplikasi</a>	Rilis ini menambahkan cookie berbasis aplikasi untuk mendukung sesi lekat untuk menyeimbangkan beban Anda.	8 Februari 2021
<a href="#">Kebijakan keamanan untuk FS yang mendukung TLS versi 1.2</a>	Rilis ini menambahkan kebijakan keamanan untuk Forward Secrecy (FS) yang mendukung TLS versi 1.2.	24 November 2020
<a href="#">WAF gagal membuka dukungan</a>	Rilis ini menambahkan dukungan untuk mengonfigurasi perilaku penyeimbang beban Anda jika terintegrasi dengan AWS WAF	13 November 2020
<a href="#">dukungan gRPC dan HTTP/2</a>	Rilis ini menambahkan dukungan untuk beban kerja gRPC dan HTTP/2. end-to-end	29 Oktober 2020
<a href="#">Dukungan pos terdepan</a>	Anda dapat menyediakan Application Load Balancer pada Anda. AWS Outposts	8 September 2020
<a href="#">Mode mitigasi desync</a>	Rilis ini menambahkan dukungan untuk mode mitigasi desinkronisasi.	17 Agustus 2020
<a href="#">Permintaan yang paling tidak beredar</a>	Rilis ini menambahkan dukungan untuk algoritme permintaan paling tidak tertunda.	25 November 2019

---

<a href="#">Kelompok sasaran tertimbang</a>	Rilis ini menambahkan dukungan untuk tindakan maju dengan beberapa kelompok target. Permintaan didistribusikan ke grup target ini berdasarkan berat yang Anda tentukan untuk setiap kelompok target.	19 November 2019
<a href="#">Atribut baru</a>	Rilis ini menambahkan dukungan untuk atribut routing.http.drop_invalid_header_fields.enabled.	15 November 2019
<a href="#">Kebijakan keamanan untuk FS</a>	Rilis ini menambahkan dukungan untuk tiga kebijakan keamanan kerahasiaan lanjutan yang telah ditentukan sebelumnya.	8 Oktober 2019
<a href="#">Perutean permintaan lanjutan</a>	Rilis ini menambahkan dukungan untuk jenis syarat tambahan untuk aturan listener Anda.	27 Maret 2019
<a href="#">Lambda berfungsi sebagai target</a>	Rilis ini menambahkan dukungan untuk mendaftarkan fungsi Lambda sebagai target.	29 November 2018
<a href="#">Tindakan pengalihan</a>	Rilis ini menambahkan dukungan untuk penyeimbang beban untuk mengalihkan permintaan ke URL yang berbeda.	25 Juli 2018

---

<a href="#">Tindakan respons tetap</a>	Rilis ini menambahkan dukungan untuk penyeimbang beban untuk mengembalikan respons HTTP kustom.	25 Juli 2018
<a href="#">Kebijakan keamanan untuk FS dan TLS 1.2</a>	Rilis ini menambahkan dukungan untuk dua kebijakan keamanan standar tambahan.	Selasa, 06 Juni 2018
<a href="#">Otentikasi pengguna</a>	Rilis ini menambahkan dukungan bagi penyeimbang beban untuk mengotentikasi pengguna aplikasi Anda menggunakan identitas perusahaan atau sosial mereka sebelum merutekan permintaan.	30 Mei 2018
<a href="#">Izin tingkat sumber daya</a>	Rilis ini menambahkan dukungan untuk izin tingkat sumber daya dan penandaan syarat kunci.	10 Mei 2018
<a href="#">Mode mulai lambat</a>	Rilis ini menambahkan dukungan untuk mode mulai lambat, yang secara bertahap meningkatkan pangsa permintaan penyeimbang beban mengirimkan ke target yang baru terdaftar saat pemanasan.	24 Maret 2018
<a href="#">Dukungan SNI</a>	Rilis ini menambahkan dukungan untuk Indikasi Nama Server (SNI).	10 Oktober 2017

---

<a href="#">Alamat IP sebagai target</a>	Rilis ini menambahkan dukungan untuk mendaftarkan alamat IP sebagai target.	31 Agustus 2017
<a href="#">Perutean berbasis host</a>	Rilis ini menambahkan dukungan untuk permintaan perutean berdasarkan nama host di header host.	5 April 2017
<a href="#">Kebijakan keamanan untuk TLS 1.1 dan TLS 1.2</a>	Rilis ini menambahkan kebijakan keamanan untuk TLS 1.1 dan TLS 1.2.	6 Februari 2017
<a href="#">IPv6 dukungan</a>	Rilis ini menambahkan dukungan untuk IPv6 alamat.	25 Januari 2017
<a href="#">Minta penelusuran</a>	Rilis ini menambahkan dukungan untuk permintaan pelacakan.	22 November 2016
<a href="#">Dukungan persentil untuk metrik TargetResponseTime</a>	Rilis ini menambahkan dukungan untuk statistik persentil baru yang didukung oleh Amazon. CloudWatch	17 November 2016
<a href="#">Jenis penyeimbang beban baru</a>	Pelepasan Elastic Load Balancing ini memperkenalkan Application Load Balancer.	11 Agustus 2016

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.