aws

Application Load Balancer

# Elastic Load Balancing

# Elastic Load Balancing: Application Load Balancer

# Table of Contents

# 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.

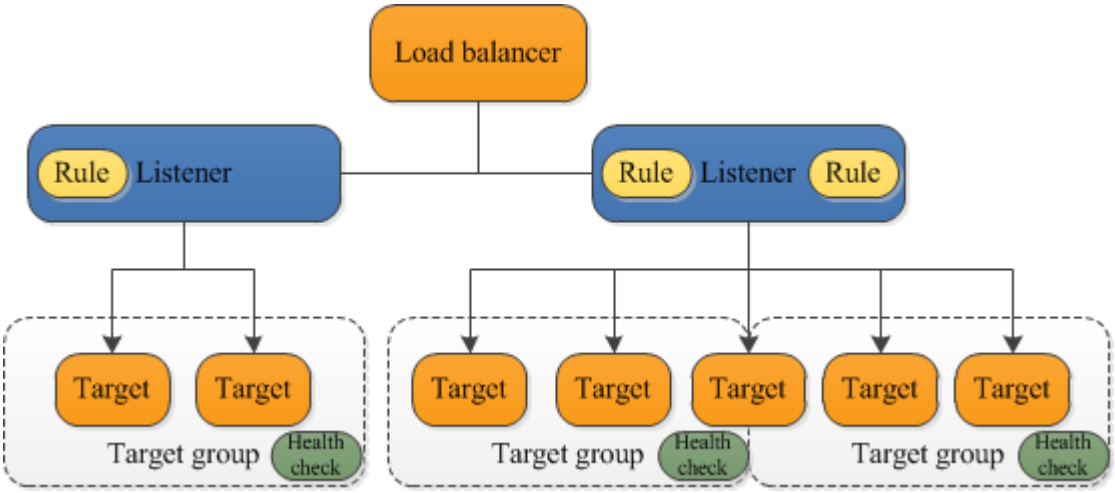
## Komponen Application Load Balancer

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 konfigurasikan. 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 setiap jenis penyeimbang beban, lihat Perbandingan produk untuk 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.
- 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 klaster 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 load balancers dalam satu Wilayah atau lebihAWS. Untuk informasi selengkapnya, lihat AWS Global Accelerator Panduan Developer.
- Route 53 — Menyediakan cara yang andal dan hemat biaya untuk mengarahkan pengunjung ke situs web dengan menerjemahkan nama domain (sepertiwww.example.com) ke alamat

IP numerik (seperti`192.0.2.1`) yang digunakan komputer untuk terhubung satu sama lain. AWSmenetapkan URL ke sumber daya Anda, seperti penyeimbang beban. Namun, Anda mungkin menginginkan 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 dengan Application Load Balancer Anda untuk mengizinkan atau memblokir permintaan berdasarkan aturan dalam daftar kontrol akses web (web ACL). Untuk informasi selengkapnya, lihat Aplikasi Load Balancer dan AWS WAF.

Untuk melihat informasi tentang layanan yang terintegrasi dengan penyeimbang beban Anda, pilih penyeimbang beban Anda di AWS Management Console 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?

# Memulai Application Load Balancer

Tutorial ini memberikan pengenalan langsung untuk Application Load Balancers melalui AWS Management Console, antarmuka berbasis web. Untuk membuat Application Load Balancer pertama Anda, selesaikan langkah berikut.

Tugas



Untuk demo konfigurasi penyeimbang beban umum, lihat Elastic Load Balancing.

## Sebelum Anda memulai

- Tentukan dua Availability Zone mana yang akan Anda gunakan untuk instans EC2 Anda. Konfigurasikan Virtual Private Cloud (VPC) Anda dengan setidaknya satu subnet publik di masing-masing Availability Zone ini. Subnet publik ini digunakan untuk mengonfigurasi penyeimbang beban. Anda dapat meluncurkan instans EC2 Anda di subnet lain dari Availability Zone ini sebagai gantinya.
- Luncurkan setidaknya satu instans EC2 di setiap Availability Zone. Pastikan untuk menginstal server web, seperti Apache atau Internet Information Services (IIS), pada setiap instans EC2. Pastikan bahwa grup keamanan untuk instans ini memungkinkan akses HTTP pada port 80.

## Langkah 1: Konfigurasi grup target Anda

Buat kelompok target, yang digunakan dalam permintaan perutean. Aturan default untuk listener Anda merutekan permintaan ke target terdaftar di grup target ini. Penyeimbang beban memeriksa kondisi target dalam grup target ini menggunakan pengaturan pemeriksaan kondisi yang ditentukan untuk grup target.

Untuk mengonfigurasi grup target Anda menggunakan konsol

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 Buat grup target.
4. Di bawah Konfigurasi dasar, pertahankan tipe Target sebagai contoh.
5. Untuk nama grup Target, masukkan nama untuk grup target baru.
6. Simpan protokol default (HTTP) dan port (80).
7. Pilih VPC yang berisi instance Anda. Pertahankan versi protokol sebagai HTTP1.
8. Untuk Pemeriksaan kondisi, simpan pengaturan default.
9. Pilih Selanjutnya.
10. Pada halaman Daftarkan target, selesaikan langkah berikut. Ini adalah langkah opsional untuk membuat penyeimbang beban. Namun, Anda harus mendaftarkan target ini jika Anda ingin menguji penyeimbang beban Anda dan memastikan bahwa itu mengarahkan lalu lintas ke target ini.
    a. Untuk Instans yang tersedia, pilih satu atau beberapa instans.
    b. Pertahankan port 80 default, dan pilih Sertakan sebagai tertunda di bawah ini.
11. Pilih Buat grup target.

## Langkah 2: Pilih jenis penyeimbang beban

Elastic Load Balancing mendukung berbagai jenis penyeimbang beban. Untuk tutorial ini, Anda perlu membuat Application Load Balancer.

Untuk membuat Application Load Balancer menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Pada bilah navigasi, pilih Wilayah untuk penyeimbang beban Anda. Pastikan untuk memilih Wilayah yang sama dengan yang Anda gunakan untuk instans EC2 Anda.
3. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Penyeimbang Beban.
4. Pilih Buat Penyeimbang Beban.
5. Untuk Application Load Balancer, pilih Create.

# Langkah 3: Konfigurasi penyeimbang beban dan pendengar Anda

Untuk membuat Application Load Balancer, Anda harus terlebih dahulu memberikan informasi konfigurasi dasar untuk penyeimbang beban Anda, seperti nama, skema, dan jenis alamat IP. Kemudian, Anda memberikan informasi tentang jaringan Anda, dan satu atau lebih pendengar. Listener adalah proses yang memeriksa permintaan koneksi. Listener dikonfigurasi dengan protokol dan port untuk koneksi dari klien ke penyeimbang beban. Untuk informasi selengkapnya tentang protokol dan port yang didukung, lihat Konfigurasi listener.

Untuk mengkonfigurasi penyeimbang beban dan pendengar Anda

1. Untuk Name penyeimbang beban, masukkan nama untuk penyeimbang beban Anda. Sebagai contoh, `my-alb`.
2. Untuk Skema dan Jenis alamat IP, simpan nilai default.
3. Untuk pemetaan Jaringan, pilih VPC yang Anda gunakan untuk instans EC2 Anda. Pilih setidaknya dua Availability Zone dan satu subnet per zona. Untuk setiap Availability Zone yang Anda gunakan untuk meluncurkan instans EC2 Anda, pilih Availability Zone dan kemudian pilih satu subnet publik untuk Availability Zone tersebut.
4. Untuk grup Keamanan, kami memilih grup keamanan default untuk VPC yang Anda pilih pada langkah sebelumnya. Anda dapat memilih grup keamanan yang berbeda sebagai gantinya. Kelompok keamanan harus menyertakan aturan yang memungkinkan penyeimbang beban untuk berkomunikasi dengan target terdaftar di port pendengar dan port pemeriksaan kesehatan. Untuk informasi selengkapnya, lihat Aturan grup keamanan.
5. Untuk Listener dan routing, pertahankan protokol dan port default, dan pilih grup target Anda dari daftar. Ini mengonfigurasi pendengar yang menerima lalu lintas HTTP pada port 80 dan meneruskan lalu lintas ke grup target yang dipilih secara default. Untuk tutorial ini, Anda tidak menciptakan listener HTTPS.
6. Untuk tindakan Default, pilih grup target yang Anda buat dan daftarkan di Langkah 1: Konfigurasikan grup target Anda.
7. (Opsional) Tambahkan tag untuk mengkategorikan penyeimbang beban Anda. Tombol tag harus unik untuk setiap penyeimbang beban. Karakter yang diperbolehkan adalah huruf, spasi, dan angka (dalam UTF-8) dan karakter berikut: + - = . _ : / @. Jangan gunakan spasi awal dan akhir. Kunci dan nilai tag peka huruf besar dan kecil.
8. Tinjau konfigurasi Anda, dan pilih Buat penyeimbang beban. Beberapa atribut default diterapkan pada penyeimbang beban Anda selama pembuatan. Anda dapat melihat dan mengeditnya

setelah membuat penyeimbang beban. Untuk informasi selengkapnya, lihat Atribut penyeimbang beban.

## Langkah 4: Uji penyeimbang beban Anda

Setelah membuat penyeimbang beban, verifikasi bahwa itu mengirim lalu lintas ke instans EC2 Anda.

Untuk menguji penyeimbang beban Anda

1. Setelah Anda diberi tahu bahwa penyeimbang beban Anda berhasil dibuat, pilih Tutup.
2. Pada panel navigasi, di bawah Penyeimbang Beban, pilih Grup Target.
3. Pilih grup target yang baru dibuat.
4. Pilih Target dan verifikasi bahwa instans Anda sudah siap. Jika status instans Anda adalah `initial`, mungkin dikarenakan instans masih dalam proses mendaftar, atau belum lulus jumlah pemeriksaan kesehatan minimum untuk dianggap sehat. Setelah status setidaknya satu instans adalah `healthy`, Anda dapat menguji penyeimbang beban Anda.
5. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Penyeimbang Beban.
6. Pilih penyeimbang beban yang baru dibuat.
7. Pilih Deskripsi dan salin nama DNS penyeimbang beban (misalnya, my-load-balancer -1234567890abcdef. elb.us-east-2.amazonaws.com). Tempelkan nama DNS ke bidang alamat browser web yang tersambung ke internet. Jika semuanya bekerja, peramban menampilkan halaman default server Anda.
8. (Opsional) Untuk menentukan aturan listener tambahan, lihat Tambahkan peraturan.

## Langkah 5: (Opsional) Hapus penyeimbang beban Anda

Segera setelah penyeimbang beban Anda tersedia, Anda akan dikenakan biaya untuk setiap jam atau sebagian jam yang Anda gunakan untuk menjalankan penyeimbang beban. Bila Anda tidak lagi memerlukan penyeimbang beban, Anda dapat menghapusnya. Segera setelah penyeimbang beban dihapus, Anda berhenti dikenakan biaya untuk itu. Perhatikan bahwa menghapus penyeimbang beban tidak memengaruhi target yang terdaftar pada penyeimbang beban. Misalnya, instans EC2 Anda terus berjalan setelah menghapus penyeimbang beban yang dibuat dalam panduan ini.

Untuk menghapus penyeimbang beban Anda menggunakan konsol

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 kotak centang untuk penyeimbang beban, pilih Tindakan, lalu pilih Hapus.
4. Ketika diminta konfirmasi, pilih Ya, Hapus.

# Tutorial: Membuat Application Load Balancer menggunakan AWS CLI

Tutorial ini memberikan pengenalan langsung ke Application Load Balancers melalui. AWS CLI

## Sebelum Anda mulai

- Gunakan perintah berikut untuk memverifikasi bahwa Anda menjalankan versi AWS CLI yang mendukung Application Load Balancers.

  ```
  aws elbv2 help
  ```

  Jika Anda mendapatkan pesan kesalahan bahwa elbv2 bukan pilihan yang valid, perbarui AWS CLI. Untuk informasi selengkapnya, lihat Menginstal AWS Command Line Interface dalam AWS Command Line Interface Panduan Pengguna Amazon EKS.
- Luncurkan instans EC2 Anda di virtual private cloud (VPC). Pastikan bahwa grup keamanan untuk instans ini memungkinkan akses pada port pendengar dan port pemeriksaan kesehatan. Untuk informasi selengkapnya, lihat Menargetkan grup keamanan.
- Putuskan apakah Anda akan membuat penyeimbang beban IPv4 atau dualstack. Gunakan IPv4 jika Anda ingin klien berkomunikasi dengan penyeimbang beban hanya menggunakan alamat IPv4. Gunakan dualstack jika Anda ingin klien berkomunikasi dengan penyeimbang beban menggunakan alamat IPv4 dan IPv6. Anda juga dapat menggunakan dualstack untuk berkomunikasi dengan target backend, seperti aplikasi IPv6 atau subnet dualstack, menggunakan IPv6.
- Pastikan untuk menginstal server web, seperti Apache atau Internet Information Services (IIS), pada setiap instans EC2. Pastikan bahwa grup keamanan untuk instans ini memungkinkan akses HTTP pada port 80.

## Membuat Load Balancer Anda

Untuk membuat Load Balancer pertama Anda, selesaikan langkah berikut.

Untuk membuat Load Balancer:

1. Gunakan create-load-balancerperintah untuk membuat penyeimbang beban. Anda harus menentukan dua subnet yang bukan dari Availability Zone yang sama.

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

Gunakan create-load-balancerperintah untuk membuat penyeimbang **dualstack** beban.

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

Ouput tersebut mencakup Amazon Resource Name (ARN) dari penyeimbang beban, dengan format berikut:

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

2. Gunakan create-target-groupperintah untuk membuat grup target, menentukan VPC yang sama yang Anda gunakan untuk instans EC2 Anda.

   Anda dapat membuat grup target IPv4 dan IPv6 untuk diasosiasikan dengan penyeimbang beban dualstack. Jenis alamat IP grup target menentukan versi IP yang akan digunakan penyeimbang beban untuk berkomunikasi, dan memeriksa kesehatan, target backend Anda.

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

   Luarannya mencakup ARN dari kelompok target, dengan format ini:

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

3. Gunakan perintah daftar-target untuk mendaftarkan instans Anda dengan grup target Anda:

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

4. Gunakan perintah buat-pendengar untuk membuat pendengar untuk Load Balancer Anda dengan aturan default yang meneruskan permintaan ke grup target Anda:

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

Luaran berisi ARN pendengar, dengan format berikut:

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

5. (Opsional) Anda dapat memverifikasi kesehatan target terdaftar untuk grup target Anda menggunakan describe-target-healthperintah ini:

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

## Menambahkan pendengar HTTPS

Jika Anda memiliki load balancer dengan pendengar HTTP, Anda dapat menambahkan pendengar HTTPS sebagai berikut.

Untuk menambahkan pendengar HTTPS ke Load Balancer

1. Buat sertifikat SSL untuk digunakan dengan Load Balancer Anda menggunakan salah satu metode berikut:
   - Membuat atau mengimpor sertifikat menggunakan AWS Certificate Manager (ACM). Untuk informasi selengkapnya, lihatMeminta sertifikatatauMengimpor sertifikatdiAWS Certificate Manager Panduan Pengguna.
   - Unggah sertifikat menggunakan AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, lihatBekerja dengan sertifikat serverdiPanduan Pengguna IAM.
2. Gunakan perintah buat-pendengar untuk membuat pendengar dengan aturan default yang meneruskan permintaan ke grup target Anda. Anda harus menentukan sertifikat SSL ketika Anda membuat pendengar HTTPS. Perhatikan bahwa Anda dapat menentukan kebijakan SSL selain default menggunakan pilihan `--ssl-policy`.

```
aws elbv2 create-listener --load-balancer-arn loadbalancer-arn \
--protocol HTTPS --port 443  \
--certificates CertificateArn=certificate-arn \
```

```
--default-actions Type=forward,TargetGroupArn=targetgroup-arn
```

## Tambahkan perutean berbasis jalur

Jika Anda memiliki pendengar dengan aturan default yang meneruskan permintaan ke satu kelompok target, Anda dapat menambahkan aturan yang meneruskan permintaan ke kelompok target lain berdasarkan URL. Misalnya, Anda dapat merutekan permintaan umum ke satu kelompok target dan permintaan untuk menampilkan gambar ke kelompok target lain.

Untuk menambahkan sebuah aturan untuk pendengar dengan sebuah pola jalur

1. Gunakan create-target-groupperintah untuk membuat grup target:

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

2. Gunakan perintah daftar-target untuk mendaftarkan instans Anda dengan grup target Anda:

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

3. Gunakan perintah buat-aturan untuk menambahkan aturan untuk pendengar Anda yang meneruskan permintaan ke kelompok target jika URL berisi pola tertentu:

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

## Menghapus Load Balancer Anda

Saat Anda tidak lagi memerlukan Load Balancer dan grup target, Anda dapat menghapusnya sebagai berikut:

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

# 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

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

- 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.

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 akan 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 satu atau beberapa subnet Zona Lokal. Pembatasan berikut berlaku:

- Anda tidak dapat menggunakan AWS WAF dengan penyeimbang beban.
- Anda tidak dapat menggunakan fungsi Lambda sebagai target.
- Anda tidak dapat menggunakan sesi lengket atau kelengketan aplikasi.

## 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, 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 AWS Health Dashboard 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 tersedia: Fungsi Lambda sebagai target, integrasi AWS WAF , sesi lekat, dukungan autentikasi, dan integrasi dengan AWS Global Accelerator.

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) |
|---|---|
| c5/c5d | |
| large | 50 |
| xlarge | 50 |
| 2xlarge | 50 |
| 4xlarge | 100 |
| m5/m5d | |
| large | 50 |
| xlarge | 50 |
| 2xlarge | 100 |
| 4xlarge | 100 |
| r5/r5d | |
| large | 50 |
| xlarge | 100 |
| 2xlarge | 100 |
| 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 direkomendasikan.

# 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

Berikut adalah atribut penyeimbang beban:

`access_logs.s3.enabled`

Menunjukkan apakah log akses yang disimpan di Amazon S3 diaktifkan. Default-nya adalah `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. Default-nya adalah `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). AWS 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`. Default-nya adalah `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`). Default-nya adalah `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`. Default-nya adalah `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`. Default-nya adalah `false`.

`routing.http.xff_header_processing.mode`

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

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

`routing.http2.enabled`

Menunjukkan apakah HTTP/2 diaktifkan. Default-nya adalah `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`. Default-nya adalah `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 salah satunya, tergantung pada 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 alamat IPv4 (misalnya, 192.0.2.1)

**`dualstack`**

Klien dapat terhubung ke penyeimbang beban menggunakan alamat IPv4 (misalnya, 192.0.2.1) dan alamat IPv6 (misalnya, 2001:0db8:85a3:0:0:8a2e:0370:7334).

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 alamat IPv4 menyelesaikan catatan DNS A. Klien yang berkomunikasi dengan penyeimbang beban menggunakan alamat IPv6 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). AWS VPN

**`dualstack-without-public-ipv4`**

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

Pertimbangan

- Autentikasi Application Load Balancer hanya mendukung IPv4 saat menghubungkan ke Endpoint Penyedia Identitas (IDP) atau Amazon Cognito. Tanpa alamat IPv4 publik penyeimbang beban tidak dapat menyelesaikan proses otentikasi, mengakibatkan kesalahan HTTP 500.

Untuk informasi selengkapnya tentang jenis alamat IP, lihatJenis alamat IP untuk Application Load Balancer Anda.

# 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 Application Load Balancer menggunakan konsol

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.

Kelompok 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. Misalnya, `HTTP:80`
  - Aturan - Pilih tindakan aturan. Misalnya, `Forward to target group`
  - Grup sasaran - Pilih nama grup target. Misalnya, `my-target-group`
  - Target - Pilih ID target. Misalnya, `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.

# 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.

## 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 konfigurasikan 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.

Secara default, Elastic Load Balancing menetapkan nilai batas waktu idle untuk penyeimbang beban Anda menjadi 60 detik, atau 1 menit. Gunakan prosedur berikut untuk mengatur nilai batas waktu idle yang berbeda.

Untuk memperbarui nilai batas waktu idle koneksi menggunakan konsol

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.

Untuk memperbarui nilai batas waktu idle menggunakan AWS CLI

Gunakan perintah modify-load-balancer-attributes dengan atribut `idle_timeout.timeout_seconds`.

## Durasi keepalive klien HTTP

Durasi keepalive klien HTTP adalah durasi maksimum waktu Application Load Balancer akan mempertahankan koneksi HTTP persisten ke klien. Setelah durasi keepalive klien HTTP yang dikonfigurasi telah berlalu, Application Load Balancer menerima satu permintaan dan 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 Balancers menetapkan nilai durasi keepalive klien HTTP 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 terus berjalan ketika tidak ada lalu lintas, dan tidak diatur ulang sampai koneksi baru dibuat.

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

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

Untuk memperbarui nilai durasi keepalive klien menggunakan konsol

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 durasi hidup klien HTTP. Kisaran yang valid adalah 60 hingga 604800 detik.
6. Pilih Simpan perubahan.

Untuk memperbarui nilai durasi keepalive klien menggunakan AWS CLI

Gunakan perintah modify-load-balancer-attributes dengan atribut `client_keep_alive.seconds`.

## 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”.

## 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.

Untuk mengaktifkan perlindungan penghapusan menggunakan konsol

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, aktifkan Perlindungan penghapusan.
6. Pilih Simpan perubahan.

Untuk menonaktifkan perlindungan penghapusan menggunakan konsol

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 halaman Konfigurasi, matikan Perlindungan penghapusan.
6. Pilih Simpan perubahan.

Untuk mengaktifkan atau menonaktifkan perlindungan penghapusan menggunakan AWS CLI

Gunakan perintah modify-load-balancer-attributes dengan atribut `deletion_protection.enabled`.

## 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[1] | Diblokir |
| Parah | Diizinkan | Diblokir | Diblokir |

[1] 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.

Untuk memperbarui mode mitigasi desync menggunakan konsol

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 Packet handling, untuk mode mitigasi Desync, pilih Defensive, Strictest, atau Monitor.
6. Pilih Simpan perubahan.

Untuk memperbarui mode mitigasi desync menggunakan AWS CLI

Gunakan perintah modify-load-balancer-attributes dengan atribut `routing.http.desync_mitigation_mode` diatur ke `monitor`, `defensive`, atau `strictest`.

# 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 pendengar HTTP/HTTPS 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 |
| 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 | `GET / index.ht` | example.com:8080 | example.com:8080 | example.com:8080 |

| Port pendengar | Contoh permintaan | Header host dalam permintaan | Pelestarian header host dinonaktifkan (perilaku default) | Pelestarian header host diaktifkan |
|---|---|---|---|---|
| port pendengar non-default dan header host memiliki port (misalnya, 8080). | `ml HTTP/1.1 Host: example.com:8080` | | | |

Untuk mengaktifkan pelestarian header host menggunakan konsol

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.

Untuk mengaktifkan pelestarian header host menggunakan AWS CLI

Gunakan perintah modify-load-balancer-attributes dengan atribut `routing.http.preserve_host_header.enabled` diatur ke `true`.

# Aplikasi Load Balancer dan AWS WAF

Anda dapat menggunakan AWS WAF Application Load Balancer untuk mengizinkan atau memblokir permintaan berdasarkan aturan dalam daftar kontrol akses web (web ACL). Untuk informasi selengkapnya, lihat Bekerja dengan ACL web di Panduan Developer AWS WAF .

Secara default, jika penyeimbang beban tidak bisa mendapatkan respons dari AWS WAF, ia mengembalikan kesalahan HTTP 500 dan tidak meneruskan permintaan. Jika Anda memerlukan penyeimbang beban untuk meneruskan permintaan ke target meskipun tidak dapat dihubungi AWS WAF, Anda dapat mengaktifkan AWS WAF integrasi. Untuk memeriksa apakah penyeimbang beban

Anda terintegrasi dengan AWS WAF, pilih penyeimbang beban Anda di AWS Management Console dan pilih tab Layanan terintegrasi.

ACL web yang telah ditentukan sebelumnya

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 mengaktifkan AWS WAF menggunakan konsol

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 Integrasi, perluas AWS Web Application Firewall (WAF), dan pilih Associate a WAF web ACL.
5. Di bawah Web ACL, pilih Auto-create pre-definied web ACL, atau pilih ACL web yang sudah ada.
6. Di bawah Tindakan aturan, pilih Blokir, atau Hitung.
7. Pilih Konfirmasi.

Untuk mengaktifkan AWS WAF gagal buka menggunakan AWS CLI

Gunakan perintah modify-load-balancer-attributes dengan atribut `waf.fail_open.enabled` diatur ke `true`.

# Membuat Application Load Balancer

Penyeimbang beban menerima permintaan dari klien dan mendistribusikannya ke seluruh target dalam grup target.

Sebelum memulai, pastikan Anda memiliki virtual private cloud (VPC) dengan setidaknya satu subnet publik di setiap zona yang digunakan oleh target Anda. Untuk informasi selengkapnya, lihat the section called “Subnet untuk penyeimbang beban Anda”.

Untuk membuat penyeimbang beban menggunakan AWS CLI, lihatTutorial: Membuat Application Load Balancer menggunakan AWS CLI.

Untuk membuat penyeimbang beban menggunakan AWS Management Console, selesaikan tugas-tugas berikut.

Tugas

- Langkah 1: Mengkonfigurasi grup target
- Langkah 2: Daftarkan target
- Langkah 3: Mengkonfigurasi penyeimbang beban dan pendengar
- Langkah 4: Uji penyeimbang beban

## Langkah 1: Mengkonfigurasi grup target

Mengkonfigurasi grup target memungkinkan Anda untuk mendaftarkan target seperti instans EC2. Grup target yang Anda mengkonfigurasi dalam langkah ini digunakan sebagai grup target dalam aturan pendengar saat Anda mengkonfigurasi penyeimbang beban. Untuk informasi selengkapnya, lihat Kelompok-kelompok target untuk Application Load Balancers.

Untuk mengonfigurasi grup target Anda menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Di panel navigasi, pilih Target Groups.
3. PilihBuat grup target.
4. Di bagian Konfigurasi dasar, atur parameter berikut:

a. Untuk Pilih jenis target, pilih Instans untuk menentukan target berdasarkan ID instans atau alamat IP untuk menentukan target hanya dengan alamat IP. Jika jenis target adalah fungsi Lambda, Anda dapat mengaktifkan pemeriksaan kesehatan dengan memilih Aktifkan di bagian Pemeriksaan Kesehatan.

b. Untuk Name, masukkan nama untuk grup target.

c. Ubah Port dan Protokol sesuai kebutuhan.

d. Jika jenis targetnya 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.

e. Untuk VPC, pilih virtual private cloud (VPC) dengan target yang ingin Anda sertakan dalam grup target Anda.

f. Untuk Versi protokol, pilih HTTP1 saat protokol permintaan adalah HTTP/1.1 atau HTTP/2; pilih HTTP2, ketika protokol permintaan adalah HTTP/2 atau gRPC; dan pilih gRPC, saat protokol permintaan adalah gRPC.

5. Di bagian Pemeriksaan Health, ubah pengaturan default sesuai kebutuhan. Untuk Pengaturan pemeriksaan kesehatan tingkat lanjut, pilih port pemeriksaan kesehatan, hitung, waktu habis, interval, dan tentukan kode keberhasilan. Jika pemeriksaan kesehatan secara berurutan melebihi jumlah Ambang batas tidak sehat, penyeimbang beban mengambil target keluar dari layanan. Jika pemeriksaan kesehatan secara berurutan melebihi jumlah Ambang batas sehat, penyeimbang beban menempatkan target kembali dalam pelayanan. Untuk informasi selengkapnya, lihat Pemeriksaan kondisi untuk grup target Anda.

6. (Opsional) Tambahkan satu atau lebih tag sebagai berikut:

   a. Perluas bagianTag.

   b. Pilih Tambahkan tanda.

   c. Masukkan tag Kunci and tag Nilai. Karakter yang diperbolehkan adalah huruf, spasi, dan angka (dalam UTF-8) dan karakter berikut: + - = . _ : / @. Jangan gunakan spasi awal dan akhir. Nilai tag peka huruf besar dan kecil.

7. Pilih Selanjutnya.

## Langkah 2: Daftarkan target

Anda dapat mendaftarkan instans EC2, alamat IP, atau fungsi Lambda sebagai target dalam grup target. Ini adalah langkah opsional untuk membuat penyeimbang beban. Namun, Anda harus mendaftarkan target Anda untuk memastikan bahwa penyeimbang beban Anda mengarahkan lalu lintas ke mereka.

1. Di halaman Daftarkan target, tambahkan satu atau lebih target sebagai berikut:
   - Jika jenis target Instans, pilih satu atau beberapa instans, masukkan satu atau beberapa port, lalu pilih Sertakan sebagai tertunda di bawah ini.
   - 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 targetnya adalah Lambda, pilih fungsi Lambda, atau masukkan ARN fungsi Lambda, lalu pilih Sertakan sebagai tertunda di bawah ini.
2. Pilih Buat grup target.

## Langkah 3: Mengkonfigurasi penyeimbang beban dan pendengar

Untuk membuat Application Load Balancer, Anda harus terlebih dahulu memberikan informasi konfigurasi dasar untuk penyeimbang beban Anda, seperti nama, skema, dan jenis alamat IP. Kemudian, Anda memberikan informasi tentang jaringan Anda, dan satu atau lebih pendengar. Listener adalah proses yang memeriksa permintaan koneksi. Listener dikonfigurasi dengan protokol dan port untuk koneksi dari klien ke penyeimbang beban. Untuk informasi selengkapnya tentang protokol dan port yang didukung, lihat Konfigurasi listener.

Untuk mengonfigurasi penyeimbang beban dan pendengar menggunakan konsol

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. Sebagai contoh, **`my-alb`**. Nama Application Load Balancer Anda harus unik dalam rangkaian Application Load Balancers dan Network Load Balancer 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-` Nama Application Load Balancer Anda tidak dapat diubah setelah dibuat.

   b. Untuk Skema, pilih Mengakses Internet atau Internal. Penyeimbang beban yang menghadap ke internet merutekan permintaan dari klien ke target melalui internet. Pengimbang beban internal merutekan permintaan ke target menggunakan alamat IP privat.

   c. Untuk jenis alamat IP, pilih IPv4, Dualstack, atau Dualstack tanpa IPv4 publik. Pilih IPv4 jika klien Anda menggunakan alamat IPv4 untuk berkomunikasi dengan penyeimbang beban. Pilih Dualstack jika klien Anda menggunakan alamat IPv4 dan IPv6 untuk berkomunikasi dengan penyeimbang beban. Pilih Dualstack tanpa IPv4 publik jika klien Anda hanya menggunakan alamat IPv6 untuk berkomunikasi dengan penyeimbang beban.

6. Pemetaan jaringan

   a. Untuk VPC, pilih VPC yang Anda gunakan untuk instans EC2 Anda. Jika Anda memilih Mengakses Internet untuk Skema, hanya VPC dengan internet gateway yang tersedia untuk dipilih.

   b. Untuk Pemetaan, aktifkan zona untuk penyeimbang beban Anda dengan memilih subnet sebagai berikut:

      - Subnet dari dua atau lebih Availability Zone
      - Subnet dari satu atau beberapa Local Zones
      - Satu subnet Outpost

      Untuk informasi selengkapnya, lihat the section called “Subnet untuk penyeimbang beban Anda”.

      Untuk penyeimbang beban internal, alamat IPv4 dan IPv6 ditetapkan dari subnet CIDR.

      Jika Anda mengaktifkan mode Dualstack untuk penyeimbang beban, pilih subnet dengan blok IPv4 dan IPv6 CIDR.

7. Untuk grup Keamanan, pilih grup keamanan yang ada, atau buat yang baru.

Grup keamanan untuk penyeimbang beban Anda harus mengizinkannya untuk berkomunikasi dengan target terdaftar pada port listener dan port pemeriksaan kondisi. Konsol dapat membuat grup keamanan untuk penyeimbang beban atas nama Anda dengan aturan yang mengizinkan komunikasi ini. Anda juga dapat membuat grup keamanan dan memilihnya sebagai gantinya. Untuk informasi selengkapnya, lihat Aturan yang direkomendasikan.

(Opsional) Untuk membuat grup keamanan baru untuk penyeimbang beban Anda, pilih Buat grup keamanan baru.

8. Untuk Listener dan routing, pendengar default menerima lalu lintas HTTP pada port 80. Anda dapat menyimpan protokol dan port default, atau memilih yang berbeda. Untuk tindakan Default, pilih grup target yang Anda buat. Anda dapat memilih Tambahkan pendengar untuk menambahkan pendengar lain (misalnya, pendengar HTTPS).
9. (Opsional) Jika menggunakan pendengar HTTPS

   Untuk kebijakan Keamanan, kami menyarankan Anda untuk selalu menggunakan kebijakan keamanan terbaru yang telah ditentukan sebelumnya.

   a. Untuk sertifikat SSL/TLS Default, opsi berikut tersedia:

      - Jika Anda membuat atau mengimpor sertifikat menggunakan AWS Certificate Manager, pilih Dari ACM, lalu pilih sertifikat dari Pilih sertifikat.
      - Jika Anda mengimpor sertifikat menggunakan IAM, pilih Dari IAM, lalu pilih sertifikat Anda dari Pilih sertifikat.
      - Jika Anda memiliki sertifikat untuk diimpor tetapi ACM tidak tersedia di Wilayah Anda, pilih Impor, lalu pilih Ke IAM. Ketik nama sertifikat di bidang Nama sertifikat. Dalam kunci privat Sertifikat, salin dan tempel isi file kunci pribadi (dikodekan PEM). Di badan Sertifikat, salin dan tempel isi file sertifikat kunci publik (dikodekan PEM). Di Rantai Sertifikat, salin dan tempel isi file rantai sertifikat (dikodekan PEM), kecuali Anda menggunakan sertifikat yang ditandatangani sendiri, dan bukan hal yang penting jika browser secara implisit menerima sertifikat.

   b. (Opsional) Untuk mengaktifkan otentikasi timbal balik, di bawah penanganan sertifikat Klien aktifkan Mutual Authentication (mTLS).

      Saat diaktifkan, mode TLS timbal balik default adalah passthrough.

      Jika Anda memilih Verifikasi dengan Trust Store:

- 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.
- Di bawah Trust Store pilih toko kepercayaan yang ada, atau pilih Toko kepercayaan baru.
  - Jika Anda memilih New trust store, berikan nama toko Trust, lokasi Otoritas Sertifikat URI S3, dan secara opsional lokasi daftar pencabutan Sertifikat URI S3.

10. (Opsional) Anda dapat mengintegrasikan layanan lain dengan penyeimbang beban selama pembuatan, di bawah Optimalkan dengan integrasi layanan.

    - Anda dapat memilih untuk menyertakan perlindungan AWS WAFkeamanan untuk penyeimbang beban Anda, dengan ACL web yang ada atau dibuat secara otomatis. Setelah pembuatan, ACL web dapat dikelola di AWS WAF konsol. Untuk informasi selengkapnya, lihat Mengaitkan atau memisahkan ACL web dengan AWS sumber daya di Panduan Pengembang.AWS WAF
    - Anda dapat memilih untuk AWS Global Acceleratormembuat akselerator untuk Anda dan mengaitkan penyeimbang beban Anda dengan akselerator. Nama akselerator dapat memiliki karakter berikut (hingga 64 karakter): a-z, A-Z, 0-9,. (periode), dan - (tanda hubung). Setelah akselerator dibuat, Anda dapat mengelolanya di AWS Global Accelerator konsol. Untuk informasi selengkapnya, lihat Menambahkan akselerator saat Anda membuat penyeimbang beban di Panduan AWS Global Accelerator Pengembang.

11. Tag dan buat

    a. (Opsional) Tambahkan tag untuk mengkategorikan penyeimbang beban Anda. Tombol tag harus unik untuk setiap penyeimbang beban. Karakter yang diperbolehkan adalah huruf, spasi, dan angka (dalam UTF-8) dan karakter berikut: + - = . _ : / @. Jangan gunakan spasi awal dan akhir. Kunci dan nilai tag peka huruf besar dan kecil.

    b. Tinjau konfigurasi Anda, dan pilih Buat penyeimbang beban. Beberapa atribut default diterapkan pada penyeimbang beban Anda selama pembuatan. Anda dapat melihat dan mengeditnya setelah membuat penyeimbang beban. Untuk informasi selengkapnya, lihat Atribut penyeimbang beban.

## Langkah 4: 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

# Availability Zone untuk Application Load Balancer Anda

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. 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 memperbarui Availability Zone menggunakan konsol

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.

Untuk memperbarui Availability Zone menggunakan AWS CLI

Gunakan perintah set-subnet.

# 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).

# Aturan yang direkomendasikan

Aturan berikut direkomendasikan untuk penyeimbang beban yang menghadap internet.

| Inbound | | |
|---|---|---|
| Source | Port Range | Comment |
| 0.0.0.0/0 | *pendengar* | Izinkan semua lalu lintas masuk pada port listener penyeimbang beban |

| Outbound | | |
|---|---|---|
| Destination | Port Range | Comment |
| *grup keamanan instans* | *pendengar contoh* | Izinkan lalu lintas keluar ke instans pada port listener instans |
| *grup keamanan instans* | *pemeriksaan kesehatan* | Izinkan lalu lintas keluar ke instans pada port pemeriksaan kondisi |

Aturan berikut direkomendasikan untuk penyeimbang beban internal.

| Inbound | | |
|---|---|---|
| Source | Port Range | Comment |
| *VPC CIDR* | *pendengar* | Izinkan lalu lintas masuk dari VPC CIDR pada port listener penyeimbang beban |

| Outbound | | |
|---|---|---|
| Destination | Port Range | Comment |

| | | |
|---|---|---|
| *grup keamanan instans* | *pendengar contoh* | Izinkan lalu lintas keluar ke instans pada port listener instans |
| *grup keamanan instans* | *pemeriksaan kesehatan* | Izinkan lalu lintas keluar ke instans pada port pemeriksaan kondisi |

Aturan berikut direkomendasikan untuk Application Load Balancer yang digunakan sebagai target Network Load Balancer.

Inbound

| Source | Port Range | Comment |
|---|---|---|
| *Alamat IP klien/CIDR* | *alb pendengar* | Izinkan lalu lintas klien masuk pada port pendengar penyeimbang beban |
| *VPC CIDR* | *alb pendengar* | Izinkan lalu lintas klien masuk melalui AWS PrivateLink pada port pendengar penyeimbang beban |
| *VPC CIDR* | *alb pendengar* | Izinkan lalu lintas kesehatan masuk dari Network Load Balancer |

Outbound

| Destination | Port Range | Comment |
|---|---|---|
| *grup keamanan instans* | *pendengar contoh* | Izinkan lalu lintas keluar ke instans pada port listener instans |

| | | |
|---|---|---|
| *grup keamanan instans* | *pemeriksaan kesehatan* | Izinkan lalu lintas keluar ke instans pada port pemeriksaan kondisi |

Perhatikan bahwa 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 mempelajari selengkapnya tentang pelacakan koneksi Amazon EC2, lihat Pelacakan koneksi grup keamanan di Panduan Pengguna Amazon EC2.

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.

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.

## Memperbarui grup keamanan terkait

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

Untuk memperbarui grup keamanan menggunakan konsol

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.

Untuk memperbarui grup keamanan menggunakan AWS CLI

Gunakan perintah set-security-groups.

# Jenis alamat IP untuk Application Load Balancer Anda

Anda dapat mengonfigurasi Application Load Balancer Anda sehingga klien dapat berkomunikasi dengan penyeimbang beban hanya menggunakan alamat IPv4, atau menggunakan alamat IPv4 dan IPv6 (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 penyeimbang beban harus memiliki blok CIDR IPv6 terkait. Untuk informasi selengkapnya, lihat alamat IPv6 di Panduan Pengguna Amazon EC2.
- Tabel rute untuk subnet penyeimbang beban harus merutekan lalu lintas IPv6.
- Grup keamanan untuk penyeimbang beban harus mengizinkan lalu lintas IPv6.
- ACL jaringan untuk subnet penyeimbang beban harus mengizinkan lalu lintas IPv6.

Untuk menetapkan jenis alamat IP pada penciptaan

Mengkonfigurasi pengaturan seperti yang dijelaskan di ???.

Untuk memperbarui jenis alamat IP menggunakan konsol

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 alamat IPv4 saja, Dualstack untuk mendukung alamat IPv4 dan IPv6, atau Dualstack tanpa IPv4 publik untuk mendukung alamat IPv6 saja.
6. Pilih Simpan perubahan.

Untuk memperbarui jenis alamat IP menggunakan AWS CLI

Gunakan perintah set-ip-address-type.

# Tag untuk Application Load Balancer Anda

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 prefiks ini tidak dihitung terhadap tag Anda per batas sumber daya.

Untuk memperbarui tag untuk penyeimbang beban menggunakan konsol

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 Tag, pilih Kelola tag, lalu lakukan satu atau beberapa hal berikut:
   a. Untuk memperbarui tanda, edit nilai Kunci dan Nilai.
   b. Untuk menambahkan tag baru, pilih Tambah tag dan kemudian masukkan nilai untuk Kunci dan Nilai.
   c. Untuk menghapus tag, pilih tombol Hapus di sebelah tag.
5. Setelah selesai memperbarui tag, pilihSimpan perubahan.

Untuk memperbarui tag untuk penyeimbang beban menggunakan AWS CLI

Gunakan perintah add-tags dan remove-tags.

# 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.

Untuk menghapus penyeimbang beban menggunakan konsol

1. 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.

   Contoh:

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

Untuk menghapus penyeimbang beban menggunakan AWS CLI

Gunakan perintah delete-load-balancer.

# Pergeseran zonal

Zonal shift adalah kemampuan di Amazon Route 53 Application Recovery Controller (Route 53 ARC). Dengan pergeseran zonal, Anda dapat mengalihkan sumber daya penyeimbang muatan dari Availability Zone yang terganggu dengan satu tindakan. Dengan cara ini, Anda dapat terus beroperasi dari Availability Zone sehat lainnya diWilayah AWS.

Saat Anda memulai pergeseran zonal, penyeimbang muatan Anda berhenti mengirim lalu lintas untuk sumber daya ke Availability Zone yang terpengaruh. Route 53 ARC menciptakan pergeseran zonal segera. Namun, diperlukan waktu singkat, biasanya hingga beberapa menit, untuk menyelesaikan koneksi yang ada dan sedang berlangsung di Availability Zone yang terpengaruh. Untuk informasi selengkapnya, lihat Cara kerja shift zonal: pemeriksaan kesehatan dan alamat IP zona di Panduan Pengembang Pengontrol Pemulihan Aplikasi Amazon Route 53.

Pergeseran zonal hanya didukung pada Application Load Balancers dan Network Load Balancers dengan load balancing lintas zona dimatikan. Jika Anda mengaktifkan load balancing lintas zona, Anda tidak dapat memulai pergeseran zonal. Untuk informasi selengkapnya, lihat Sumber daya yang didukung untuk perubahan zona di Panduan Developer Pengendali Pemulihan Aplikasi Amazon Route 53.

Sebelum Anda menggunakan pergeseran zonal, tinjau hal berikut:

- Penyeimbangan beban lintas zona tidak didukung dengan pergeseran zonal. Anda harus mematikan penyeimbangan beban lintas zona untuk menggunakan kemampuan ini.
- Pergeseran zona tidak didukung saat Anda menggunakan Application Load Balancer sebagai titik akhir akseleratorAWS Global Accelerator.
- Anda dapat memulai pergeseran zonal untuk penyeimbang muatan tertentu hanya untuk Availability Zone tunggal. Anda tidak dapat memulai pergeseran zonal untuk beberapa Availability Zone.
- AWSsecara proaktif menghapus alamat IP penyeimbang beban zonal dari DNS ketika beberapa infrastruktur mengeluarkan layanan berdampak. Selalu periksa kapasitas Availability Zone saat ini sebelum Anda memulai pergeseran zonal. Jika load balancer Anda telah mematikan load balancing lintas zona dan Anda menggunakan pergeseran zonal untuk menghapus alamat IP load balancer zonal, Availability Zone yang dipengaruhi oleh pergeseran zonal juga kehilangan kapasitas target.

- Ketika Application Load Balancer adalah target Network Load Balancer, selalu mulai pergeseran zonal dari Network Load Balancer. Jika Anda memulai pergeseran zonal dari Application Load Balancer, Network Load Balancer tidak mengenali shift dan terus mengirim lalu lintas ke Application Load Balancer.

Untuk panduan dan informasi selengkapnya, lihat Praktik terbaik dengan pergeseran zonal Route 53 ARC di Panduan Pengembang Pengontrol Pemulihan Aplikasi Amazon Route 53.

## Mulai pergeseran zonal

Langkah-langkah dalam prosedur ini menjelaskan cara memulai shift zonal menggunakan konsol Amazon EC2. Untuk langkah-langkah untuk memulai pergeseran zonal menggunakan konsol Route 53 ARC, lihat Memulai pergeseran zonal dalam Panduan Pengembang Pengontrol Pemulihan Aplikasi Amazon Route 53.

Untuk memulai shift zonal menggunakan konsol

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 nama penyeimbang beban.
4. Pada tab Integrasi, di bawah Route 53 Application Recovery Controller, pilih Mulai pergeseran zona.
5. Pilih Availability Zone tempat Anda ingin memindahkan lalu lintas.
6. Pilih atau masukkan kedaluwarsa untuk pergeseran zonal. Pergeseran zonal awalnya dapat diatur dari 1 menit hingga tiga hari (72 jam).

   Semua pergeseran zonal bersifat sementara. Anda harus menetapkan kedaluwarsa, tetapi Anda dapat memperbarui shift aktif nanti untuk menetapkan kedaluwarsa baru.
7. Masukkan komentar. Anda dapat memperbarui pergeseran zonal nanti untuk mengedit komentar, jika Anda suka.
8. Pilih kotak centang untuk mengetahui bahwa memulai pergeseran zonal akan mengurangi kapasitas aplikasi Anda dengan mengalihkan lalu lintas dari Availability Zone.
9. Pilih Mulai.

Untuk memulai pergeseran zonal menggunakanAWS CLI

Untuk bekerja dengan pergeseran zonal secara terprogram, lihat Panduan Referensi Zonal Shift API.

## Memperbarui pergeseran zonal

Langkah-langkah dalam prosedur ini menjelaskan cara memperbarui shift zonal menggunakan konsol Amazon EC2. Untuk langkah-langkah memperbarui pergeseran zonal menggunakan konsol Amazon Route 53 Application Recovery Controller, lihat Memperbarui pergeseran zonal di Panduan Pengembang Pengontrol Pemulihan Aplikasi Amazon Route 53.

Untuk memperbarui shift zona menggunakan konsol

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 nama load balancer yang memiliki pergeseran zonal aktif.
4. Pada tab Integrasi, di bawah Route 53 Application Recovery Controller, pilih Update zonal shift.

   Ini membuka konsol Route 53 ARC untuk melanjutkan pembaruan.
5. Untuk Tetapkan kedaluwarsa pergeseran zona, pilih atau masukkan kedaluwarsa secara opsional.
6. Untuk Komentar, opsional mengedit komentar yang ada atau masukkan komentar baru.
7. Pilih Update (Perbarui).

Untuk memperbarui pergeseran zonal menggunakanAWS CLI

Untuk bekerja dengan pergeseran zonal secara terprogram, lihat Panduan Referensi Zonal Shift API.

## Membatalkan shift zonal

Langkah-langkah dalam prosedur ini menjelaskan cara membatalkan shift zonal menggunakan konsol Amazon EC2. Untuk langkah-langkah untuk membatalkan pergeseran zonal menggunakan konsol Amazon Route 53 Application Recovery Controller, lihat Membatalkan pergeseran zonal dalam Panduan Pengembang Pengontrol Pemulihan Aplikasi Amazon Route 53.

Untuk membatalkan shift zonal menggunakan konsol

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 nama load balancer yang memiliki pergeseran zonal aktif.

4. Pada tab Integrasi, di bawah Route 53 Application Recovery Controller, pilih Cancel zonal shift.

   Ini membuka konsol Route 53 ARC untuk melanjutkan pembatalan.
5. Pilih Batalkan pergeseran zona.
6. Pada kotak dialog konfirmasi, pilih Konfirmasi.

Untuk membatalkan pergeseran zonal menggunakanAWS CLI

Untuk bekerja dengan pergeseran zonal secara terprogram, lihat Panduan Referensi Zonal Shift API.

# Listener untuk Application Load Balancer Anda

Listener adalah proses yang memeriksa permintaan koneksi, menggunakan protokol dan port yang Anda konfigurasikan. 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

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.

Application Load Balancers memberikan dukungan asli untuk. WebSockets Anda dapat meng-upgrade koneksi HTTP/1.1 yang ada ke koneksi WebSocket (`wsatauwss`) 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. Untuk informasi selengkapnya, lihat Cara Kerja WebSocket Protokol di Panduan CloudFront Pengembang Amazon.

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.

Untuk informasi lebih lanjut, lihat Perutean permintaan di Panduan Pengguna Elastic Load Balancing.

# Peraturan listener

Setiap pendengar memiliki tindakan default, juga dikenal sebagai aturan default. Aturan default tidak dapat dihapus dan selalu dilakukan terakhir. Aturan tambahan dapat dibuat dan terdiri dari prioritas, satu atau lebih tindakan, dan satu atau lebih kondisi. Anda dapat menambahkan atau mengedit peraturan kapan saja. Untuk informasi selengkapnya, lihat Mengedit peraturan.

## Peraturan default

Bila Anda membuat listener, Anda menentukan tindakan untuk peraturan default. Peraturan default tidak dapat memiliki syarat. Jika tidak ada syarat untuk peraturan listener yang terpenuhi, maka tindakan untuk peraturan default akan dilakukan.

Berikut ini adalah contoh peraturan default seperti yang ditunjukkan dalam konsol:

| Priority | Conditions (If) | Actions (Then) |
| --- | --- | --- |
| Last (default) | *If no other rule applies* | **Forward to target group**<br>• my-targets: 1 (100%)<br>• Group-level stickiness: Off |

### Prioritas peraturan

Setiap peraturan memiliki prioritas. 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 peraturan default. Untuk informasi selengkapnya, lihat Perbarui prioritas aturan.

### Tindakan aturan

Setiap tindakan aturan memiliki jenis, prioritas, dan informasi yang diperlukan untuk melakukan tindakan. Untuk informasi selengkapnya, lihat Jenis tindakan peraturan.

### Syarat peraturan

Setiap syarat peraturan memiliki jenis dan konfigurasi informasi. Bila syarat untuk suatu peraturan terpenuhi, maka tindakannya dilakukan. Untuk informasi selengkapnya, lihat Jenis syarat peraturan.

## Jenis tindakan peraturan

Berikut ini adalah jenis tindakan yang didukung untuk peraturan listener:

`authenticate-cognito`

[Listener HTTPS] Gunakan Amazon Cognito untuk mengautentikasi pengguna. Untuk informasi selengkapnya, lihat Mengautentikasi pengguna menggunakan Application Load Balancer.

`authenticate-oidc`

[Listener HTTPS] Gunakan penyedia identitas yang sesuai dengan OpenID Connect (OIDC) untuk mengautentikasi 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.

`redirect`

Mengalihkan permintaan dari satu URL ke URL lainnya. Untuk informasi selengkapnya, lihat Tindakan pengalihan.

Tindakan dengan prioritas terendah dilakukan terlebih dahulu. Setiap peraturan harus menyertakan satu dari tindakan berikut: `forward`, `redirect`, atau `fixed-response`, dan itu harus menjadi tindakan terakhir yang harus dilakukan.

Jika versi protokol adalah gRPC atau HTTP/2, satu-satunya tindakan yang didukung adalah tindakan. `forward`

## Tindakan respons tetap

Anda dapat menggunakan `fixed-response` untuk menjatuhkan permintaan klien dan mengembalikan respons HTTP khusus. 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 untuk AWS CLI

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

Anda dapat menggunakan tindakan `forward` untuk mengarahkan permintaan ke satu grup target atau lebih. 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.

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 kelengketan 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 dienkode 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.

Example 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"
              }
          ]
      }
  }
]
```

Example Contoh tindakan maju dengan dua kelompok target 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
              }
          ]
      }
  }
]
```

Example 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

Anda dapat menggunakan tindakan `redirect` untuk 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.

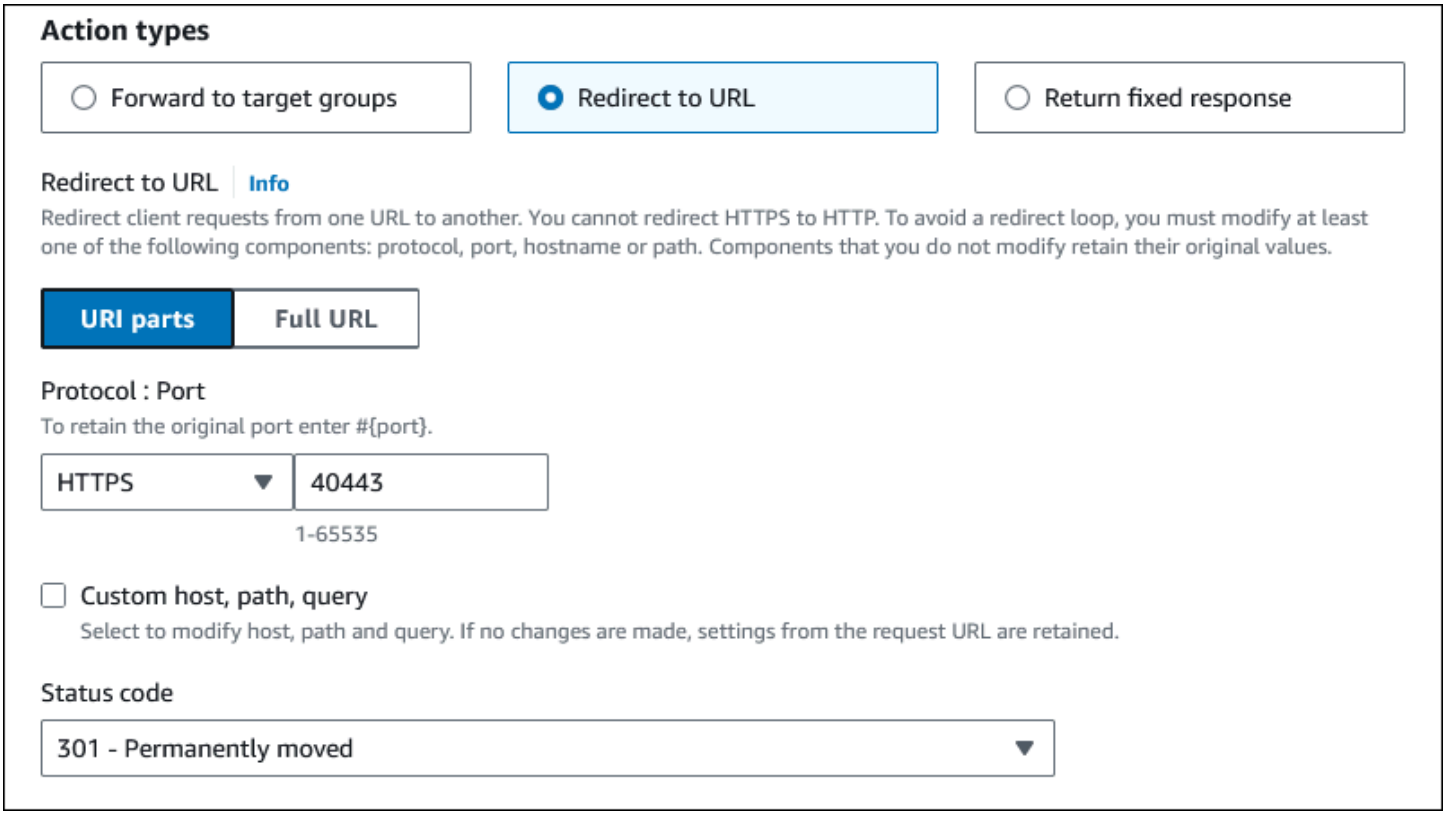
Anda dapat menggunakan kembali komponen URI dari URL asli di URL target menggunakan kata kunci cadangan berikut:

- `#{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 kuery.

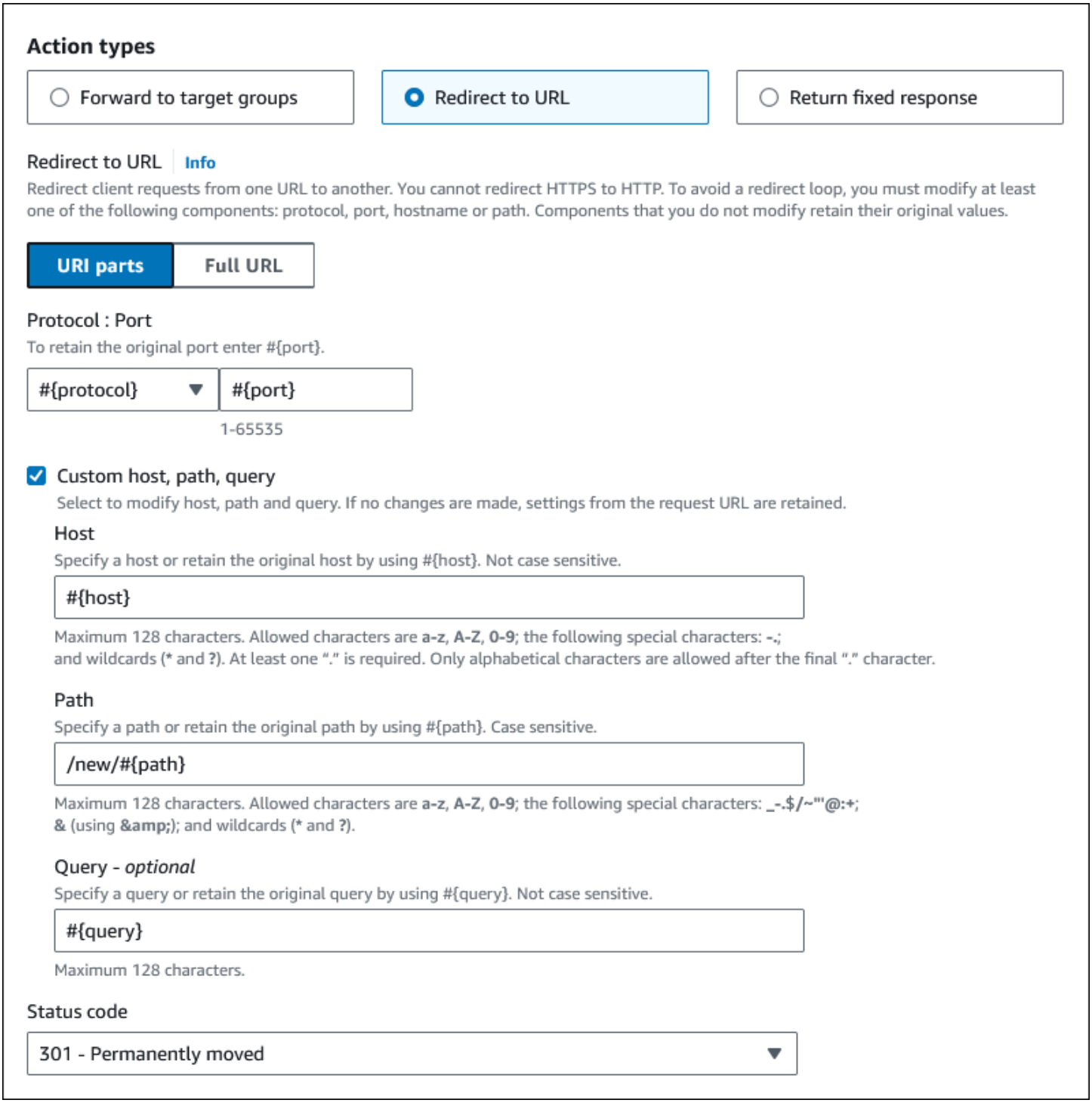
Saat tindakan `redirect` diambil, tindakan tersebut dicatat dalam log akses. Untuk informasi selengkapnya, lihat Entri log akses. Hitungan tindakan `redirect` yang berhasil dilaporkan dalam metrik `HTTP_Redirect_Count`. Untuk informasi selengkapnya, lihat Metrik Application Load Balancer.

Example Contoh tindakan pengalihan menggunakan konsol

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}".




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}".




Example Contoh tindakan pengalihan untuk AWS CLI

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.

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

## Jenis syarat peraturan

Berikut adalah jenis syarat yang didukung untuk peraturan:

`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`

Rutekan berdasarkan pola jalur di URL permintaan. Untuk informasi selengkapnya, lihat Syarat jalur.

`query-string`

Rutekan berdasarkan pasangan kunci/nilai 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.

Setiap peraturan secara opsional dapat menyertakan hingga salah satu dari masing-masing syarat berikut: `host-header`, `http-request-method`, `path-pattern`, dan `source-ip`. Setiap peraturan dapat juga secara opsional mencakup satu atau lebih dari masing-masing syarat berikut: `http-header` dan `query-string`.

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.

Untuk demo, lihat Perutean permintaan lanjutan.

## 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.

Example Contoh kondisi header HTTP untuk AWS CLI

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.

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

## 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 untuk AWS CLI

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 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 untuk AWS CLI

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 host yang cocok dengan string yang ditentukan.

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

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

## 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 untuk AWS CLI

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.

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

## Syarat string kueri

Anda dapat menggunakan syarat string kueri untuk mengonfigurasi aturan yang merutekan permintaan berdasarkan pasangan kunci/nilai 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 untuk AWS CLI

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 string kueri yang menyertakan pasangan kunci/nilai "versi=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 alamat IPv4 dan IPv6. 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.

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

Example Contoh kondisi IP sumber untuk AWS CLI

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"]
      }
  }
]
```

# 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.
- 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 listener HTTP menggunakan konsol

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: Port, pilih HTTP dan simpan port default atau masukkan port yang berbeda.
6. Untuk tindakan Default, pilih salah satu dari berikut ini:

    - Teruskan ke grup sasaran — Pilih satu atau lebih kelompok sasaran untuk meneruskan lalu lintas ke. Untuk menambahkan grup target pilih Tambahkan grup target. Jika menggunakan lebih dari satu kelompok sasaran, pilih bobot untuk setiap kelompok sasaran dan tinjau persentase yang terkait. Anda harus mengaktifkan kelengketan tingkat grup pada aturan, jika Anda telah mengaktifkan kekakuan pada satu atau beberapa grup target.
    - Redirect ke URL - Tentukan URL tempat permintaan klien akan dialihkan. Ini dapat dilakukan dengan memasukkan setiap bagian secara terpisah pada tab bagian URI, atau dengan memasukkan alamat lengkap pada tab URL Lengkap. Untuk kode Status, Anda dapat

mengonfigurasi pengalihan sebagai sementara (HTTP 302) atau permanen (HTTP 301) berdasarkan kebutuhan Anda.

- Kembalikan respons tetap - Tentukan kode Respons yang akan dikembalikan ke permintaan klien yang dijatuhkan. Selain itu, Anda dapat menentukan jenis Konten dan isi Respons, tetapi tidak diperlukan.

7. Pilih Tambahkan.

Untuk menambahkan pendengar HTTP menggunakan AWS CLI

Gunakan perintah buat-listener untuk membuat listener dan peraturan default, serta perintah buat-peraturan untuk menentukan peraturan listener tambahan.

# 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.

Application Load Balancers tidak mendukung tombol ED25519.

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.

Daftar Isi

## Sertifikat SSL

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 alat SSL/TLS 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 (IAM). AWS Identity and Access Management 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 Anda membuat listener HTTPS, memiliki sertifikat default dan daftar sertifikat kosong. Anda dapat menambahkan sertifikat ke daftar sertifikat untuk listener secara opsional. 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)
- Algoritme hashing (lebih suka SHA daripada MD5)

- 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

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:

- Application Load Balancers mendukung renegosiasi SSL hanya untuk koneksi target.
- Application Load Balancer tidak mendukung kebijakan keamanan kustom.
- `ELBSecurityPolicy-TLS13-1-2-2021-06`Kebijakan ini adalah kebijakan keamanan default untuk pendengar HTTPS yang dibuat menggunakan. AWS Management Console
- `ELBSecurityPolicy-2016-08`Kebijakan ini adalah kebijakan keamanan default untuk pendengar HTTPS yang dibuat menggunakan. AWS CLI
- Saat Anda membuat pendengar HTTPS, memilih kebijakan keamanan diperlukan.
  - Kami merekomendasikan kebijakan `ELBSecurityPolicy-TLS13-1-2-2021-06` keamanan, yang mencakup TLS 1.3, dan kompatibel dengan TLS 1.2.
- Anda dapat memilih kebijakan keamanan yang digunakan untuk koneksi front-end, tetapi tidak koneksi backend.
  - Untuk koneksi backend, jika pendengar HTTPS Anda menggunakan kebijakan keamanan TLS 1.3, kebijakan keamanan akan `ELBSecurityPolicy-TLS13-1-0-2021-06` digunakan. Jika tidak, kebijakan `ELBSecurityPolicy-2016-08` keamanan digunakan untuk koneksi backend.
- 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 Akses log untuk Application Load Balancer Anda.
- Anda dapat membatasi kebijakan keamanan yang tersedia untuk pengguna di seluruh Anda Akun AWS dan AWS Organizations dengan menggunakan kunci kondisi Elastic Load Balancing di IAM dan kebijakan kontrol layanan (SCP) Anda. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan (SCP) di AWS Organizations Panduan Pengguna

## Kebijakan keamanan TLS 1.3

Elastic Load Balancing menyediakan kebijakan keamanan TLS 1.3 berikut untuk Application Load Balancer:

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

## Kebijakan keamanan FIPS

> **Important**
>
> Semua pendengar aman yang melekat pada Application Load Balancer harus menggunakan kebijakan keamanan FIPS atau kebijakan keamanan non-FIPS; mereka tidak dapat dicampur. Jika Application Load Balancer yang ada memiliki dua atau lebih pendengar yang menggunakan kebijakan non-FIPS dan Anda ingin pendengar menggunakan kebijakan keamanan FIPS sebagai gantinya, hapus semua pendengar hingga hanya ada satu. Ubah kebijakan keamanan pendengar ke FIPS dan kemudian buat pendengar tambahan menggunakan kebijakan keamanan FIPS. Atau, Anda dapat membuat Application Load Balancer baru dengan pendengar baru hanya menggunakan 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.

Elastic Load Balancing menyediakan kebijakan keamanan FIPS berikut untuk Application Load Balancer:

- `ELBSecurityPolicy-TLS13-1-3-FIPS-2023-04`
- `ELBSecurityPolicy-TLS13-1-2-Res-FIPS-2023-04`

- `ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04`(Direkomendasikan)
- `ELBSecurityPolicy-TLS13-1-2-Ext0-FIPS-2023-04`
- `ELBSecurityPolicy-TLS13-1-2-Ext1-FIPS-2023-04`
- `ELBSecurityPolicy-TLS13-1-2-Ext2-FIPS-2023-04`
- `ELBSecurityPolicy-TLS13-1-1-FIPS-2023-04`
- `ELBSecurityPolicy-TLS13-1-0-FIPS-2023-04`

## Kebijakan yang didukung FS

Elastic Load Balancing menyediakan kebijakan keamanan yang didukung FS (Forward Secrecy) berikut untuk Application Load Balancer:

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

## Kebijakan keamanan TLS 1.0 - 1.2

Elastic Load Balancing menyediakan kebijakan keamanan TLS 1.0 - 1.2 berikut untuk Application Load Balancer:

- `ELBSecurityPolicy-TLS-1-2-Ext-2018-06`
- `ELBSecurityPolicy-TLS-1-2-2017-01`
- `ELBSecurityPolicy-TLS-1-1-2017-01`
- `ELBSecurityPolicy-2016-08`
- `ELBSecurityPolicy-TLS-1-0-2015-04`
- `ELBSecurityPolicy-2015-05`(identik dengan **`ELBSecurityPolicy-2016-08`**)

# Protokol dan cipher TLS

## TLS 1.3

Tabel berikut menjelaskan protokol dan cipher TLS yang didukung untuk kebijakan keamanan TLS 1.3 yang tersedia.

Catatan: `ELBSecurityPolicy-` Awalan telah dihapus dari nama kebijakan di baris kebijakan keamanan.

Contoh: Kebijakan keamanan `ELBSecurityPolicy-TLS13-1-2-2021-06` ditampilkan sebagai`TLS13-1-2-2021-06`.

| Kebijakan keamanan | TLS13-1-2-2021-06 | TLS13-1-3-2021-06 | TLS13-1-2-Res-2021-06 | TLS13-1-2-Ext2-2021-06 | TLS13-1-2-Ext1-2021-06 | TLS13-1-1-2021-06 | TLS13-1-0-2021-06 |
|---|---|---|---|---|---|---|---|
| Protokol TLS | | | | | | | |
| Protocol-TLSv1 | | | | | | | ✓ |
| Protocol-TLSv1.1 | | | | | | ✓ | ✓ |
| Protocol-TLSv1.2 | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| Protokol-TLSV1.3 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Cipher TLS | | | | | | | |
| TLS_AES_128_GCM_SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

| Kebijakan keamanan | TLS13-1-2-2021-06 | TLS13-1-3-2021-06 | TLS13-1-2-Res-2021-06 | TLS13-1-2-Ext2-2021-06 | TLS13-1-2-Ext1-2021-06 | TLS13-1-1-2021-06 | TLS13-1-0-2021-06 |
|---|---|---|---|---|---|---|---|
| TLS_AES_256_GCM_SHA384 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TLS_CHACHA20_POLY1305_SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-ECDSA-AES128-GCM-SHA256 | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES128-GCM-SHA256 | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-ECDSA-AES128-SHA256 | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES128-SHA256 | ✓ | | | ✓ | ✓ | ✓ | ✓ |

| Kebijakan keamanan | TLS13-1-2-2021-06 | TLS13-1-3-2021-06 | TLS13-1-2-Res-2021-06 | TLS13-1-2-Ext2-2021-06 | TLS13-1-2-Ext1-2021-06 | TLS13-1-1-2021-06 | TLS13-1-0-2021-06 |
|---|---|---|---|---|---|---|---|
| ECDHE-ECDSA-AES128-SHA | | | | ✓ | | ✓ | ✓ |
| ECDHE-RSA-AES128-SHA | | | | ✓ | | ✓ | ✓ |
| ECDHE-ECDSA-AES256-GCM-SHA384 | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES256-GCM-SHA384 | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-ECDSA-AES256-SHA384 | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES256-SHA384 | ✓ | | | ✓ | ✓ | ✓ | ✓ |

| Kebijakan keamanan | TLS13-1-2-2021-06 | TLS13-1-3-2021-06 | TLS13-1-2-Res-2021-06 | TLS13-1-2-Ext2-2021-06 | TLS13-1-2-Ext1-2021-06 | TLS13-1-1-2021-06 | TLS13-1-0-2021-06 |
|---|---|---|---|---|---|---|---|
| ECDHE-RSA-AES256-SHA | | | | ✓ | | ✓ | ✓ |
| ECDHE-ECDSA-AES256-SHA | | | | ✓ | | ✓ | ✓ |
| AES128-GCM-SHA256 | | | | ✓ | ✓ | ✓ | ✓ |
| AES128-SHA256 | | | | ✓ | ✓ | ✓ | ✓ |
| AES128-SHA | | | | ✓ | | ✓ | ✓ |
| AES256-GCM-SHA384 | | | | ✓ | ✓ | ✓ | ✓ |
| AES256-SHA256 | | | | ✓ | ✓ | ✓ | ✓ |
| AES256-SHA | | | | ✓ | | ✓ | ✓ |

Untuk membuat pendengar HTTPS yang menggunakan kebijakan TLS 1.3 menggunakan CLI

Gunakan perintah create-listener dengan kebijakan keamanan TLS 1.3 apa pun.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-TLS13-1-2-2021-06` keamanan.

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

Untuk memodifikasi pendengar HTTPS agar menggunakan kebijakan TLS 1.3 menggunakan CLI

Gunakan perintah modify-listener dengan kebijakan keamanan TLS 1.3 apa pun.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-TLS13-1-2-2021-06` keamanan.

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

Untuk melihat kebijakan keamanan yang digunakan oleh pendengar menggunakan CLI

Gunakan perintah describe-listeners dengan listener Anda. `arn`

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

Untuk melihat konfigurasi kebijakan keamanan TLS 1.3 menggunakan CLI

Gunakan perintah describe-ssl-policies dengan kebijakan keamanan TLS 1.3 apa pun.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-TLS13-1-2-2021-06` keamanan.

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

FIPS

> **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 modul FIPS140, mereka mungkin tidak sesuai dengan panduan NIST terbaru untuk konfigurasi TLS.

Tabel berikut menjelaskan protokol dan cipher TLS yang didukung untuk kebijakan keamanan FIPS yang tersedia.

Catatan: `ELBSecurityPolicy-` Awalan telah dihapus dari nama kebijakan di baris kebijakan keamanan.

Contoh: Kebijakan keamanan `ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04` ditampilkan sebagai`TLS13-1-2-FIPS-2023-04`.

| Kebijakan keamanan | TLS13-1-3-FIPS-2023-04 | TLS13-1-2-Res-FIPS-2023-04 | TLS13-1-2-FIPS-2023-04 | TLS13-1-2-Ext0-FIPS-2023-04 | TLS13-1-2-Ext1-FIPS-2023-04 | TLS13-1-2-Ext2-FIPS-2023-04 | TLS13-1-1-FIPS-2023-04 | TLS13-1-0-FIPS-2023-04 |
|---|---|---|---|---|---|---|---|---|
| Protokol TLS | | | | | | | | |
| Protocol-TLSv1 | | | | | | | | ✓ |
| Protocol-TLSv1.1 | | | | | | | ✓ | ✓ |
| Protocol-TLSv1.2 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Protokol-TLSV1.3 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Cipher TLS | | | | | | | | |

| Kebijakan keamanan | TLS13-1-3-FIPS-2023-04 | TLS13-1-2-Res-FIPS-2023-04 | TLS13-1-2-FIPS-2023-04 | TLS13-1-2-Ext0-FIPS-2023-04 | TLS13-1-2-Ext1-FIPS-2023-04 | TLS13-1-2-Ext2-FIPS-2023-04 | TLS13-1-1-FIPS-2023-04 | TLS13-1-0-FIPS-2023-04 |
|---|---|---|---|---|---|---|---|---|
| TLS_AES_128_GCM_SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| TLS_AES_256_GCM_SHA384 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-ECDSA-AES128-GCM-SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES128-GCM-SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-ECDSA-AES128-SHA256 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

| Kebijakan keamanan | TLS13-1-3-FIPS-2023-04 | TLS13-1-2-Res-FIPS-2023-04 | TLS13-1-2-FIPS-2023-04 | TLS13-1-2-Ext0-FIPS-2023-04 | TLS13-1-2-Ext1-FIPS-2023-04 | TLS13-1-2-Ext2-FIPS-2023-04 | TLS13-1-1-FIPS-2023-04 | TLS13-1-0-FIPS-2023-04 |
|---|---|---|---|---|---|---|---|---|
| ECDHE-RSA-AES128-SHA256 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-ECDSA-AES128-SHA | | | | ✓ | | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES128-SHA | | | | ✓ | | ✓ | ✓ | ✓ |
| ECDHE-ECDSA-AES256-GCM-SHA384 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

| Kebijakan keamanan | TLS13-1-3-FIPS-2023-04 | TLS13-1-2-Res-FIPS-2023-04 | TLS13-1-2-FIPS-2023-04 | TLS13-1-2-Ext0-FIPS-2023-04 | TLS13-1-2-Ext1-FIPS-2023-04 | TLS13-1-2-Ext2-FIPS-2023-04 | TLS13-1-1-FIPS-2023-04 | TLS13-1-0-FIPS-2023-04 |
|---|---|---|---|---|---|---|---|---|
| ECDHE-RSA-AES256-GCM-SHA384 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-ECDSA-AES256-SHA384 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES256-SHA384 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES256-SHA | | | | ✓ | | ✓ | ✓ | ✓ |
| ECDHE-ECDSA-AES256-SHA | | | | ✓ | | ✓ | ✓ | ✓ |

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

Untuk membuat pendengar HTTPS yang menggunakan kebijakan FIPS menggunakan CLI

Gunakan perintah create-listener dengan kebijakan keamanan FIPS apa pun.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04` keamanan.

```
aws elbv2 create-listener --name my-listener \
--protocol HTTPS --port 443 \
```

```
--ssl-policy ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04
```

Untuk memodifikasi pendengar HTTPS untuk menggunakan kebijakan FIPS menggunakan CLI

Gunakan perintah modify-listener dengan kebijakan keamanan FIPS apa pun.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04` keamanan.

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

Untuk melihat kebijakan keamanan yang digunakan oleh pendengar menggunakan CLI

Gunakan perintah describe-listeners dengan listener Anda. `arn`

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

Untuk melihat konfigurasi kebijakan keamanan FIPS menggunakan CLI

Gunakan perintah describe-ssl-policies dengan kebijakan keamanan FIPS apa pun.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-TLS13-1-2-FIPS-2023-04` keamanan.

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

FS

Tabel berikut menjelaskan protokol dan cipher TLS yang didukung untuk kebijakan keamanan yang didukung FS yang tersedia.

Catatan: `ELBSecurityPolicy-` Awalan telah dihapus dari nama kebijakan di baris kebijakan keamanan.

Contoh: Kebijakan keamanan `ELBSecurityPolicy-FS-2018-06` ditampilkan sebagai`FS-2018-06`.

| Kebijakan keamanan | Default | FS-1-2-Res-2020-10 | FS-1-2-Res-2019-08 | FS-1-2-2019-08 | FS-1-1-2019-08 | FS-2018-06 |
|---|---|---|---|---|---|---|
| Protokol TLS | | | | | | |
| Protocol-TLSv1 | ✓ | | | | | ✓ |
| Protocol-TLSv1.1 | ✓ | | | | ✓ | ✓ |
| Protocol-TLSv1.2 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Cipher TLS | | | | | | |
| ECDHE-ECDSA-AES128-GCM-SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES128-GCM-SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-ECDSA- | ✓ | | ✓ | ✓ | ✓ | ✓ |

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

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

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

Untuk membuat pendengar HTTPS yang menggunakan kebijakan yang didukung FS menggunakan CLI

Gunakan perintah create-listener dengan kebijakan keamanan yang didukung FS.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-FS-2018-06` keamanan.

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

Untuk memodifikasi pendengar HTTPS agar menggunakan kebijakan yang didukung FS menggunakan CLI

Gunakan perintah modify-listener dengan kebijakan keamanan yang didukung FS.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-FS-2018-06` keamanan.

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

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

Untuk melihat kebijakan keamanan yang digunakan oleh pendengar menggunakan CLI

Gunakan perintah describe-listeners dengan listener Anda. `arn`

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

Untuk melihat konfigurasi kebijakan keamanan yang didukung FS menggunakan CLI

Gunakan perintah describe-ssl-policies dengan kebijakan keamanan yang didukung FS.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-FS-2018-06` keamanan.

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

TLS 1.0 - 1.2

Tabel berikut menjelaskan protokol dan cipher TLS yang didukung untuk kebijakan keamanan TLS 1.0-1.2 yang tersedia.

Catatan: `ELBSecurityPolicy-` Awalan telah dihapus dari nama kebijakan di baris kebijakan keamanan.

Contoh: Kebijakan keamanan `ELBSecurityPolicy-TLS-1-2-Ext-2018-06` ditampilkan sebagai`TLS-1-2-Ext-2018-06`.

| Kebijakan keamanan | Default | TLS-1-2-Ext-2018-06 | TLS-1-2-2017-01 | TLS-1-1-2017-01 | TLS-1-0-2015-04* |
|---|---|---|---|---|---|
| Protokol TLS | | | | | |

| Kebijakan keamanan | Default | TLS-1-2-Ext-2018-06 | TLS-1-2-2017-01 | TLS-1-1-2017-01 | TLS-1-0-2015-04* |
|---|---|---|---|---|---|
| Protocol-TLSv1 | ✓ | | | | ✓ |
| Protocol-TLSv1.1 | ✓ | | | ✓ | ✓ |
| Protocol-TLSv1.2 | ✓ | ✓ | ✓ | ✓ | ✓ |
| Cipher TLS | | | | | |
| ECDHE-ECDSA-AES128-GCM-SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES128-GCM-SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-ECDSA-AES128-SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES128-SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ |

| Kebijakan keamanan | Default | TLS-1-2-Ext-2018-06 | TLS-1-2-2017-01 | TLS-1-1-2017-01 | TLS-1-0-2015-04* |
|---|---|---|---|---|---|
| ECDHE-ECDSA-AES128-SHA | ✓ | ✓ | | ✓ | ✓ |
| ECDHE-RSA-AES128-SHA | ✓ | ✓ | | ✓ | ✓ |
| ECDHE-ECDSA-AES256-GCM-SHA384 | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES256-GCM-SHA384 | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-ECDSA-AES256-SHA384 | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES256-SHA384 | ✓ | ✓ | ✓ | ✓ | ✓ |
| ECDHE-RSA-AES256-SHA | ✓ | ✓ | | ✓ | ✓ |

| Kebijakan keamanan | Default | TLS-1-2-Ext-2018-06 | TLS-1-2-2017-01 | TLS-1-1-2017-01 | TLS-1-0-2015-04* |
|---|---|---|---|---|---|
| ECDHE-ECDSA-AES256-SHA | ✓ | ✓ | | ✓ | ✓ |
| AES128-GCM-SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ |
| AES128-SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ |
| AES128-SHA | ✓ | ✓ | | ✓ | ✓ |
| AES256-GCM-SHA384 | ✓ | ✓ | ✓ | ✓ | ✓ |
| AES256-SHA256 | ✓ | ✓ | ✓ | ✓ | ✓ |
| AES256-SHA | ✓ | ✓ | | ✓ | ✓ |
| DES-CBC3-SHA | | | | | ✓ |

* Jangan gunakan kebijakan ini kecuali Anda harus mendukung klien lama yang memerlukan sandi DES-CBC3-SHA, yang merupakan sandi lemah.

Untuk membuat pendengar HTTPS yang menggunakan kebijakan TLS 1.0-1.2 menggunakan CLI

Gunakan perintah create-listener dengan kebijakan keamanan yang didukung TLS 1.0-1.2.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-2016-08` keamanan.

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

Untuk memodifikasi pendengar HTTPS agar menggunakan kebijakan TLS 1.0-1.2 menggunakan CLI

Gunakan perintah modify-listener dengan kebijakan keamanan yang didukung TLS 1.0-1.2.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-2016-08` keamanan.

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

Untuk melihat kebijakan keamanan yang digunakan oleh pendengar menggunakan CLI

Gunakan perintah describe-listeners dengan listener Anda. `arn`

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

Untuk melihat konfigurasi kebijakan keamanan TLS 1.0-1.2 menggunakan CLI

Gunakan perintah describe-ssl-policies dengan kebijakan keamanan yang didukung TLS 1.0-1.2.

Contohnya menggunakan kebijakan `ELBSecurityPolicy-2016-08` keamanan.

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

## Menambahkan listener HTTPS

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.

Prasyarat

- Untuk membuat listener HTTPS, Anda harus menentukan sertifikat dan kebijakan keamanan. Penyeimbang beban menggunakan sertifikat untuk mengakhiri koneksi dan mendekripsi permintaan dari klien sebelum mengarahkannya ke target. Penyeimbang beban menggunakan kebijakan keamanan saat menegosiasikan koneksi SSL dengan klien.
- Untuk menambahkan tindakan maju ke peraturan listener default, Anda harus menentukan grup target yang tersedia. Untuk informasi selengkapnya, lihat Buat grup target.
- 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.

Untuk menambahkan listener HTTPS menggunakan konsol

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: Port, pilih HTTPS dan simpan port default atau masukkan port yang berbeda.
6. (Opsional) Untuk mengaktifkan otentikasi, di bawah Otentikasi pilih Gunakan OpenID atau Amazon Cognito, dan berikan informasi yang diminta. Untuk informasi selengkapnya, lihat Mengautentikasi pengguna menggunakan Application Load Balancer.
7. Untuk Tindakan default, lakukan salah satu dari berikut ini:

   - Teruskan ke grup sasaran — Pilih satu atau lebih kelompok sasaran untuk meneruskan lalu lintas ke. Untuk menambahkan grup target pilih Tambahkan grup target. Jika menggunakan lebih dari satu kelompok sasaran, pilih bobot untuk setiap kelompok sasaran dan tinjau persentase yang terkait. Anda harus mengaktifkan kelengketan tingkat grup pada aturan, jika Anda telah mengaktifkan kekakuan pada satu atau beberapa grup target.
   - Redirect ke URL - Tentukan URL tempat permintaan klien akan dialihkan. Ini dapat dilakukan dengan memasukkan setiap bagian secara terpisah pada tab bagian URI, atau dengan memasukkan alamat lengkap pada tab URL Lengkap. Untuk kode Status, Anda dapat mengonfigurasi pengalihan sebagai sementara (HTTP 302) atau permanen (HTTP 301) berdasarkan kebutuhan Anda.

- Kembalikan respons tetap - Tentukan kode Respons yang akan dikembalikan ke permintaan klien yang dijatuhkan. Selain itu, Anda dapat menentukan jenis Konten dan isi Respons, tetapi tidak diperlukan.

8. Untuk kebijakan Keamanan, kami menyarankan Anda untuk selalu menggunakan kebijakan keamanan terbaru yang telah ditentukan sebelumnya.
9. Untuk sertifikat SSL/TLS Default, opsi berikut tersedia:

   - Jika Anda membuat atau mengimpor sertifikat menggunakan AWS Certificate Manager, pilih Dari ACM, lalu pilih sertifikat dari Pilih sertifikat.
   - Jika Anda mengimpor sertifikat menggunakan IAM, pilih Dari IAM, lalu pilih sertifikat Anda dari Pilih sertifikat.
   - Jika Anda memiliki sertifikat untuk diimpor tetapi ACM tidak tersedia di Wilayah Anda, pilih Impor, lalu pilih Ke IAM. Ketik nama sertifikat di bidang Nama sertifikat. Dalam kunci privat Sertifikat, salin dan tempel isi file kunci pribadi (dikodekan PEM). Di badan Sertifikat, salin dan tempel isi file sertifikat kunci publik (dikodekan PEM). Di Rantai Sertifikat, salin dan tempel isi file rantai sertifikat (dikodekan PEM), kecuali Anda menggunakan sertifikat yang ditandatangani sendiri, dan bukan hal yang penting jika browser secara implisit menerima sertifikat.

10. (Opsional) Untuk mengaktifkan otentikasi timbal balik, di bawah penanganan sertifikat Klien aktifkan Mutual Authentication (mTLS).

    Saat diaktifkan, mode TLS timbal balik default adalah passthrough.

    Jika Anda memilih Verifikasi dengan Trust Store:

    - 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.
    - Di bawah Trust Store pilih toko kepercayaan yang ada, atau pilih Toko kepercayaan baru.
      - Jika Anda memilih New trust store, berikan nama toko Trust, lokasi Otoritas Sertifikat URI S3, dan secara opsional lokasi daftar pencabutan Sertifikat URI S3.

11. Pilih Simpan.

Untuk menambahkan pendengar HTTPS menggunakan AWS CLI

Gunakan perintah buat-listener untuk membuat listener dan peraturan default, serta perintah buat-peraturan untuk menentukan peraturan listener tambahan.

# Aturan listener untuk Application Load Balancer Anda

Peraturan yang Anda tetapkan untuk listener Anda menentukan cara load balancer mengarahkan permintaan ke target dalam satu atau beberapa grup target.

Setiap peraturan terdiri dari prioritas, satu tindakan atau lebih, dan satu syarat atau lebih. Untuk informasi selengkapnya, lihat Peraturan listener.

## Persyaratan

- Aturan hanya dapat dilampirkan untuk pendengar yang aman.
- Setiap peraturan harus mencakup salah satu tindakan berikut: `forward`, `redirect`, atau `fixed-response`, dan harus menjadi tindakan terakhir yang harus dilakukan.
- Setiap peraturan dapat mencakup tidak ada atau salah satu dari kondisi berikut: `host-header`, `http-request-method`, `path-pattern`, dan `source-ip`, dan tidak ada atau salah satu ketentuan-ketentuan berikut: `http-header` dan `query-string`.
- Anda dapat menentukan hingga tiga string perbandingan per syarat dan hingga lima per peraturan.
- 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.

## Tambahkan peraturan

Anda menentukan peraturan default ketika Anda membuat listener, dan Anda dapat menentukan aturan nondefault tambahan setiap saat.

Untuk menambahkan peraturan menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban untuk melihat detailnya.
4. Pada tab Listeners and rules, lakukan salah satu hal berikut:

a. Pilih teks di kolom Protocol:Port untuk membuka halaman detail untuk pendengar.

   Pada tab Aturan, pilih Tambahkan aturan.

b. Pilih listener yang ingin Anda tambahkan aturannya.

   Pilih Kelola aturan, lalu Tambahkan aturan.

5. Anda dapat menentukan nama untuk aturan Anda di bawah Nama dan tag, meskipun itu tidak diperlukan.

   Untuk menambahkan tag tambahan pilih teks Tambahkan tag tambahan.

6. Pilih Selanjutnya.
7. Pilih Tambahkan syarat.
8. Tambahkan satu atau beberapa kondisi berikut:

   - Header Host - Tentukan header host. Misalnya: `*.example.com`. Untuk menyimpan kondisi, pilih Konfirmasi.

     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?).

   - Path — Tentukan jalurnya. Misalnya: `/item/*` . Untuk menyimpan kondisi, pilih Konfirmasi.

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

   - Metode permintaan HTTP - Tentukan metode permintaan HTTP. Untuk menyimpan kondisi, pilih Konfirmasi.

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

   - Sumber IP — Tentukan alamat IP sumber dalam format CIDR. Untuk menyimpan kondisi, pilih Konfirmasi.

     IPv4 dan IPv6 CIDR diperbolehkan. Wildcard tidak didukung.

   - Header HTTP — Masukkan nama header dan tambahkan satu atau lebih string perbandingan. Untuk menyimpan kondisi, pilih Konfirmasi.
     - Nama header HTTP - Aturan akan menilai permintaan yang berisi header ini untuk mengonfirmasi nilai yang cocok.

Maksimal 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.

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

  Maksimal 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?).

- String kueri - Permintaan rute berdasarkan pasangan kunci: nilai atau nilai dalam string kueri. Untuk menyimpan kondisi, pilih Konfirmasi.

  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?).

9. Pilih Selanjutnya.
10. Tentukan salah satu tindakan berikut untuk aturan Anda:

    - Teruskan ke grup sasaran — Pilih satu atau lebih kelompok sasaran untuk meneruskan lalu lintas ke. Untuk menambahkan grup target pilih Tambahkan grup target. Jika menggunakan lebih dari satu kelompok sasaran, pilih bobot untuk setiap kelompok sasaran dan tinjau persentase yang terkait. Anda harus mengaktifkan kelengketan tingkat grup pada aturan, jika Anda telah mengaktifkan kekakuan pada satu atau beberapa grup target.
    - Redirect ke URL - Tentukan URL tempat permintaan klien akan dialihkan. Ini dapat dilakukan dengan memasukkan setiap bagian secara terpisah pada tab bagian URI, atau dengan memasukkan alamat lengkap pada tab URL Lengkap. Untuk kode Status, Anda dapat mengonfigurasi pengalihan sebagai sementara (HTTP 302) atau permanen (HTTP 301) berdasarkan kebutuhan Anda.
    - Kembalikan respons tetap - Tentukan kode Respons yang akan dikembalikan ke permintaan klien yang dijatuhkan. Selain itu, Anda dapat menentukan jenis Konten dan isi Respons, tetapi tidak diperlukan.

11. Pilih Selanjutnya.
12. Tentukan Prioritas aturan Anda dengan memasukkan nilai dari 1-50000.
13. Pilih Selanjutnya.
14. Tinjau semua detail dan pengaturan yang saat ini dikonfigurasi untuk aturan baru Anda. Setelah puas dengan pilihan Anda, pilih Buat.

Untuk menambahkan aturan menggunakan AWS CLI

Gunakan perintah buat-peraturan untuk membuat peraturan. Gunakan perintah jelaskan-peraturan untuk menampilkan informasi tentang peraturan.

## Mengedit peraturan

Anda dapat mengedit tindakan dan syarat untuk peraturan 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.

Untuk mengedit peraturan menggunakan konsol

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, lakukan salah satu hal berikut:
   - Pilih teks di kolom Protocol:Port untuk membuka halaman detail untuk pendengar.
     i. Pada tab Aturan, di bagian Aturan pendengar, pilih teks di kolom Tag nama untuk aturan yang ingin Anda edit.

        Pilih Tindakan, lalu Edit aturan.
     ii. Pada tab Aturan, di bagian Aturan pendengar, pilih aturan yang ingin Anda edit.

        Pilih Tindakan, lalu Edit aturan.
5. Ubah nama dan tag sesuai kebutuhan. Untuk menambahkan tag tambahan pilih teks Tambahkan tag tambahan.
6. Pilih Berikutnya
7. Ubah kondisi sesuai kebutuhan. Anda dapat menambahkan, mengedit kondisi yang ada, atau menghapus.
8. Pilih Berikutnya
9. Ubah tindakan sesuai kebutuhan.
10. Pilih Berikutnya
11. Ubah prioritas aturan sesuai kebutuhan. Anda dapat memasukkan nilai dari 1-50000.

12. Pilih Berikutnya
13. Tinjau semua detail dan pengaturan terbaru yang dikonfigurasi untuk aturan Anda. Setelah puas dengan pilihan Anda, pilih Simpan perubahan.

Untuk mengedit aturan menggunakan AWS CLI

Gunakan perintah modifikasi-peraturan.

## 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 peraturan default.

Untuk memperbarui prioritas aturan menggunakan konsol

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, lakukan salah satu hal berikut:
   a. Pilih teks di kolom Protocol:Port atau Rules untuk membuka halaman detail untuk listener.
      i. Pilih Tindakan, lalu Prioritaskan ulang aturan.
      ii. Pada tab Aturan, di bagian Aturan pendengar, pilih Tindakan lalu Prioritaskan ulang aturan.
   b. Pilih pendengar.
      - Pilih Kelola aturan, lalu Prioritaskan ulang aturan
5. Di bagian Aturan Listener, kolom Prioritas menampilkan prioritas aturan saat ini. Anda dapat memperbarui prioritas aturan dengan memasukkan nilai dari 1-50000.
6. Setelah puas dengan perubahan, pilih Simpan perubahan.

Untuk memperbarui prioritas aturan menggunakan AWS CLI

Gunakan perintah atur-prioritas-peraturan.

## Menghapus peraturan

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

Untuk menghapus peraturan menggunakan konsol

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, lakukan salah satu hal berikut:
   a. Pilih teks di kolom Protocol:Port atau Rules untuk membuka halaman detail untuk listener.
      i. Pilih aturan yang ingin Anda hapus.
      ii. Pilih Tindakan, lalu Hapus aturan
      iii. Ketik `confirm` bidang teks, lalu pilih Hapus.
   b. Pilih teks di kolom Tag nama untuk membuka halaman detail aturan.
      i. Pilih Tindakan, lalu Hapus aturan.
      ii. Ketik `confirm` bidang teks, lalu pilih Hapus.

Untuk menghapus aturan menggunakan AWS CLI

Gunakan perintah hapus-peraturan.

# 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

## Mengganti sertifikat default

Anda dapat mengganti sertifikat default untuk listener Anda menggunakan prosedur berikut. Untuk informasi selengkapnya, lihat Sertifikat SSL.

Untuk mengubah sertifikat default menggunakan konsol

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. Pilih Simpan sebagai default.

Untuk mengubah sertifikat default menggunakan AWS CLI

Gunakan perintah modifikasi-listener.

## Menambahkan sertifikat ke daftar sertifikat

Anda dapat menambahkan sertifikat ke daftar sertifikat untuk listener Anda menggunakan prosedur berikut. Ketika Anda pertama kali membuat listener HTTPS, daftar sertifikatnya kosong. Anda dapat menambahkan satu atau beberapa sertifikat. Anda dapat menambahkan sertifikat default untuk memastikan bahwa sertifikat ini digunakan dengan protokol SNI bahkan jika diganti sebagai sertifikat default. Untuk informasi selengkapnya, lihat Sertifikat SSL.

Untuk mengubah sertifikat default menggunakan konsol

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 Tambahkan sertifikat.

6. Dalam tabel sertifikat ACM dan IAM, pilih sertifikat yang akan ditambahkan dan pilih Sertakan sebagai tertunda di bawah ini.
7. Jika Anda memiliki sertifikat yang tidak dikelola oleh ACM atau IAM, pilih Impor sertifikat, lengkapi formulir, dan pilih Impor.
8. Pilih Tambahkan sertifikat yang tertunda.

Untuk menambahkan sertifikat ke daftar sertifikat menggunakan AWS CLI

Gunakan perintah tambahkan-sertifikat-listener.

## Menghapus sertifikat dari daftar sertifikat

Anda dapat menghapus sertifikat dari daftar sertifikat untuk HTTPS listener menggunakan prosedur berikut. Untuk menghapus sertifikat default untuk listener HTTPS, lihat Mengganti sertifikat default.

Untuk menghapus sertifikat dari daftar sertifikat menggunakan konsol

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.

Untuk menghapus sertifikat dari daftar sertifikat menggunakan AWS CLI

Gunakan perintah hapus-sertifikat-listener.

## 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.

Menggunakan kebijakan FIPS pada Application Load Balancer Anda:

Semua pendengar aman yang melekat pada Application Load Balancer harus menggunakan kebijakan keamanan FIPS atau kebijakan keamanan non-FIPS; mereka tidak dapat dicampur. Jika Application Load Balancer yang ada memiliki dua atau lebih pendengar yang menggunakan kebijakan non-FIPS dan Anda ingin pendengar menggunakan kebijakan keamanan FIPS sebagai gantinya, hapus semua pendengar hingga hanya ada satu. Ubah kebijakan keamanan pendengar ke FIPS dan kemudian buat pendengar tambahan menggunakan kebijakan keamanan FIPS. Atau, Anda dapat membuat Application Load Balancer baru dengan pendengar baru hanya menggunakan kebijakan keamanan FIPS.

Untuk memperbarui kebijakan keamanan menggunakan konsol

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 halaman Detail, pilih Tindakan, lalu Edit pendengar.
6. Di bagian Pengaturan pendengar aman, di bawah Kebijakan keamanan, pilih kebijakan keamanan baru.
7. Pilih Simpan perubahan.

Untuk memperbarui kebijakan keamanan menggunakan AWS CLI

Gunakan perintah modifikasi-listener.

# 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 menyederhanakan manajemen otentikasi dan mengurangi beban pada aplikasi Anda.

Dengan menggunakan TLS timbal balik dengan Application Load Balancer, 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, Application

Load Balancer mengautentikasi klien dengan sertifikat dari otoritas sertifikat pihak ketiga (CA) atau dengan menggunakan AWS Private Certificate Authority (PCA), secara opsional, dengan pemeriksaan pencabutan. Application Load Balancer meneruskan informasi sertifikat klien ke backend, yang dapat digunakan aplikasi Anda untuk otorisasi. Dengan menggunakan TLS timbal balik di Application Load Balancer, Anda bisa mendapatkan otentikasi terkelola bawaan, terukur, dan terkelola untuk entitas berbasis sertifikat, yang menggunakan pustaka yang sudah mapan.

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

Catatan: Sertifikat klien X.509v1 tidak didukung.

- Mutual TLS passthrough: Ketika Anda menggunakan modus passthrough TLS timbal balik, Application Load Balancer mengirimkan seluruh rantai sertifikat klien ke target menggunakan header HTTP. Kemudian, dengan menggunakan rantai sertifikat klien, Anda dapat menerapkan logika otentikasi dan otorisasi yang sesuai dalam aplikasi Anda.
- Verifikasi TLS bersama: Saat Anda menggunakan mode verifikasi TLS timbal balik, Application Load Balancer melakukan otentikasi sertifikat klien X.509 untuk klien saat penyeimbang beban menegosiasikan koneksi TLS.

Untuk memulai dengan TLS timbal balik di Application Load Balancer menggunakan passthrough, Anda hanya perlu mengonfigurasi listener untuk menerima sertifikat apa pun dari klien. Untuk menggunakan TLS timbal balik dengan verfikasi, Anda harus melakukan 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.

Untuk step-by-step prosedur untuk mengonfigurasi modus verifikasi TLS timbal balik dengan Application Load Balancer Anda, 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 RSA/SHA256, 384, 512 dengan hash EC/SHA256,384,512 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 ModifyTrustStore API.

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, lihatHeader 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. #
- 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-Mtls`Header 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 Klien

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 string RFC2253 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 string RFC2253 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 format ISO8601 dari dan tanggal. `notBefore` `notAfter`

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...>NmrUlw
%0A-----END%20CERTIFICATE-----%0A
```

## Mengkonfigurasi TLS timbal balik pada Application Load Balancer

Bagian ini mencakup prosedur untuk mengonfigurasi modus verifikasi TLS timbal balik untuk otentikasi pada Application Load Balancers.

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.

Ikuti prosedur di bagian ini untuk mengonfigurasi modus verifikasi TLS timbal balik pada Application Load Balancer Anda di. AWS Management Console Untuk mengonfigurasi TLS timbal balik dengan menggunakan operasi API alih-alih konsol, lihat Panduan Referensi API Application Load Balancer.

Tugas

- Buat toko kepercayaan
- Kaitkan toko kepercayaan
- Lihat detail toko kepercayaan
- Memodifikasi toko kepercayaan
- Hapus toko kepercayaan

## Buat toko kepercayaan

Ada tiga cara untuk membuat toko kepercayaan: saat Anda membuat Application Load Balancer, saat Anda membuat pendengar yang aman, dan dengan menggunakan konsol Trust Store. Saat Anda menambahkan toko kepercayaan saat membuat penyeimbang beban atau pendengar, toko kepercayaan secara otomatis dikaitkan dengan pendengar baru. Saat Anda membuat toko kepercayaan menggunakan konsol Trust Store, Anda harus mengaitkannya dengan pendengar sendiri.

Bagian ini mencakup pembuatan toko kepercayaan menggunakan konsol Trust Store, tetapi langkah-langkah yang digunakan saat membuat Application Load Balancer atau pendengar adalah sama. Untuk info selengkapnya, lihat Mengonfigurasi penyeimbang beban dan pendengar dan Menambahkan pendengar HTTPS.

Prasyarat:

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

Untuk membuat toko kepercayaan menggunakan konsol

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 bundel otoritas Sertifikat, masukkan jalur Amazon S3 ke bundel sertifikat ca yang ingin digunakan oleh toko kepercayaan Anda.

Opsional: Gunakan versi Object untuk memilih versi sebelumnya dari bundel sertifikat ca. Jika tidak, versi saat ini digunakan.

5. Untuk Pencabutan, Anda dapat menambahkan daftar pencabutan sertifikat ke toko kepercayaan Anda secara opsional.

   - Di bawah Daftar pencabutan sertifikat, masukkan jalur Amazon S3 ke daftar pencabutan sertifikat yang ingin digunakan oleh toko kepercayaan Anda.

     Opsional: Gunakan versi Objek untuk memilih versi sebelumnya dari daftar pencabutan sertifikat. Jika tidak, versi saat ini digunakan.

6. Untuk tag toko Trust, Anda dapat memasukkan hingga 50 tag secara opsional untuk diterapkan ke toko kepercayaan Anda.
7. Pilih Buat toko kepercayaan.

## 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.

Bagian ini mencakup mengaitkan toko kepercayaan ke pendengar yang ada. Atau, Anda dapat mengaitkan toko kepercayaan saat membuat Application Load Balancer atau pendengar. Untuk info selengkapnya, lihat Mengonfigurasi penyeimbang beban dan pendengar dan Menambahkan pendengar HTTPS.

Untuk mengaitkan toko kepercayaan menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Pada panel navigasi, pilih Load Balancers.
3. Pilih penyeimbang beban untuk melihat halaman detailnya.
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. (Opsional) Jika TLS timbal balik tidak diaktifkan, pilih Mutual authentication (mTLS) di bawah penanganan sertifikat Klien dan kemudian pilih Verifikasi dengan trust store.

7. Di bawah Trust store, pilih toko kepercayaan yang Anda buat.
8. Pilih Simpan perubahan.

## Lihat detail toko kepercayaan

### Bundel sertifikat CA

Bundel sertifikat CA adalah komponen wajib 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.

Anda dapat melihat konten bundel sertifikat CA saat ini di toko kepercayaan Anda kapan saja.

### Lihat bundel sertifikat CA

Untuk melihat bundel sertifikat CA menggunakan konsol

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 detail.
4. Pilih Tindakan, lalu Dapatkan bundel CA.
5. Pilih tautan Bagikan, atau Unduh.

### 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. ID pencabutan meningkat untuk setiap daftar pencabutan yang ditambahkan ke toko kepercayaan, dan mereka tidak dapat diubah. Jika daftar pencabutan sertifikat dihapus dari toko kepercayaan, ID pencabutan itu juga dihapus dan tidak digunakan kembali selama masa pakai toko kepercayaan.

> **Note**
>
> Application Load Balancers tidak dapat mencabut sertifikat yang memiliki nomor seri negatif, dalam daftar pencabutan sertifikat.

Melihat daftar pencabutan sertifikat

Untuk melihat daftar pencabutan menggunakan konsol

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 detail.
4. Pada tab Daftar pencabutan sertifikat, pilih Tindakan, lalu Dapatkan daftar pencabutan.
5. Pilih tautan Bagikan, atau Unduh.

## Memodifikasi toko kepercayaan

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.

Ganti bundel sertifikat CA

Untuk mengganti bundel sertifikat CA menggunakan konsol

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 detail.
4. Pilih Tindakan, lalu 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.

### Tambahkan daftar pencabutan sertifikat

Untuk menambahkan daftar pencabutan menggunakan konsol

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, lalu 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

### Menghapus daftar pencabutan sertifikat

Untuk menghapus daftar pencabutan menggunakan konsol

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 detail.
4. Pada tab Daftar pencabutan sertifikat, pilih Tindakan, lalu Hapus daftar pencabutan.
5. Konfirmasikan penghapusan dengan mengetik. `confirm`
6. Pilih Hapus.

## Hapus toko kepercayaan

Ketika Anda tidak lagi memiliki penggunaan untuk toko kepercayaan, Anda dapat menghapusnya.

Catatan: Anda tidak dapat menghapus toko kepercayaan yang saat ini dikaitkan dengan pendengar.

Untuk menghapus toko kepercayaan menggunakan konsol

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. Pilih Tindakan, lalu Hapus toko kepercayaan.
5. Konfirmasikan penghapusan dengan mengetik. `confirm`
6. Pilih Hapus

## 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

# 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), atau OAuth, 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 tepercaya.
- Entri DNS untuk titik akhir harus dapat diselesaikan secara publik, bahkan jika mereka memutuskan ke alamat IP pribadi.
- Izinkan salah satu URL pengalihan berikut di aplikasi IdP Anda, mana pun yang akan digunakan pengguna Anda, dengan DNS adalah nama domain penyeimbang beban Anda dan CNAME adalah alias DNS untuk aplikasi Anda:
  - https://*DNS*/oauth2/idpresponse
  - https://*CNAME*/oauth2/idpresponse

## Bersiap menggunakan Amazon Cognito

Wilayah Tersedia

Integrasi Amazon Cognito untuk Application Load Balancers tersedia di wilayah berikut:

- AS Timur (N. Virginia)
- AS Timur (Ohio)
- AS Barat (California Utara)
- AS Barat (Oregon)
- Kanada (Pusat)
- Eropa (Stockholm)
- Eropa (Milan)
- Eropa (Frankfurt)
- Eropa (Zürich)
- Eropa (Irlandia)
- Eropa (London)
- Eropa (Paris)
- Amerika Selatan (São Paulo)
- Asia Pasifik (Tokyo)

- Asia Pasifik (Seoul)
- Asia Pasifik (Osaka)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Singapura)
- Asia Pasifik (Sydney)
- Asia Pasifik (Jakarta)
- Timur Tengah (UEA)
- Timur Tengah (Bahrain)
- Afrika (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 membuat rahasia klien, menggunakan alur pemberian kode, dan mendukung cakupan OAuth 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 lebih lanjut, lihat Menambahkan nama Domain untuk kolam pengguna di Panduan Developer 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.

  Catatan: Application Load Balancer tidak mendukung token akses khusus yang dikeluarkan oleh Amazon Cognito. Untuk informasi selengkapnya, lihat Pra pembuatan token di Panduan Pengembang Amazon Cognito.
- Untuk bergabung dengan IdP sosial atau perusahaan, aktifkan IdP di bagian federasi. Untuk informasi lebih lanjut, lihat Menambahkan sign-in sosial ke kolam pengguna atau Menambahkan sign-in dengan IdP SAML ke kolam pengguna di Panduan Developer Amazon Cognito.
- Izinkan URL pengalihan berikut di bidang URL panggilan balik untuk Amazon Cognito, di mana DNS adalah nama domain penyeimbang beban Anda, dan CNAME adalah alias DNS untuk aplikasi Anda (jika Anda menggunakannya):

  - https://*DNS*/oauth2/idpresponse
  - https://*CNAME*/oauth2/idpresponse
- Izinkan domain kolam pengguna Anda di URL panggilan balik aplikasi IdP Anda. Gunakan format untuk IdP Anda. Sebagai contoh:
  - https://*domain-prefix*.auth.*region*.amazoncognito.com/saml2/idpresponse
  - https://*user-pool-domain*/oauth2/idpresponse

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.

## 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 AuthenticateCognitoActionConfigdan AuthenticateOidcActionConfigdi 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 Balancers tidak mendukung nilai-nilai cookie yang URL dikodekan.

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 ACL jaringan untuk VPC Anda mengizinkan akses ke titik akhir. Saat menggunakan penyeimbang beban internal atau jenis alamat IP`dualstack-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": "abcdefghijklmnopqrstuvwxyz123456789",
        "ClientSecret": "12345678901234567890123456789O",
        "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
}]
```

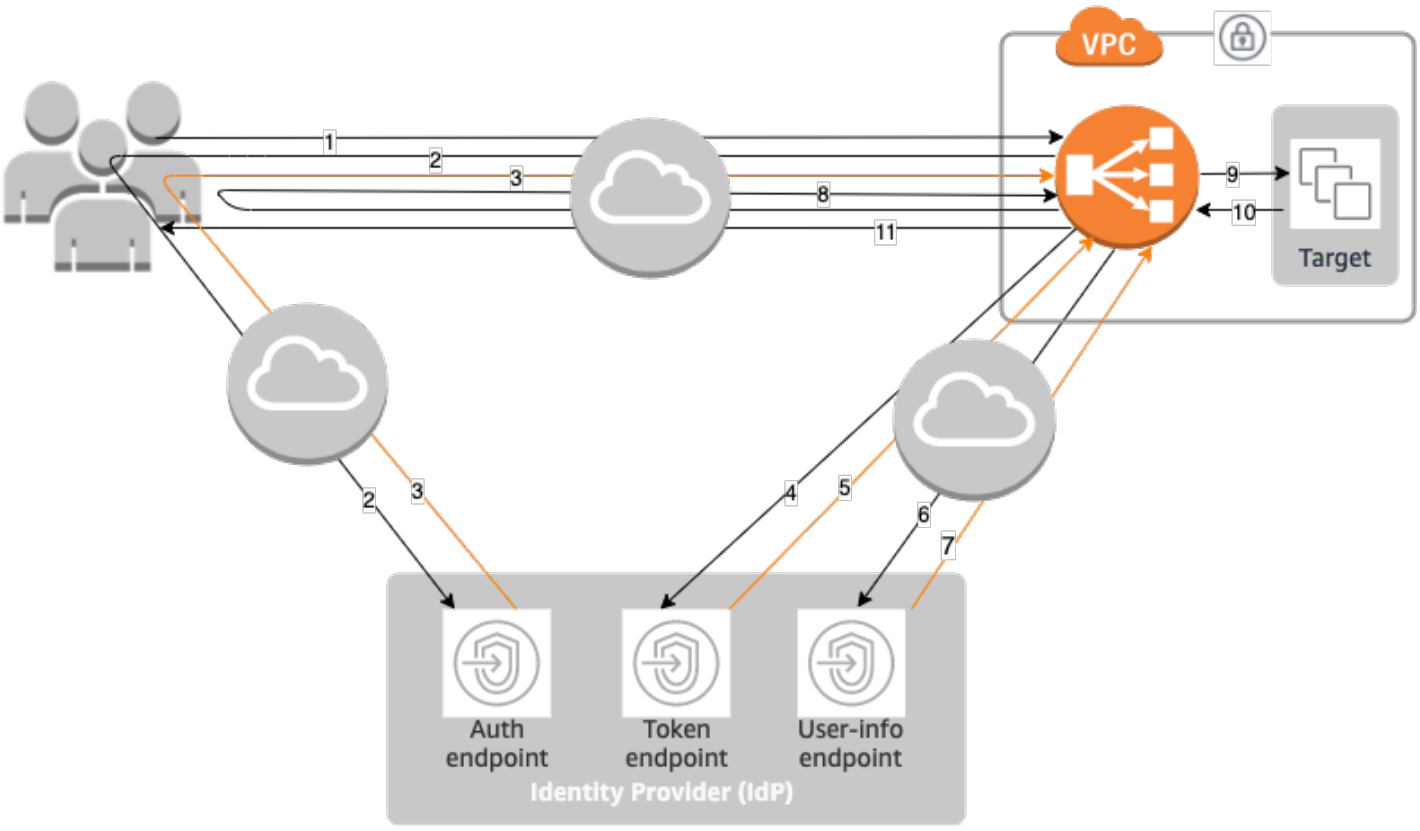
Berikut ini adalah contoh file `actions.json` yang menentukan tindakan `authenticate-cognito` dan tindakan `forward`.

```
[{
    "Type": "authenticate-cognito",
    "AuthenticateCognitoConfig": {
        "UserPoolArn": "arn:aws:cognito-idp:region-code:account-id:userpool/user-pool-id",
        "UserPoolClientId": "abcdefghijklmnopqrstuvwxyz123456789",
        "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 Peraturan listener.

## 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.

`AWSALBAuthNonce`Cookie 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 dienkripsi.

## 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",
   ...
}
```

Karena penyeimbang beban tidak mengenkripsi klaim pengguna, kami menyarankan Anda mengonfigurasi grup target untuk menggunakan HTTPS. Jika Anda mengonfigurasi grup target untuk menggunakan HTTP, pastikan untuk membatasi lalu lintas ke penyeimbang beban Anda menggunakan grup keamanan.

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
```

Contoh berikut menunjukkan cara mendapatkan ID kunci, kunci publik, dan payload di Python 3.x:

```
import jwt
import requests
import base64
import json

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

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

assert expected_alb_arn == received_alb_arn, "Invalid Signer"

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

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

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

Contoh berikut menunjukkan cara mendapatkan ID kunci, kunci publik, dan payload di Python 2.7:

```
import jwt
import requests
import base64
import json

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

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

assert expected_alb_arn == received_alb_arn, "Invalid Signer"

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

# Step 3: Get the public key from regional endpoint
```

```
url = 'https://public-keys.auth.elb.' + region + '.amazonaws.com/' + kid
req = requests.get(url)
pub_key = req.text

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

Pertimbangan

- Contoh-contoh ini tidak mencakup cara memvalidasi tanda tangan penerbit dengan tanda tangan di token.
- Pustaka standar tidak kompatibel dengan padding yang disertakan dalam token otentikasi Application Load Balancer dalam format 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 menghubungi IdP untuk menentukan apakah pengguna masih masuk.

Halaman arahan logout klien adalah halaman yang tidak diautentikasi. Ini berarti bahwa itu tidak boleh berada di belakang aturan Application Load Balancer yang memerlukan autentikasi.

- 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.

# 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.

## 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 adalah contoh header permintaan `X-Forwarded-For` untuk klien dengan alamat IPv6 `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 alamat IPv4 `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 alamat IPv6 `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

`preserve`Mode dalam atribut memastikan bahwa `X-Forwarded-For` header dalam permintaan HTTP tidak dimodifikasi dengan cara apa pun sebelum dikirim ke target.

### Menghapus

`remove`Mode dalam atribut menghapus `X-Forwarded-For` header dalam permintaan HTTP sebelum dikirim ke target.

> **Note**
>
> 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 | XFF dalam mode **`append`** | XFF dalam mode **`preserve`** | XFF dalam mode **`remove`** |
|---|---|---|---|---|
| Permintaan dikirim tanpa header XFF | `GET / index.ht ml HTTP/1.1` | `X-Forward ed-For: 127.0.0.1` | Tidak hadir | Tidak hadir |

| Minta deskripsi | Contoh permintaan | XFF dalam mode **append** | XFF dalam mode **preserve** | XFF dalam mode **remove** |
|---|---|---|---|---|
| | `Host: example.com` | | | |
| Permintaan dikirim dengan header XFF dan alamat IP klien. | `GET / index.html HTTP/1.1 Host: example.com X-Forwarded-For: 127.0.0.4` | `X-Forwarded-For: 127.0.0.4, 127.0.0.1` | `X-Forwarded-For: 127.0.0.4` | Tidak hadir |
| Permintaan dikirim dengan header XFF dengan beberapa alamat IP klien. | `GET / index.html HTTP/1.1 Host: example.com X-Forwarded-For: 127.0.0.4, 127.0.0.8` | `X-Forwarded-For: 127.0.0.4, 127.0.0.8, 127.0.0.1` | `X-Forwarded-For: 127.0.0.4, 127.0.0.8` | Tidak hadir |

Untuk memodifikasi, mempertahankan, atau menghapus X-Forwarded-For header menggunakan konsol

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.

Untuk memodifikasi, mempertahankan, atau menghapus X-Forwarded-For header menggunakan AWS CLI

Gunakan perintah modify-load-balancer-attributes dengan atribut `routing.http.xff_header_processing.mode`.

## 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 headerdan 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.

# Tag untuk pendengar dan aturan

Tag membantu Anda mengkategorikan pendengar dan aturan Anda dengan cara yang berbeda. Misalnya, Anda dapat menandai sumber daya berdasarkan tujuan, pemilik, atau lingkungan.

Anda dapat menambahkan beberapa tag ke setiap pendengar dan aturan. Kunci tag harus unik untuk setiap pendengar dan aturan. Jika Anda menambahkan tag dengan kunci yang sudah dikaitkan dengan pendengar dan aturan, itu 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.

## Perbarui tag pendengar

Untuk memperbarui tag untuk pendengar menggunakan konsol

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 nama penyeimbang beban yang berisi pendengar yang ingin Anda perbarui, untuk membuka halaman detailnya.
4. Pada tab Listeners and rules, lakukan salah satu hal berikut:

   a. Pilih teks di kolom Protocol:Port untuk membuka halaman detail untuk pendengar.

      Di bagian tab Tanda, pilih Kelola tanda.

   b. Pilih pendengar yang ingin Anda perbarui tag.

      Pilih Kelola pendengar, lalu Kelola tag.

   c. Pilih teks di kolom Tag untuk membuka halaman detail pendengar, pada tab tag.

      Pilih Kelola tanda.

5. Pada halaman Kelola tag, lakukan satu atau beberapa hal berikut:
   a. Untuk memperbarui tag, masukkan nilai baru untukKuncidanNilai.
   b. Untuk menambahkan tag, pilih Tambahkan tag baru dan masukkan nilai untuk Kunci dan Nilai.
   c. Untuk menghapus sebuah tag, pilih Remove di samping tag yang akan dihapus.
6. Setelah selesai memperbarui tag, pilihSimpan perubahan.

Untuk memperbarui tag untuk pendengar menggunakan AWS CLI

Penggunaan perintahPenambahan tagdanHapus tag.

## Perbarui tag aturan

Untuk memperbarui tag untuk aturan menggunakan konsol

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 nama penyeimbang beban yang berisi aturan yang ingin Anda perbarui, untuk membuka halaman detailnya.
4. Pada tab Listeners and rules, pilih teks di kolom Protocol:Port dari listener yang berisi aturan yang ingin Anda perbarui, untuk membuka halaman detail listener
5. Pada halaman detail pendengar, lakukan salah satu hal berikut:
   a. Pilih teks di kolom Tag nama untuk membuka halaman detail aturan.

      Pada halaman detail aturan, pilih Kelola tag.
   b. Pilih teks di kolom Tag untuk aturan yang ingin Anda perbarui.

      Dalam ringkasan tag muncul pilih Kelola tag.
6. Pada halaman Kelola tag, lakukan satu atau beberapa hal berikut:
   a. Untuk memperbarui tag, masukkan nilai baru untukKuncidanNilai.
   b. Untuk menambahkan tag, pilih Tambahkan tag baru dan masukkan nilai untuk Kunci dan Nilai.
   c. Untuk menghapus sebuah tag, pilih Remove di samping tag yang akan dihapus.
7. Setelah selesai memperbarui tag, pilihSimpan perubahan.

Untuk memperbarui tag untuk aturan menggunakan AWS CLI

Penggunaan perintahPenambahan tagdanHapus tag.

# Menghapus listener untuk Application Load Balancer Anda

Anda dapat menghapus listener kapan saja. Saat Anda menghapus penyeimbang beban, semua listenernya akan dihapus.

Untuk menghapus listener menggunakan konsol

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. Ketika diminta konfirmasi, masukkan **`confirm`** lalu pilih Hapus.

Untuk menghapus pendengar menggunakan AWS CLI

Gunakan perintah hapus-listener.

# 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

- Buat grup target
- Pemeriksaan kondisi untuk grup target Anda
- Penyeimbangan beban lintas zona
- Kesehatan kelompok sasaran
- Daftarkan target dengan grup target Anda
- Sesi lengket untuk Application Load Balancer Anda
- Lambda berfungsi sebagai target
- Tag untuk grup target Anda
- Menghapus grup target

## 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

Jika grup target dikonfigurasi dengan protokol HTTPS atau menggunakan pemeriksaan kesehatan HTTPS, sambungan TLS ke target menggunakan pengaturan keamanan dari`ELBSecurityPolicy-2016-08`kebijakan. 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 target tersebut.

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 AWS Direct Connect atau koneksi VPN Site-to-Site.

> **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 Lambda berfungsi sebagai target.

Anda dapat mengonfigurasi Amazon Elastic Container Service (Amazon ECS) sebagai target Application Load Balancer Anda. Untuk informasi selengkapnya, lihat Membuat Application Load Balancer di Panduan Pengguna Amazon Elastic Container Service untuk. AWS Fargate

## 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.

Application Load Balancers mendukung kelompok sasaran IPv4 dan IPv6. Pilihan default adalah IPv4.

Pertimbangan

- Semua alamat IP dalam grup target harus memiliki jenis alamat IP yang sama. Misalnya, Anda tidak dapat mendaftarkan target IPv4 dengan grup target IPv6.

- Kelompok sasaran IPv6 hanya dapat digunakan dengan penyeimbang `dualstack` beban.
- Kelompok target IPv6 mendukung target tipe IP dan Instance.

## 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 dari klien. Load balancer tidak mendukung streaming ke target.

# 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 deregisterasi target menghapus itu dari kelompok target Anda, tetapi tidak mempengaruhi target sebaliknya. Load balancer berhenti routing permintaan ke target segera setelah pendaftaran terbatalkan. 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 dari Application Load Balancer lain di VPC yang sama. Jika Application Load Balancer lainnya ada di VPC yang mengintip ke VPC penyeimbang beban, Anda dapat mendaftarkan alamat IP-nya.
- Anda tidak dapat mendaftarkan instans dengan instans ID jika mereka berada di VPC yang mengintip ke VPC penyeimbang beban (Wilayah yang sama atau Wilayah yang berbeda). Anda dapat mendaftarkan instnas ini dengan alamat IP.

## 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 load balancing menentukan bagaimana load balancer memilih target saat merutekan permintaan. 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` ada`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`. Standarnya adalah `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`. Standarnya 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 load balancer.

`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 zona tersebut sebagai tidak sehat di DNS, sehingga lalu lintas hanya diarahkan ke zona sehat. Nilai yang mungkin adalah`off`, atau bilangan bulat dari 1 ke jumlah maksimum target. Ketika`off`, DNS gagal dinonaktifkan, artinya setiap grup target secara independen berkontribusi pada kegagalan 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 zona sebagai tidak sehat di DNS, sehingga lalu lintas dialihkan hanya ke zona sehat. Nilai yang mungkin adalah`off`, atau bilangan bulat dari 1 ke jumlah target maksimum. Ketika`off`, DNS

gagal dinonaktifkan, artinya setiap grup target secara independen berkontribusi pada kegagalan DNS. Default-nya adalah 1.

`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.

## 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.

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 serupa dalam kompleksitas, target terdaftar serupa dalam kemampuan pemrosesan, atau jika Anda perlu mendistribusikan permintaan secara merata di antara target.

Permintaan paling tidak tertunda

- Algoritme 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 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.

## Memodifikasi algoritma routing dari kelompok target

Anda dapat memodifikasi algoritma routing untuk grup target Anda kapan saja.

Untuk mengubah algoritma routing menggunakan konsol baru

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. Pada halaman detail grup target, pada tab Atribut, pilih Edit.
5. Pada halaman Edit atribut grup target, di bagian konfigurasi Lalu lintas, di bawah Algoritma penyeimbangan beban, pilih Round robin, Permintaan paling tidak beredar, atau Weighted random.
6. Pilih Simpan perubahan.

Untuk memodifikasi algoritma routing menggunakan AWS CLI

Penggunaanmodifikasi-target-kelompok-atributperintah dengan perintah`load_balancing.algorithm.type`atribut.

# 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.

## 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, pertama-tama harus lulus pemeriksaan kesehatan untuk mulai menerima lalu lintas. Setelah target menerima target, 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

ATW terus menerbitkan status deteksi anomali yang dilakukannya pada target. Anda dapat melihat status saat ini kapan saja menggunakan AWS Management Console atau AWS CLI.

Untuk melihat status deteksi anomali menggunakan konsol

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. Pada halaman detail grup target, pilih tab Target.
5. Dalam tabel Target terdaftar, Anda dapat melihat setiap status anomali target di kolom Hasil deteksi anomali.

   Jika tidak ada anomali yang terdeteksi, hasilnya adalah. `normal`

   Jika anomali terdeteksi, hasilnya adalah. `anomalous`

Untuk melihat hasil deteksi anomali menggunakan AWS CLI

Gunakan perintah describe-target-health dengan nilai atribut yang disetel ke. `Include.member.N` `AnomalyDetection`

# Mitigasi anomali

> **Important**
>
> Fungsi mitigasi anomali ATW hanya tersedia saat menggunakan algoritma perutean acak tertimbang.

Mitigasi anomali ATW mengarahkan lalu lintas menjauh dari target anomali secara otomatis, memberi mereka kesempatan untuk pulih.

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.

## Aktifkan mitigasi anomali ATW

Anda dapat mengaktifkan mitigasi anomali kapan saja.

Untuk mengaktifkan mitigasi anomali menggunakan konsol

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. Pada halaman detail grup target, pada tab Atribut, pilih Edit.
5. Pada halaman Edit atribut grup target, di bagian Konfigurasi lalu lintas, di bawah Algoritma penyeimbangan beban, pastikan acak tertimbang dipilih.

   Catatan: Ketika algoritma acak tertimbang awalnya dipilih, deteksi anomali aktif secara default.
6. Di bawah mitigasi anomali, pastikan Aktifkan mitigasi anomali dipilih.

7. Pilih Simpan perubahan.

Untuk mengaktifkan mitigasi anomali menggunakan AWS CLI

Penggunaanmodifikasi-target-kelompok-atributperintah dengan perintah`load_balancing.algorithm.anomaly_mitigation`atribut.

Status mitigasi anomali

Setiap kali ATW melakukan mitigasi pada target, Anda dapat melihat status saat ini kapan saja menggunakan atau. AWS Management Console AWS CLI

Untuk melihat status mitigasi anomali menggunakan konsol

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. Pada halaman detail grup target, pilih tab Target.
5. Dalam tabel Target terdaftar, Anda dapat melihat setiap status mitigasi anomali target di kolom Mitigasi berlaku.

   Jika mitigasi tidak sedang berlangsung, statusnya. `yes`

   Jika mitigasi sedang berlangsung, statusnya. no

Untuk melihat status mitigasi anomali menggunakan AWS CLI

Gunakan perintah describe-target-health dengan nilai atribut yang disetel ke. `Include.member.N AnomalyDetection`

# 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.

Untuk memperbarui nilai penundaan deregistrasi menggunakan konsol

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. Pada tabDetail grup, di bagian atribut, pilihEdit.
5. Pada laman Edit atribut, mengubah nilaiPenundaan Deregistrasiseperlunya.
6. Pilih Simpan perubahan.

Untuk memperbarui nilai penundaan deregistrasi menggunakan AWS CLI

Penggunaan perintah modifikasi-target-kelompok-atribut dengan perintah`deregistration_delay.timeout_seconds`atribut.

# 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

- 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 mendaftar 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 mendaftar 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.

Untuk memperbarui nilai durasi mulai lambat menggunakan konsol

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. Pada tabDetail kelompok, di bagianatribut, pilihEdit.
5. Pada lamanEdit atribut, ubah nilaiDurasi mulai lambatseperlunya. Untuk menonaktifkan mode mulai lambat, atur durasi ke 0.
6. Pilih Simpan perubahan.

Untuk memperbarui nilai durasi mulai lambat menggunakan AWS CLI

Penggunaan perintahmodifikasi-target-kelompok-atribut dengan perintah`slow_start.duration_seconds`atribut.

# Buat grup target

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 mendaftar 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 Peraturan listener. 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 grup target Anda. Anda juga dapat mengubah pengaturan pemeriksaan kesehatan untuk grup target Anda. Untuk informasi selengkapnya, lihat Memodifikasi pengaturan pemeriksaan kondisi dari grup target.

Untuk membuat grup target menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. PilihBuat grup target.
4. UntukPilih jenis targetPilihInstansuntuk mendaftarkan target berdasarkan contoh ID,Alamat IPuntuk mendaftarkan target berdasarkan alamat IP, atauFungsi Lambdauntuk mendaftarkan fungsi Lambda sebagai target.
5. UntukNama 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)ProtokoldanPort, memodifikasi nilai default yang diperlukan.
7. Jika jenis targetnya 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. UntukVPC, pilih Virtual Private Cloud (VPC). Perhatikan bahwa untuk jenis target alamat IP, VPC yang tersedia untuk dipilih adalah yang mendukung jenis alamat IP yang Anda pilih pada langkah sebelumnya.
9. (Opsional)Versi protokol, mengubah nilai default yang diperlukan.
10. (Opsional) pada bagianPemeriksaan Health, ubah pengaturan default sesuai kebutuhan.
11. Jika jenis target adalahFungsi Lambda, Anda dapat mengaktifkan pemeriksaan kesehatan dengan memilihMengaktifkandi bagianPemeriksaan Health.
12. (Opsional) Tambahkan satu atau lebih tag sebagai berikut:
    a. Perluas bagianTag.
    b. PilihTambahkan tanda.
    c. Masukkan kunci dan nilai untuk tanda tersebut.
13. Pilih Selanjutnya.
14. (Opsional) Tambahkan satu atau lebih target sebagai berikut:
    - Jika jenis target adalahInstansPilih satu atau beberapa instans, masukkan satu atau beberapa port, lalu pilihSertakan sebagai tertunda di bawah ini.

      Catatan: Instance harus memiliki alamat IPv6 primer yang ditetapkan untuk didaftarkan dengan grup target IPv6.
    - 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.
15. PilihBuat grup target.
16. (Opsional) Anda dapat menentukan kelompok target dalam aturan pendengar. Untuk informasi selengkapnya, lihatAturan pendengar.

Untuk membuat grup target menggunakan AWS CLI

Penggunaan perintahmembuat-target-kelompok untuk membuat grup target,Penambahan tagperintah untuk menandai kelompok target Anda, dan perintahRegister-targetuntuk menambahkan target.

# Pemeriksaan kondisi untuk grup target Anda

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.

## 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 HealthCheckIntervalSecondsdetik, 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 UnhealthyThresholdCountberturut-turut, penyeimbang beban mengeluarkan target dari layanan. Ketika pemeriksaan kesehatan melebihi keberhasilan HealthyThresholdCountberturut-turut, penyeimbang beban menempatkan target kembali dalam layanan.

| Pengaturan | Deskripsi |
| --- | --- |
| HealthCheckProtocol | Protokol yang digunakan penyeimbang beban saat melakukan pemeriksaan kondisi pada target. Protokol yang mungkin adalah HTTP dan HTTPS. Defaultnya adalah protokol HTTP.<br><br>Protokol ini menggunakan metode HTTP GET untuk mengirim permintaan pemeriksaan kesehatan. |
| HealthCheckPort | 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. |
| HealthCheckPath | Tujuan pemeriksaan kondisi pada target.<br><br>Jika versi protokol adalah HTTP/1.1 atau HTTP/2, tentukan URI yang valid (/PATH?query). Defaultnya adalah /.<br><br>Jika versi protokol adalah gRPC, tentukan jalur metode pemeriksaan kondisi kustom dengan format `/package.service/method` . Default-nya adalah `/AWS.ALB/healthcheck` . |
| HealthCheckTimeoutSeconds | 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`. |
| HealthCheckIntervalSeconds | Perkiraan jumlah waktu, dalam hitungan detik, antara pemeriksaan kondisi dari target individu. |

| Pengaturan | Deskripsi |
| --- | --- |
| | Rentangnya adalah 5-300 detik. Defaultny a adalah 30 detik jika jenis target adalah `instance` atau `ip` dan 35 detik jika jenis target adalah `lambda`. |
| HealthyThresholdCount | Jumlah pemeriksaan kondisi yang berhasil berturut-turut diperlukan sebelum menganggap target yang tidak sehat memiliki kondisi sehat. Rentangnya adalah 2–10. Defaultnya adalah 5. |
| UnhealthyThresholdCount | Jumlah pemeriksaan kondisi yang gagal berturut-turut diperlukan sebelum mengangga p target yang tidak memiliki kondisi sehat. Rentangnya adalah 2–10. Defaultnya adalah 2. |
| Pencocokan | Kode yang digunakan saat memeriksa respons yang berhasil dari target. Ini disebut Kode berhasil pada konsol.<br><br>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.<br><br>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. |

## 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 |
| --- | --- |
| `initial` | Penyeimbang beban sedang dalam proses mendaftarkan target atau melakukan pemeriksaan kondisi awal pada target.<br><br>Kode alasan terkait: `Elb.RegistrationInProgress` \| `Elb.InitialHealthChecking` |
| `healthy` | Targetnya sehat.<br><br>Kode alasan terkait: Tidak ada |
| `unhealthy` | Target tidak merespons pemeriksaan kondisi atau gagal dalam pemeriksaan kondisi.<br><br>Kode alasan terkait: `Target.ResponseCodeMismatch` \| `Target.Timeout` \| `Target.FailedHealthChecks` \| `Elb.InternalError` |
| `unused` | 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.<br><br>Kode alasan terkait: `Target.NotRegistered` \| `Target.NotInUse` \| `Target.InvalidState` \| `Target.IpUnusable` |
| `draining` | Target membatalkan pendaftaran dan pengosongan koneksi sedang dalam proses.<br><br>Kode alasan terkait: `Target.DeregistrationInProgress` |

| Nilai | Deskripsi |
| --- | --- |
| `unavailable` | Pemeriksaan kondisi dinonaktifkan untuk grup target.<br><br>Kode alasan terkait: `Target.HealthCheck Disabled` |

## 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.RegistrationIn Progress` | Pendaftaran target sedang berlangsung |
| `Target.Deregistrat ionInProgress` | Pembatalan pendaftaran target sedang berlangsung |
| `Target.FailedHealthChecks` | Pemeriksaan kondisi gagal |
| `Target.HealthCheck Disabled` | Pemeriksaan kondisi dinonaktifkan |
| `Target.InvalidState` | Target dalam keadaan berhenti<br><br>Target dalam keadaan dihentikan<br><br>Target berada dalam keadaan dihentikan atau berhenti<br><br>Target dalam keadaan tidak valid |

| Kode alasan | Deskripsi |
| --- | --- |
| `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<br><br>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 |

## Periksa kondisi target Anda

Anda dapat memeriksa status kondisi target yang terdaftar dengan kelompok target Anda.

Untuk memeriksa kesehatan target Anda menggunakan konsol

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 halaman detailnya.
4. Pada tab Target, kolom Status menunjukkan status setiap target.
5. Jika status adalah nilai selain `Healthy`, kolom Detail status berisi informasi lebih lanjut. Untuk bantuan dengan kegagalan pemeriksaan kesehatan, lihat Pemecahan masalah.

Untuk memeriksa kesehatan target Anda menggunakan AWS CLI

Gunakan perintah describe-target-health. Keluaran dari perintah ini berisi status kondisi target. Jika status adalah nilai selain `Healthy`, output juga termasuk kode alasan.

Untuk menerima pemberitahuan email tentang target yang tidak sehat

Gunakan CloudWatch alarm untuk memicu fungsi Lambda untuk mengirim detail tentang target yang tidak sehat. Untuk step-by-step petunjuk, lihat posting blog berikut: Mengidentifikasi target penyeimbang beban Anda yang tidak sehat.

## Memodifikasi pengaturan pemeriksaan kondisi dari grup target

Anda dapat mengubah pengaturan pemeriksaan kondisi untuk grup target kapan saja.

Untuk mengubah pengaturan pemeriksaan kesehatan grup target menggunakan konsol

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 Detail grup, di bagian Pengaturan pemeriksaan kondisi, pilih Edit.
5. Pada halaman Mengedit pengaturan pemeriksaan kondisi, ubah pengaturan sesuai kebutuhan, lalu pilih Simpan perubahan.

Untuk memodifikasi pengaturan pemeriksaan kesehatan dari kelompok target menggunakan AWS CLI

Gunakan perintah modify-target-group.

# Penyeimbangan beban lintas zona

Node untuk Load Balancer Anda mendistribusikan permintaan dari klien ke target yang telah terdaftar. Ketika load balancing lintas zona aktif, setiap node Load Balancer mendistribusikan lalu lintas di seluruh target yang terdaftar di semua Availability Zone yang telah terdaftar. Ketika load balancing lintas zona dinonaktifkan, setiap node Load Balancer mendistribusikan lalu lintas hanya di target yang terdaftar di Availability Zonenya. Ini bisa jadi jika domain kegagalan zonal lebih disukai daripada regional, memastikan bahwa zona sehat tidak terpengaruh oleh zona yang tidak sehat, atau untuk peningkatan latensi secara keseluruhan.

Dengan Application Load Balancer, Load Balancer lintas zona selalu dinyalakan pada tingkat penyeimbangan beban, Load Balancer lintas zona selalu dinonaktifkan. Untuk grup target, defaultnya adalah menggunakan pengaturan load balancer, tetapi Anda dapat mengganti default dengan secara eksplisit mematikan load balancing lintas zona di tingkat grup target.

Pertimbangan-pertimbangan

- Kelekatan target tidak didukung ketika load balancing lintas zona dinonaktifkan.
- Lambda berfungsi sebagai target tidak didukung ketika load balancing lintas zona dinonaktifkan.
- Mencoba mematikan penyeimbangan beban lintas zona melalui`ModifyTargetGroupAttributes` API jika ada target yang`AvailabilityZone` disetel parameter untuk`all` menghasilkan kesalahan.
- Saat mendaftarkan target,`AvailabilityZone` parameter diperlukan. Nilai spesifik Availability Zone hanya diperbolehkan ketika load balancing lintas zona dinonaktifkan. 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 grup target. Jika Anda tidak dapat merencanakan kapasitas yang cukup di semua Availability Zone yang berpartisipasi, kami menyarankan agar Anda tetap menyeimbangkan beban lintas zona.
- Saat mengonfigurasi Application Load Balancer dengan beberapa grup target, pastikan semua grup target berpartisipasi dalam Availability Zone yang sama, dalam Wilayah yang dikonfigurasi. Hal ini untuk menghindari Availability Zone menjadi kosong saat load balancing lintas zona dimatikan, 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.
- Ada kejadian di mana kelompok target dengan penyeimbangan beban lintas zona dimatikan memiliki kapasitas target yang cukup direncanakan 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 node load balancer dihapus dari DNS. Setelah grup target memiliki setidaknya satu target sehat, alamat IP dikembalikan ke DNS.

## Matikan penyeimbangan beban lintas zona

Anda dapat menonaktifkan penyeimbangan beban lintas zona kapan saja.

Untuk menonaktifkan penyeimbangan beban lintas zona menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Pada panel navigasi, di bawah Load Balancing, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan detailnya.
4. Pada tab Atribut, pilih Edit.
5. Pada halaman Edit atribut grup target, pilih Nonaktif untuk penyeimbangan beban lintas zona.
6. Pilih Save changes (Simpan perubahan).

Untuk mematikan penyeimbangan beban lintas zona menggunakanAWS CLI

Gunakan modify-target-group-attributesperintah dan aturload_balancing.cross_zone.enabled atributnyafalse.

```
aws elbv2 modify-target-group-attributes --target-group-arn my-targetgroup-arn --
attributes Key=load_balancing.cross_zone.enabled,Value=false
```

Berikut contoh responsnya:

```
{
    "Attributes": [
        {
            "Key": "load_balancing.cross_zone.enabled",
            "Value": "false"
        },
    ]
}
```

## Aktifkan penyeimbangan beban lintas zona

Anda dapat mengaktifkan penyeimbangan beban lintas zona kapan saja. Pengaturan penyeimbangan beban lintas zona pada tingkat grup target menimpa pengaturan pada tingkat penyeimbang beban.

Untuk mengaktifkan penyeimbangan beban lintas zona menggunakan konsol

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

2. Pada panel navigasi, di bawah Load Balancing, pilih Grup Target.
3. Pilih nama grup target untuk menampilkan detailnya.
4. Pada tab Atribut, pilih Edit.
5. Pada halaman Edit atribut grup target, pilih Aktif untuk penyeimbangan beban lintas zona.
6. Pilih Save changes (Simpan perubahan).

Untuk mengaktifkan penyeimbangan beban lintas zona menggunakanAWS CLI

Gunakan modify-target-group-attributesperintah dan aturload_balancing.cross_zone.enabled atributnyatrue.

```
aws elbv2 modify-target-group-attributes --target-group-arn my-targetgroup-arn --
attributes Key=load_balancing.cross_zone.enabled,Value=true
```

Berikut contoh responsnya:

```
{
    "Attributes": [
        {
            "Key": "load_balancing.cross_zone.enabled",
            "Value": "true"
        },
    ]
}
```

# 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.

## 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

- 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.
- Dengan failover DNS, kami menghapus alamat IP untuk 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).
- Ketika failover DNS terjadi, ini 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, 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 target sehat yang cukup yang dapat menyebabkan kegagalan DNS, seperti kesehatan zona.

## Pemantauan

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.

## Ubah pengaturan kesehatan kelompok sasaran

Anda dapat mengubah pengaturan kesehatan grup target untuk grup target Anda sebagai berikut.

Untuk mengubah pengaturan kesehatan grup target menggunakan konsol

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.

Untuk memodifikasi pengaturan kesehatan kelompok target menggunakan AWS CLI

Penggunaan perintah ubah-atribut-grup-target. 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 arn:aws:elasticloadbalancing:region:123456789012:targetgroup/my-
targets/73e2d6bc24d8a067 \
--attributes
 Key=target_group_health.dns_failover.minimum_healthy_targets.percentage,Value=50 \

 Key=target_group_health.unhealthy_state_routing.minimum_healthy_targets.percentage,Value=50
```

## 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.

Menggunakan evaluasi kesehatan target

- Ketika mengevaluasi kesehatan target diatur ke `Yes` catatan alias untuk Application Load Balancer, Route 53 mengevaluasi kesehatan sumber daya yang ditentukan oleh nilai. `alias target` Untuk Application Load Balancer, Route 53 menggunakan pemeriksaan kesehatan kelompok sasaran yang terkait dengan load balancer.
- Ketika semua kelompok sasaran dalam Application Load Balancer sehat, Route 53 menandai catatan alias sebagai sehat. Jika kelompok sasaran berisi setidaknya satu target sehat, pemeriksaan kesehatan kelompok sasaran lolos. Route 53 kemudian mengembalikan catatan sesuai dengan kebijakan perutean Anda. Jika kebijakan routing failover digunakan, Route 53 mengembalikan catatan utama.

- Jika salah satu kelompok sasaran dalam Application Load Balancer tidak sehat, catatan alias gagal dalam pemeriksaan kesehatan Route 53 (fail-open). Jika menggunakan evaluasi kesehatan target, ini akan gagal dalam kebijakan perutean failover.
- Jika semua kelompok target dalam Application Load Balancer kosong (tidak ada target), maka Route 53 menganggap catatan tidak sehat (fail-open). Jika menggunakan evaluasi kesehatan target, ini akan gagal dalam kebijakan perutean failover.

Untuk informasi selengkapnya, lihat Mengonfigurasi failover DNS di Panduan Pengembang Amazon Route 53.

## Daftarkan target dengan grup target 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 mendaftar ID contoh, 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 deregristrasi target, load balancer menunggu hingga permintaan dalam penerbangan selesai. Hal ini dikenal sebagaiPengurasan 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.

## 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 |
| --- | --- | --- |
| *kelompok keamanan penyeimbang beban* | *pendengar contoh* | Izinkan lalu lintas dari penyeimbang beban pada port pendengar instance |
| *kelompok keamanan penyeimbang beban* | *pemeriksaan kesehatan* | 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.

## 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.

## Mendaftar atau membatalkan pendaftaran target

Jenis target grup target Anda menentukan bagaimana Anda mendaftarkan target dengan kelompok target tersebut. Untuk informasi selengkapnya, lihat Tipe target.

Daftar Isi



## Register atau target deregister berdasarkan ID instance

> **Note**
>
> Saat mendaftarkan target dengan ID instance untuk grup target IPv6, target harus memiliki alamat IPv6 primer yang ditetapkan. Untuk mempelajari lebih lanjut, lihat alamat IPv6 di Panduan Pengguna Amazon EC2

Instans harus berada di Virtual Private Cloud (VPC) yang Anda tentukan untuk grup target. Contoh juga harus dalam keadaan`running` saat Anda mendaftarkannya.

Untuk mendaftarkan atau membatalkan pendaftaran target berdasarkan ID instans menggunakan konsol

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 tabTarget.
5. Untuk mendaftarkan contoh, pilihTarget daftar. Pilih satu atau beberapa instans, masukkan port default sesuai kebutuhan, lalu pilihSertakan sebagai tertunda di bawah ini. Setelah selesai menambah instans, pilihMendaftarkan target tertunda.

   Catatan:

   - Instance harus memiliki alamat IPv6 primer yang ditetapkan untuk didaftarkan dengan grup target IPv6.
   - AWS GovCloud (US) Region s tidak mendukung penetapan alamat IPv6 utama menggunakan konsol. Anda harus menggunakan API untuk menetapkan alamat IPv6 utama di s. AWS GovCloud (US) Region

6. Untuk membatalkan pendaftaran contoh, pilih contoh dan kemudian pilihDeregister.

## Mendaftar atau membatalkan pendaftaran target berdasarkan alamat IP

Target IPv4

Alamat IP yang Anda daftarkan harus 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)

Anda tidak dapat mendaftarkan alamat IP dari Application Load Balancer lain di VPC yang sama. Jika Application Load Balancer lainnya ada di VPC yang mengintip ke VPC penyeimbang beban, Anda dapat mendaftarkan alamat IP-nya.

Target IPv6

- Alamat IP yang Anda daftarkan harus berada di dalam blok VPC CIDR atau dalam blok CIDR VPC yang dipeer.

Untuk mendaftarkan atau membatalkan pendaftaran target berdasarkan alamat IP menggunakan konsol

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 tabTarget.
5. Untuk mendaftarkan alamat IP, pilihTarget daftar. Untuk setiap alamat IP, pilih rangkaian, masukkan alamat IP dan port, dan pilihSertakan sebagai tertunda di bawah ini. Setelah selesai menentukan alamat, pilihMendaftarkan target tertunda.
6. Untuk membatalkan pendaftaran alamat IP, pilih alamat IP, lalu pilihDeregister. Jika Anda memiliki banyak alamat IP terdaftar, menambahkan filter atau mengubah urutan pengurutan mungkin akan membantu Anda.

### Mendaftar atau membatalkan pendaftaran fungsi Lambda

Anda dapat mendaftarkan fungsi Lambda tunggal dengan masing-masing grup target. Elastic Load Balancing harus memiliki izin untuk memacu 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, lebih baik untuk membuat kelompok target baru sebagai gantinya. Untuk informasi selengkapnya, lihat Lambda berfungsi sebagai target.

Untuk mendaftarkan atau membatalkan pendaftaran fungsi Lambda menggunakan konsol

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 tabTarget.
5. Jika tidak ada fungsi Lambda terdaftar, pilihPendaftaran. Pilih fungsi LambdaPendaftaran.
6. Untuk membatalkan pendaftaran fungsi Lambda, pilihDeregister. Ketika diminta konfirmasi, pilih Deregister.

### Mendaftar atau membatalkan pendaftaran target menggunakan AWS CLI

Penggunaan perintahRegister-target untuk menambahkan target dan perintah Target deregister untuk menghapus target.

## Sesi lengket untuk Application Load Balancer Anda

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 load balancer yang dihasilkan cookie.

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 beban HTTP/HTTPS.
- Setidaknya satu contoh sehat di setiap Availability Zone.

Pertimbangan

- Sesi lengket tidak didukung jika penyeimbangan beban lintas zona dinonaktifkan. Mencoba mengaktifkan sesi lengket saat penyeimbangan beban lintas zona dinonaktifkan akan 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.
- Stickiness berbasis aplikasi tidak bekerja dengan kelompok target tertimbang.
- Jika Anda memilikiTindakan ke depandengan 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.

## 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 hadir tetapi tidak dapat diterjemahkan, atau jika mengacu pada target yang deregistered 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.

Cara mengaktifkan kelekatan berbasis durasi menggunakan konsol

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. Pada tabDetail kelompok, di bagianatribut, pilihEdit.
5. Di laman Edit Perilaku , lakukan hal berikut:
   a. PilihKelekatan.
   b. UntukJenis kelekatanPilihCookies yang dihasilkan load balancer.
   c. UntukDurasi kelekatan, tentukan nilai antara 1 detik dan 7 hari.
   d. Pilih Simpan perubahan.

Untuk mengaktifkan kelengketan berbasis durasi menggunakan AWS CLI

Penggunaan perintahmodifikasi-target-kelompok-atributdengan atributstickiness.enabledanstickiness.lb_cookie.duration_seconds.

Gunakan perintah berikut untuk mengaktifkan kelekatan berbasis durasi.

```
aws elbv2 modify-target-group-attributes --target-group-arn ARN --attributes
 Key=stickiness.enabled,Value=true
 Key=stickiness.lb_cookie.duration_seconds,Value=time-in-seconds
```

Output Anda harus serupa dengan berikut ini.

```
    {
       "Attributes": [
          ...
           {
                "Key": "stickiness.enabled",
                "Value": "true"
             },
             {
                 "Key": "stickiness.lb_cookie.duration_seconds",
                 "Value": "86500"

             },
          ...
        ]
     }
```

# 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 dengan ukuran 4K, load balancer memecah cookie aplikasi yang lebih besar dari 4K ke dalam 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.

Cara mengaktifkan kelekatan berbasis aplikasi menggunakan konsol

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. Pada tabDetail kelompok, di bagianatribut, pilihEdit.
5. Di laman Edit Perilaku , lakukan hal berikut:
   a. PilihKelekatan.
   b. UntukJenis kelekatanPilihCookie berbasis aplikasi.
   c. UntukDurasi lengket, tentukan nilai antara 1 detik dan 7 hari.
   d. UntukNama 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.
   e. Pilih Simpan perubahan.

Untuk mengaktifkan kelengketan berbasis aplikasi menggunakan AWS CLI

Penggunaan perintahmodifikasi-target-kelompok-atribut dengan atribut berikut:

- `stickiness.enabled`
- `stickiness.type`
- `stickiness.app_cookie.cookie_name`
- `stickiness.app_cookie.duration_seconds`

Gunakan perintah berikut untuk mengaktifkan kelekatan berbasis aplikasi.

```
aws elbv2 modify-target-group-attributes --target-group-arn ARN --attributes
 Key=stickiness.enabled,Value=true Key=stickiness.type,Value=app_cookie
 Key=stickiness.app_cookie.cookie_name,Value=my-cookie-name
 Key=stickiness.app_cookie.duration_seconds,Value=time-in-seconds
```

Output Anda harus serupa dengan berikut ini.

```
      {
           "Attributes": [
               ...
               {
                   "Key": "stickiness.enabled",
                   "Value": "true"
               },
               {
                   "Key": "stickiness.app_cookie.cookie_name",
                   "Value": "MyCookie"
               },
               {
                   "Key": "stickiness.type",
                   "Value": "app_cookie"
               },
               {
                   "Key": "stickiness.app_cookie.duration_seconds",
                   "Value": "86500"
               },
               ...
           ]
       }
```

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 akan mencegah klien mengirim cookie ke Application Load Balancer, yang akan memulai ulang proses pembentukan kelekatan.

- Tetapkan durasi yang sangat singkat pada konfigurasi kelekatan berbasis aplikasi load balancer, misalnya, 1 detik. Ini memaksa Application Load Balancer untuk membangun kembali kelekatan meskipun cookie yang ditetapkan oleh target belum kedaluwarsa.

# Lambda berfungsi sebagai target

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.

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, lihatBatas 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



Untuk demo, lihatTarget 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 AWS Management Console, 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 deputi 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, lihatAWS Lambda versi fungsi dan aliasdanPergeseran lalu lintas menggunakan aliasdiAWS Lambda Panduan 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 dipekirakan. Untuk informasi tentang nilai timeout default dan cara mengubahnya, lihatDasar AWS Lambda Konfigurasi fungsi. Untuk informasi tentang nilai timeout maksimum yang dapat Anda konfigurasikan, lihatAWS Lambda Batasan.

## 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.

Untuk membuat grup target dan mendaftarkan fungsi Lambda menggunakan konsol

1. Buka konsol Amazon EC2 di https://console.aws.amazon.com/ec2/.
2. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
3. PilihBuat grup target.
4. UntukPilih jenis targetPilihFungsi Lambda.
5. UntukNama kelompok target, ketik nama untuk grup target.
6. (Opsional) Untuk mengaktifkan pemeriksaan kesehatan, pilihMengaktifkandiPemeriksaan HealthBagian.
7. (Opsional) Tambahkan satu atau lebih tanda sebagai berikut:
   a. Perluas bagianTag.
   b. PilihTambahkan tanda.
   c. Masukkan kunci dan nilai untuk tanda tersebut.
8. Pilih Selanjutnya.
9. Tentukan fungsi Lambda tunggal atau hilangkan langkah ini dan kemudian tentukan fungsi Lambda.
10. PilihBuat grup target.

Untuk membuat grup target dan mendaftarkan fungsi Lambda menggunakan AWS CLI

Gunakan perintahmembuat-target-kelompokdanRegister-target.

## 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 memasangkan `isBase64Encoded`ketrue.

Jika`content-encoding`header tidak hadir, encoding Base64 tergantung pada jenis konten. Untuk jenis berikut, load balancer mengirimkan tubuh seperti apa adanya dan memasang`isBase64Encoded`kefalse: teks/*, aplikasi/json, aplikasi/javascript, dan aplikasi/xml. Jika tidak, load balancer Base64 mengkodekan tubuh dan memasangkan`isBase64Encoded`ketrue.

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 respons dari fungsi Lambda berbasis nodejs.

```
{
    "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 Balancer, lihat[aplikasi-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

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 target`lambda`.

Untuk mengaktifkan header multi-nilai menggunakan konsol

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. Pada tabDetail kelompok, di bagianatribut, pilihEdit.
5. Pilih atau hapusHeader nilai ganda.
6. Pilih Simpan perubahan.

Untuk mengaktifkan header multi-nilai menggunakan AWS CLI

Penggunaan perintah modifikasi-target-kelompok-atribut dengan `lambda.multi_value_headers.enabled`atribut.

## 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 agen 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
}
```

Untuk mengaktifkan pemeriksaan kesehatan untuk grup target menggunakan konsol

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 Detail kelompok, di bagianPengaturan pemeriksaan Health, pilihEdit.
5. UntukPemeriksaan KesehatanPilihAktifkan.
6. Pilih Simpan perubahan.

Untuk mengaktifkan pemeriksaan kesehatan untuk kelompok sasaran menggunakan AWS CLI

Penggunaan perintahmodifikasi-target-kelompok dengan pilihan`--health-check-enabled`.

## Deregristrasi 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 kelompok target baru, dan memperbarui aturan pendengar untuk menggunakan kelompok target baru bukan yang sudah ada.

Untuk membatalkan pendaftaran fungsi Lambda menggunakan konsol

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, pilihDeregister.
5. Ketika diminta konfirmasi, pilih Akhiri.

Untuk membatalkan pendaftaran fungsi Lambda menggunakan AWS CLI

Gunakan perintah Target deregister.

# Tag untuk grup target 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.

Untuk memperbarui tag untuk grup target menggunakan konsol

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 untukKuncidanNilai.
   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. Setelah selesai memperbarui tag, pilihSimpan perubahan.

Untuk memperbarui tag untuk grup target menggunakan AWS CLI

Penggunaan perintahPenambahan tagdanHapus tag.

# Menghapus grup target

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.

Untuk menghapus grup target menggunakan konsol

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 grup target dan pilih Tindakan, Hapus.
4. Saat diminta konfirmasi, pilih Ya, hapus.

Untuk menghapus grup target menggunakan AWS CLI

Gunakan perintah hapus target grup .

# 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.

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 Pengelogan panggilan API untuk Application Load Balancer Anda menggunakan AWS CloudTrail.

# CloudWatch metrik untuk Application Load Balancer

Elastic Load Balancing menerbitkan titik data ke Amazon CloudWatch untuk penyeimbang beban dan target Anda. CloudWatchmemungkinkan 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.

Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

Daftar Isi



## Metrik Application Load Balancer

- Fungsi Lambda
- Otentikasi pengguna

Namespace `AWS/ApplicationELB` menyertakan metrik berikut untuk penyeimbang beban.

| Metrik | Deskripsi |
| --- | --- |
| `ActiveConnectionCount` | Jumlah total koneksi TCP bersamaan yang aktif dari klien ke penyeimbang beban dan dari penyeimbang beban ke target.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah `Sum`.<br><br>Dimensi<br><br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |
| `AnomalousHostCount` | Jumlah host terdeteksi dengan anomali.<br><br>Reporting criteria: Selalu dilaporkan<br><br>Statistics: Statistik yang paling berguna adalah `Average`, `Minimum`, dan `Maximum`.<br><br>Dimensi<br><br>• `TargetGroup` , `LoadBalancer`<br>• `TargetGroup` , `AvailabilityZone` , `LoadBalancer` |
| `ClientTLSNegotiationErrorCount` | 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.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah `Sum`. |

| Metrik | Deskripsi |
| --- | --- |
| | Dimensi<br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |
| `ConsumedLCUs` | Jumlah unit kapasitas penyeimbang beban (LCU) yang digunakan oleh penyeimbang beban Anda. Anda membayar untuk jumlah LCU yang digunakan per jam. Untuk informasi selengkapnya, lihat Harga Elastic Load Balancing.<br>Reporting criteria: Selalu dilaporkan<br>Statistics: Semua<br>Dimensi<br>• `LoadBalancer` |
| `DesyncMitigationMode_NonCompliant_Request_Count` | Jumlah permintaan yang tidak sesuai dengan RFC 7230.<br>Reporting criteria: Ada nilai bukan nol<br>Statistics: Statistik yang paling berguna adalah `Sum`.<br>Dimensi<br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |

| Metrik | Deskripsi |
| --- | --- |
| DroppedInvalidHeaderRequestCount | 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 routing.http.drop_invalid_header_fields.enabled diatur ke true.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Semua<br><br>Dimensi<br><br>• AvailabilityZone , LoadBalancer |
| MitigatedHostCount | Jumlah target di bawah mitigasi.<br><br>Reporting criteria: Selalu dilaporkan<br><br>Statistics: Statistik yang paling berguna adalah Average, Minimum, dan Maximum.<br><br>Dimensi<br><br>• TargetGroup , LoadBalancer<br>• TargetGroup , AvailabilityZone , LoadBalancer |

| Metrik | Deskripsi |
| --- | --- |
| ForwardedInvalidHeaderRequestCount | 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 routing.http.drop_invalid_header_fields.enabled diatur ke false.<br><br>Reporting criteria: Selalu dilaporkan<br><br>Statistics: Semua<br><br>Dimensi<br><br>• AvailabilityZone , LoadBalancer |
| GrpcRequestCount | Jumlah permintaan gRPC yang diproses melalui IPv4 dan IPv6.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah Sum. Minimum, Maximum, dan Average semua kembali 1.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |
| HTTP_Fixed_Response_Count | Jumlah tindakan respons tetap yang berhasil.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |

| Metrik | Deskripsi |
| --- | --- |
| HTTP_Redirect_Count | Jumlah tindakan pengalihan yang berhasil.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |
| HTTP_Redirect_Url_Limit_Exceeded_Count | Jumlah tindakan pengalihan yang tidak dapat diselesaikan karena URL di header lokasi respons lebih besar dari 8K.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |
| HTTPCode_ELB_3XX_Count | Jumlah kode pengalihan HTTP 3XX yang berasal dari penyeimbang beban. Jumlah ini tidak termasuk kode respons yang dihasilkan oleh target.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |

| Metrik | Deskripsi |
| --- | --- |
| `HTTPCode_ELB_4XX_Count` | Jumlah kode kesalahan klien HTTP 4XX yang berasal dari penyeimbang beban. Jumlah ini tidak termasuk kode respons yang dihasilkan oleh target.<br><br>Kesalahan klien dihasilkan saat permintaan salah format atau tidak lengkap. Permintaan ini tidak diterima oleh target, selain jika penyeimbang beban mengembalikan kode kesalahan HTTP 460. Jumlah ini tidak termasuk kode respons apa pun yang dihasilkan oleh target.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah `Sum`. `Minimum`, `Maximum`, dan `Average` semua kembali 1.<br><br>Dimensi<br><br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |
| `HTTPCode_ELB_5XX_Count` | Jumlah kode kesalahan server HTTP 5XX yang berasal dari penyeimbang beban. Jumlah ini tidak termasuk kode respons apa pun yang dihasilkan oleh target.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah `Sum`. `Minimum`, `Maximum`, dan `Average` semua kembali 1.<br><br>Dimensi<br><br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |

| Metrik | Deskripsi |
| --- | --- |
| `HTTPCode_ELB_500_Count` | Jumlah kode kesalahan HTTP 500 yang berasal dari penyeimbang beban.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah `Sum`.<br><br>Dimensi<br><br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |
| `HTTPCode_ELB_502_Count` | Jumlah kode kesalahan HTTP 502 yang berasal dari penyeimbang beban.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah `Sum`.<br><br>Dimensi<br><br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |
| `HTTPCode_ELB_503_Count` | Jumlah kode kesalahan HTTP 503 yang berasal dari penyeimbang beban.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah `Sum`.<br><br>Dimensi<br><br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |

| Metrik | Deskripsi |
| --- | --- |
| HTTPCode_ELB_504_Count | Jumlah kode kesalahan HTTP 504 yang berasal dari penyeimbang beban.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |
| IPv6ProcessedBytes | Jumlah total byte yang diproses oleh penyeimbang beban melalui IPv6. Hitungan ini termasuk dalam ProcessedBytes .<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |
| IPv6RequestCount | Jumlah permintaan IPv6 yang diterima oleh penyeimbang beban.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah Sum. Minimum, Maximum, dan Average semua kembali 1.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |

| Metrik | Deskripsi |
| --- | --- |
| NewConnectionCount | Jumlah total koneksi TCP baru yang dibuat dari klien ke penyeimbang beban dan dari penyeimbang beban ke target.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |
| NonStickyRequestCount | 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.<br><br>Reporting criteria: Kelekatan diaktifkan pada grup target.<br><br>Statistics: Satu-satunya statistik yang bermakna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |

| Metrik | Deskripsi |
| --- | --- |
| ProcessedBytes | Jumlah total byte yang diproses oleh penyeimbang beban melalui IPv4 dan IPv6 (header HTTP dan payload HTTP). Jumlah ini mencakup lalu lintas ke dan dari klien dan fungsi Lambda serta lalu lintas dari Penyedia Identitas (IdP) jika autentikasi pengguna diaktifkan.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |
| RejectedConnectionCount | Jumlah koneksi yang ditolak karena penyeimbang beban telah mencapai jumlah koneksi maksimumnya.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |

| Metrik | Deskripsi |
| --- | --- |
| RequestCount | Jumlah permintaan yang diproses melalui 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.<br><br>Reporting criteria: Selalu dilaporkan<br><br>Statistics: Statistik yang paling berguna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• LoadBalancer , AvailabilityZone<br>• LoadBalancer , TargetGroup<br>• LoadBalancer , AvailabilityZone , TargetGroup |
| RuleEvaluations | Jumlah aturan yang diproses oleh penyeimbang beban yang diberikan tingkat permintaan rata-rata lebih dari satu jam.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer |

Namespace AWS/ApplicationELB menyertakan metrik berikut untuk target.

| Metrik | Deskripsi |
| --- | --- |
| HealthyHostCount | Jumlah target yang dianggap sehat.<br><br>Reporting criteria: Dilaporkan jika pemeriksaan kondisi diaktifkan |

| Metrik | Deskripsi |
| --- | --- |
| | Statistics: Statistik yang paling berguna adalah `Average`, `Minimum`, dan `Maximum`.<br><br>Dimensi<br><br>• `LoadBalancer` , `TargetGroup`<br>• `LoadBalancer` , `AvailabilityZone` , `TargetGroup` |
| `HTTPCode_Target_2XX_Count` , `HTTPCode_Target_3XX_Count` , `HTTPCode_Target_4XX_Count` , `HTTPCode_Target_5XX_Count` | Jumlah kode respons HTTP yang dihasilkan oleh target. Jumlah ini tidak termasuk kode respons apa pun yang dihasilkan oleh penyeimbang beban.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah `Sum`. `Minimum`, `Maximum`, dan `Average` semua kembali 1.<br><br>Dimensi<br><br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer`<br>• `TargetGroup` , `LoadBalancer`<br>• `TargetGroup` , `AvailabilityZone` , `LoadBalancer` |

| Metrik | Deskripsi |
| --- | --- |
| RequestCountPerTarget | 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.<br><br>Hitungan ini menggunakan jumlah total permintaan yang diterima oleh kelompok sasaran, dibagi dengan jumlah target sehat dalam kelompok sasaran. Jika tidak ada target yang sehat dalam kelompok sasaran, jumlah total target dilaporkan.<br><br>Reporting criteria: Selalu dilaporkan<br><br>Statistics: Satu-satunya statistik yang valid adalah Sum. Statistik ini mewakili rata-rata bukan jumlah.<br><br>Dimensi<br><br>• TargetGroup<br>• TargetGroup , AvailabilityZone<br>• LoadBalancer , TargetGroup<br>• LoadBalancer , AvailabilityZone , TargetGroup |
| TargetConnectionErrorCount | Jumlah koneksi yang tidak berhasil dibuat antara penyeimbang beban dan target. Metrik ini tidak berlaku jika targetnya adalah fungsi Lambda.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer<br>• TargetGroup , LoadBalancer<br>• TargetGroup , AvailabilityZone , LoadBalancer |

| Metrik | Deskripsi |
| --- | --- |
| TargetResponseTime | Waktu berlalu, dalam hitungan detik, setelah permintaan meninggal kan penyeimbang beban hingga target mulai mengirim header respons. Ini setara dengan bidang target_processing_time di log akses.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah Average dan pNN.NN (persentil).<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer<br>• TargetGroup , LoadBalancer<br>• TargetGroup , AvailabilityZone , LoadBalancer |
| TargetTLSNegotiationErrorCount | 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.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer<br>• TargetGroup , LoadBalancer<br>• TargetGroup , AvailabilityZone , LoadBalancer |

| Metrik | Deskripsi |
| --- | --- |
| UnHealthyHostCount | Jumlah target yang dianggap tidak sehat.<br><br>Reporting criteria: Dilaporkan jika pemeriksaan kondisi diaktifkan<br><br>Statistics: Statistik yang paling berguna adalah Average, Minimum, dan Maximum.<br><br>Dimensi<br><br>• LoadBalancer , TargetGroup<br>• LoadBalancer , AvailabilityZone , TargetGroup |

AWS/ApplicationELBNamespace menyertakan metrik berikut untuk kesehatan grup target. Untuk informasi selengkapnya, lihat the section called “Kesehatan kelompok sasaran”.

| Metrik | Deskripsi |
| --- | --- |
| HealthyStateDNS | Jumlah zona yang memenuhi persyaratan status sehat DNS.<br><br>Statistics: Statistik yang paling berguna adalah Min.<br><br>Dimensi<br><br>• LoadBalancer , TargetGroup<br>• AvailabilityZone , LoadBalancer , TargetGroup |
| HealthyStateRouting | Jumlah zona yang memenuhi persyaratan keadaan sehat perutean.<br><br>Statistics: Statistik yang paling berguna adalah Min.<br><br>Dimensi<br><br>• LoadBalancer , TargetGroup<br>• AvailabilityZone , LoadBalancer , TargetGroup |
| UnhealthyRoutingRequestCount | Jumlah permintaan yang dirutekan menggunakan tindakan failover routing (gagal terbuka). |

| Metrik | Deskripsi |
| --- | --- |
| | Statistics: Statistik yang paling berguna adalah `Sum`.<br><br>Dimensi<br><br>• `LoadBalancer` , `TargetGroup`<br>• `AvailabilityZone` , `LoadBalancer` , `TargetGroup` |
| `UnhealthyStateDNS` | Jumlah zona yang tidak memenuhi persyaratan keadaan sehat DNS dan karenanya ditandai tidak sehat di DNS.<br><br>Statistics: Statistik yang paling berguna adalah `Min`.<br><br>Dimensi<br><br>• `LoadBalancer` , `TargetGroup`<br>• `AvailabilityZone` , `LoadBalancer` , `TargetGroup` |
| `UnhealthyStateRouting` | 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.<br><br>Statistics: Statistik yang paling berguna adalah `Min`.<br><br>Dimensi<br><br>• `LoadBalancer` , `TargetGroup`<br>• `AvailabilityZone` , `LoadBalancer` , `TargetGroup` |

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 |
| --- | --- |
| | mendapatkan kode alasan kesalahan, periksa bidang error_reason dari log akses.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah `Sum`.<br><br>Dimensi<br><br>• `TargetGroup`<br>• `TargetGroup` , `LoadBalancer` |
| `LambdaTargetProcessedBytes` | Jumlah total byte yang diproses oleh penyeimbang beban untuk permintaan ke dan respons dari fungsi Lambda.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah `Sum`.<br><br>Dimensi<br><br>• `LoadBalancer` |
| `LambdaUserError` | 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 error_reason dari log akses.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah `Sum`.<br><br>Dimensi<br><br>• `TargetGroup`<br>• `TargetGroup` , `LoadBalancer` |

Namespace AWS/ApplicationELB menyertakan metrik berikut untuk autentikasi pengguna.

| Metrik | Deskripsi |
| --- | --- |
| ELBAuthError | 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 error_reason dari log akses.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |
| ELBAuthFailure | 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 error_reason dari log akses.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah Sum.<br><br>Dimensi<br><br>• LoadBalancer<br>• AvailabilityZone , LoadBalancer |
| ELBAuthLatency | 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.<br><br>Reporting criteria: Ada nilai bukan nol |

| Metrik | Deskripsi |
| --- | --- |
| | Statistics: Semua statistik bermakna.<br><br>Dimensi<br><br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |
| `ELBAuthRefreshTokenSuccess` | Frekuensi penyeimbang beban berhasil merefresh klaim pengguna menggunakan token refresh yang diberikan oleh IdP.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah `Sum`.<br><br>Dimensi<br><br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |
| `ELBAuthSuccess` | Jumlah tindakan autentikasi yang berhasil. Metrik ini bertambah di akhir alur kerja autentikasi setelah penyeimbang beban mengambil klaim pengguna dari IdP.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Statistik yang paling berguna adalah `Sum`.<br><br>Dimensi<br><br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |

| Metrik | Deskripsi |
| --- | --- |
| `ELBAuthUserClaimsSizeExceeded` | Frekuensi IdP yang dikonfigurasi mengembalikan klaim pengguna yang ukurannya melebihi 11K byte.<br><br>Reporting criteria: Ada nilai bukan nol<br><br>Statistics: Satu-satunya statistik yang bermakna adalah `Sum`.<br><br>Dimensi<br><br>• `LoadBalancer`<br>• `AvailabilityZone` , `LoadBalancer` |

## 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 seperti berikut: app/load-balancer-name/1234567890123456 (bagian akhir dari ARN penyeimbang beban). |
| `TargetGroup` | Memfilter data metrik berdasarkan grup target. Tentukan grup target sebagai berikut: targetgroup/target-group-name/1234567890123456 (bagian akhir dari ARN grup target). |

## 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 load balancer 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

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 perintah get-metric-statistics 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

- 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-akun-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://my-bucket/my-prefix/AWSLogs/123456789012/elasticloadbalancing/us-
east-2/2022/05/01/123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Berikut ini adalah contoh nama file log tanpa awalan:

```
s3://my-bucket/AWSLogs/123456789012/elasticloadbalancing/us-
east-2/2022/05/01/123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

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 Layanan Penyimpanan Sederhana Amazon.

## 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. Elastic Load Balancing tidak mencatat permintaan pemeriksaan kondisi.

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



## Sintaks

Tabel berikut menjelaskan bidang entri log akses, secara berurutan. Semua bidang dibatasi oleh spasi. Saat bidang baru diperkenalkan, mereka ditambahkan ke akhir entri log. Anda harus mengabaikan bidang apa pun pada akhir entri log yang tidak Anda harapkan.

| Bidang | Deskripsi |
| --- | --- |
| jenis | Jenis permintaan atau koneksi. Nilai yang mungkin adalah sebagai berikut (abaikan nilai lainnya):<br>• `http` — HTTP<br>• `https` — HTTP melalui TLS<br>• `h2` — HTTP/2 melalui TLS<br>• `grpcs`— gRPC melalui TLS<br>• `ws` — WebSockets<br>• `wss`— WebSockets lebih dari TLS |
| waktu | Waktu saat penyeimbang beban menghasilkan respons terhadap klien, dalam format ISO 8601. Sebab WebSockets, ini adalah waktu ketika koneksi ditutup. |

| Bidang | Deskripsi |
| --- | --- |
| elb | ID sumber daya penyeimbang beban. Jika Anda mengurai entri log akses, perhatikan bahwa ID sumber daya dapat berisi garis miring (/). |
| client:port | Alamat IP dan port dari klien yang meminta. Jika ada proxy di depan penyeimbang beban, bidang ini berisi alamat IP proxy. |
| target:port | Alamat IP dan port target yang diproses permintaan ini.<br><br>Jika klien tidak mengirimkan permintaan penuh, penyeimbang beban tidak dapat mengirimkan permintaan ke target, dan nilai ini diatur ke -.<br><br>Jika target adalah fungsi Lambda, nilai ini diatur ke -.<br><br>Jika permintaan diblokir oleh AWS WAF, nilai ini diatur ke - dan nilai elb_status_code diatur ke 403. |
| request_processing_time | Total waktu yang berlalu (dalam detik, dengan presisi milidetik) sejak penyeimbang beban menerima permintaan hingga saat mengirimkan permintaan ke target.<br><br>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.<br><br>Nilai ini juga dapat diatur ke -1 jika target terdaftar tidak merespons sebelum batas waktu idle.<br><br>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. `request_processing_time` |

| Bidang | Deskripsi |
| --- | --- |
| target_processing_time | Total waktu yang berlalu (dalam detik, dengan presisi milidetik) sejak penyeimbang beban mengirimkan permintaan ke target hingga target mulai mengirimkan header respons.<br><br>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.<br><br>Nilai ini juga dapat diatur ke -1 jika target terdaftar tidak merespons sebelum batas waktu idle.<br><br>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` |
| response_processing_time | 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.<br><br>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. |
| elb_status_code | Kode status respons dari penyeimbang beban. |
| target_status_code | Kode status respons dari target. Nilai ini dicatat hanya jika koneksi dibuat ke target dan target mengirimkan respon. Jika tidak, nilainya diatur ke -. |
| received_bytes | 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. |

| Bidang | Deskripsi |
| --- | --- |
| sent_bytes | Ukuran respons dalam byte, dikirim ke klien (peminta). Untuk permintaa n HTTP, ini termasuk header. Untuk WebSockets, ini adalah jumlah total byte yang dikirim ke klien pada koneksi. |
| "request" | 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. |
| "user_agent" | 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. |
| ssl_cipher | [Listener HTTPS] Cipher SSL. Nilai ini diatur ke - jika listener bukan listener HTTPS. |
| ssl_protocol | [Listener HTTPS] Protokol SSL. Nilai ini diatur ke - jika listener bukan listener HTTPS. |
| target_group_arn | Amazon Resource Name (ARN) dari grup target. |
| "trace_id" | Isi dari header X-Amzn-Trace-Id, diapit dalam tanda kutip ganda. |
| "domain_name" | [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. |
| "chosen_cert_arn" | [Listener HTTPS] ARN sertifikat yang disajikan kepada klien, diapit dalam tanda kutip ganda. Nilai ini diatur ke `session-reused` jika sesi digunakan kembali. Nilai ini diatur ke - jika listener bukan listener HTTPS. |

| Bidang | Deskripsi |
| --- | --- |
| matched_rule_priority | 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 -. |
| request_creation_time | Waktu saat penyeimbang beban menerima permintaan dari klien, dalam format ISO 8601. |
| "actions_executed" | 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 Tindakan yang diambil. Jika tidak ada tindakan yang diambil, seperti untuk permintaan yang salah format, nilai ini diatur ke -. |
| "redirect_url" | URL target pengalihan untuk header lokasi respons HTTP, diapit dalam tanda kutip ganda. Jika tidak ada tindakan pengalihan yang diambil, nilai ini diatur ke -. |
| "error_reason" | Kode alasan kesalahan, diapit dalam tanda kutip ganda. Jika permintaa n gagal, ini adalah salah satu kode kesalahan yang dijelaskan dalam Kode alasan kesalahan. Jika tindakan yang diambil tidak menyertakan tindakan autentikasi atau target bukan fungsi Lambda, nilai ini diatur ke -. |
| "target:port_list" | 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.<br><br>Jika klien tidak mengirimkan permintaan penuh, penyeimbang beban tidak dapat mengirimkan permintaan ke target, dan nilai ini diatur ke -.<br><br>Jika target adalah fungsi Lambda, nilai ini diatur ke -.<br><br>Jika permintaan diblokir oleh AWS WAF, nilai ini diatur ke - dan nilai elb_status_code diatur ke 403. |

| Bidang | Deskripsi |
| --- | --- |
| "target_status_code_list" | 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.<br><br>Nilai ini dicatat hanya jika koneksi dibuat ke target dan target mengirimkan respon. Jika tidak, nilainya diatur ke -. |
| "classification" | 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.<br><br>Jika permintaan sesuai dengan RFC 7230, nilai ini diatur ke -. |
| "classification_reason" | Kode alasan klasifikasi, diapit dalam tanda kutip ganda. Jika permintaan tidak sesuai dengan RFC 7230, ini adalah salah satu kode klasifikasi yang dijelaskan dalam Alasan klasifikasi. Jika permintaan sesuai dengan RFC 7230, nilai ini diatur ke -. |
| conn_trace_id | 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. |

## 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.
- `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.
- `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 |
|---|---|---|
| `AmbiguousUri` | URI Permintaan berisi karakter kontrol. | Ambigu |
| `BadContentLength` | Header Content-Length berisi nilai yang tidak dapat diuraikan atau bukan angka yang valid. | Parah |
| `BadHeader` | Header berisi karakter null atau carriage return. | Parah |
| `BadTransferEncoding` | Header Transfer-Encoding berisi nilai yang buruk. | Parah |
| `BadUri` | URI permintaan berisi karakter null atau carriage return. | Parah |
| `BadMethod` | Metode permintaannya salah format. | Parah |
| `BadVersion` | Versi permintaannya salah format. | Parah |
| `BothTeClPresent` | Permintaan berisi header Transfer-Encoding dan header Content-Length. | Ambigu |
| `DuplicateContentLength` | Ada beberapa header Content-Length dengan nilai yang sama. | Ambigu |

| Kode | Deskripsi | Klasifikasi |
| --- | --- | --- |
| `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 |
| `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` |
| `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` |

| Kode | Deskripsi | Metrik |
| --- | --- | --- |
| `AuthTokenEpRequestFailed` | Ada respons kesalahan (non-2xx) dari titik akhir token. | `ELBAuthError` |
| `AuthTokenEpRequestTimeout` | Penyeimbang beban tidak dapat berkomunikasi dengan titik akhir token. | `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. | `ELBAuthError` |
| `AuthUserinfoResponseSizeExceeded` | Ukuran klaim yang dikembalikan oleh IdP melebihi 11K byte. | `ELBAuthUserClaimsSizeExceeded` |

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. |

## Contoh Entri log

Berikut ini adalah contoh entri log. Perhatikan bahwa teks muncul pada 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" "-" "-"
```

### 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_123456
```

### 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" "-" "-"
```

### 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" "-" "-"
```

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" "-" "-"
```

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" "-" "-" "-" "-" "-" "-"
```

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" "-" "-" "-" "-"
```

## Memproses berkas log akses

Berkas log akses terkompresi. 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 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: Konfigurasikan 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 Layanan Penyimpanan Sederhana Amazon.
    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 yang tersedia

Kebijakan bucket yang akan Anda gunakan bergantung pada Wilayah AWS dan jenis zona.

Wilayah tersedia per Agustus 2022 atau lebih baru

Kebijakan ini memberikan izin ke layanan pengiriman log yang ditentukan. Gunakan kebijakan ini untuk penyeimbang beban di Availability Zone dan Local Zones di Wilayah berikut:

- Asia Pasifik (Hyderabad)
- Asia Pasifik (Melbourne)
- Kanada Barat (Calgary)

- Eropa (Spanyol)
- Eropa (Zürich)
- Israel (Tel Aviv)
- Timur Tengah (UEA)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*"
    }
  ]
}
```

Wilayah tersedia sebelum Agustus 2022

Kebijakan ini memberikan izin ke ID akun Elastic Load Balancing yang ditentukan. Gunakan kebijakan ini untuk load balancers di Availability Zones atau Local Zones di Daerah pada daftar di bawah ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::elb-account-id:root"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*"
    }
  ]
}
```

Ganti *elb-account-id dengan ID* Elastic Load Akun AWS Balancing untuk Wilayah Anda:

- 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

Ganti *my-s3-arn* dengan ARN lokasi untuk log akses Anda. ARN yang Anda tentukan tergantung pada apakah Anda berencana untuk menentukan awalan saat Anda mengaktifkan log akses di langkah 3.

- Contoh ARN dengan awalan

```
arn:aws:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*
```

- Contoh ARN tanpa awalan

```
arn:aws:s3:::bucket-name/AWSLogs/aws-account-id/*
```

Menggunakan NotPrincipal kapan EffectDeny.

Jika kebijakan bucket Amazon S3 digunakan `Effect` dengan nilai `Deny` dan menyertakan `NotPrincipal` seperti yang ditunjukkan pada contoh di bawah ini, pastikan kebijakan tersebut `logdelivery.elasticloadbalancing.amazonaws.com` disertakan dalam daftar. `Service`

```
{
  "Effect": "Deny",
  "NotPrincipal": {
    "Service": [
       "logdelivery.elasticloadbalancing.amazonaws.com",
       "example.com"
    ]
  }
},
```

AWS GovCloud (US) Regions

Kebijakan ini memberikan izin ke ID akun Elastic Load Balancing yang ditentukan. Gunakan kebijakan ini untuk load balancers di Availability Zones atau Local Zones di AWS GovCloud (US) Daerah pada daftar di bawah ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws-us-gov:iam::elb-account-id:root"
      },
      "Action": "s3:PutObject",
      "Resource": "my-s3-arn"
    }
  ]
}
```

Ganti *elb-account-id dengan ID* Elastic Load Akun AWS Balancing untuk Wilayah Anda: AWS GovCloud (US)

- AWS GovCloud (AS-Barat) — 048591011584
- AWS GovCloud (AS-Timur) — 190560391635

Ganti *my-s3-arn* dengan ARN lokasi untuk log akses Anda. ARN yang Anda tentukan tergantung pada apakah Anda berencana untuk menentukan awalan saat Anda mengaktifkan log akses di langkah 3.

- Contoh ARN dengan awalan

```
arn:aws-us-gov:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*
```

- Contoh ARN tanpa awalan

```
arn:aws-us-gov:s3:::bucket-name/AWSLogs/aws-account-id/*
```

Zona Outposts

Kebijakan berikut memberikan izin ke layanan pengiriman log yang ditentukan. Gunakan kebijakan ini untuk penyeimbang beban di Zona Outposts.

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": "logdelivery.elb.amazonaws.com"
    },
    "Action": "s3:PutObject",
    "Resource": "arn:aws:s3:::bucket-name/prefix/AWSLogs/your-aws-account-id/*",
    "Condition": {
        "StringEquals": {
            "s3:x-amz-acl": "bucket-owner-full-control"
        }
    }
}
```

Untuk melampirkan kebijakan bucket untuk log akses ke bucket Anda menggunakan konsol Amazon 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.

## Langkah 3: Konfigurasikan log akses

Gunakan prosedur berikut untuk mengonfigurasi log akses 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 mengaktifkan log akses untuk penyeimbang beban Anda menggunakan konsol

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://bucket-name/prefix`
   - URI tanpa awalan: `s3://bucket-name`
7. Pilih Simpan perubahan.

Untuk mengaktifkan log akses menggunakan AWS CLI

Gunakan perintah modify-load-balancer-attributes.

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.

### 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 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 akses.
3. Arahkan ke file pengujian,`ELBAccessLogTestFile`. Lokasi tergantung pada apakah Anda menggunakan awalan.
   - Lokasi dengan awalan: *`my-bucket`*`/`*`prefix`*`/AWSLogs/`*`123456789012`*`/ELBAccessLogTestFile`
   - Lokasi tanpa awalan: *`my-bucket`*`/AWSLogs/`*`123456789012`*`/ELBAccessLogTestFile`

### 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 Bekerja dengan bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Untuk menonaktifkan log akses menggunakan konsol

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.

Untuk menonaktifkan log akses menggunakan AWS CLI

Gunakan perintah modify-load-balancer-attributes.

# 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 jabat 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

- 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-akun-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://my-bucket/my-prefix/AWSLogs/123456789012/elasticloadbalancing/us-
east-2/2022/05/01/conn_log.123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

Berikut ini adalah contoh nama file log tanpa awalan:

```
s3://my-bucket/AWSLogs/123456789012/elasticloadbalancing/us-
east-2/2022/05/01/conn_log.123456789012_elasticloadbalancing_us-east-2_app.my-
loadbalancer.1234567890abcdef_20220215T2340Z_172.160.001.192_20sg8hgm.log.gz
```

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 Layanan Penyimpanan Sederhana Amazon.

## 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



## Sintaks

Entri log koneksi menggunakan format berikut:

```
[timestamp] [client_ip] [client_port] [listener_port] [tls_protocol] [tls_cipher]
 [tls_handshake_latency] [leaf_client_cert_subject] [leaf_client_cert_validity]
 [leaf_client_cert_serial_number] [tls_verify_status]
```

Tabel berikut menjelaskan bidang entri log koneksi, secara berurutan. Semua bidang dibatasi oleh spasi. Saat bidang baru diperkenalkan, mereka ditambahkan ke akhir entri log. Anda harus mengabaikan bidang apa pun pada akhir entri log yang tidak Anda harapkan.

| Bidang | Deskripsi |
| --- | --- |
| timestamp | Waktu, dalam format ISO 8601, ketika penyeimbang beban berhasil dibuat atau gagal membuat koneksi. |
| client_ip | Alamat IP dari klien yang meminta. |
| client_port | Port klien yang meminta. |
| listener_port | Port pendengar penyeimbang beban menerima permintaan klien. |
| tls_protocol | [HTTPS listener] Protokol SSL/TLS yang digunakan selama jabat tangan. Bidang ini diatur - untuk permintaan non-SSL/TLS. |
| tls_cipher | [HTTPS listener] Protokol SSL/TLS yang digunakan selama jabat tangan. Bidang ini diatur - untuk permintaan non-SSL/TLS. |
| tls_handshake_latency | [HTTPS listener] Total waktu dalam hitungan detik, dengan presisi milidetik, berlalu saat membuat jabat tangan yang sukses. Bidang ini diatur ke - kapan:<br>• Permintaan yang masuk bukan permintaan SSL/TLS. |

| Bidang | Deskripsi |
| --- | --- |
| | • Jabat tangan tidak berhasil dibuat. |
| leaf_client_cert_subject | [HTTPS listener] Nama subjek dari sertifikat klien daun. Bidang ini diatur ke - kapan:<br>• Permintaan yang masuk bukan permintaan SSL/TLS.<br>• Pendengar penyeimbang beban tidak dikonfigurasi dengan mTL diaktifkan.<br>• Server tidak dapat memuat/mengurai sertifikat klien daun. |
| leaf_client_cert_validitas | [HTTPS listener] Validitas, dengan `not-before` dan `not-after` dalam format ISO 8601, dari sertifikat klien daun. Bidang ini diatur ke - kapan:<br>• Permintaan yang masuk bukan permintaan SSL/TLS.<br>• Pendengar penyeimbang beban tidak dikonfigurasi dengan mTL diaktifkan.<br>• Server tidak dapat memuat/mengurai sertifikat klien daun. |
| leaf_client_cert_serial_number | [HTTPS listener] Nomor seri sertifikat klien daun. Bidang ini diatur ke - kapan:<br>• Permintaan yang masuk bukan permintaan SSL/TLS.<br>• Pendengar penyeimbang beban tidak dikonfigurasi dengan mTL diaktifkan.<br>• Server tidak dapat memuat/mengurai sertifikat klien daun. |
| tls_verify_status | [HTTPS listener] Status permintaan koneksi. Nilai ini adalah `Success` jika koneksi berhasil dibuat. Pada koneksi yang gagal nilainya adalah`Failed:$error_code` . |
| conn_trace_id | 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. |

## Kode alasan kesalahan

Jika penyeimbang beban tidak dapat membuat koneksi, penyeimbang beban menyimpan salah satu kode alasan berikut di log koneksi.

| Kode | Deskripsi | |
|---|---|---|
| `ClientCertMaxChainDepthExceeded` | Kedalaman rantai sertifikat klien maksimum telah terlampaui | |
| `ClientCertMaxSizeExceeded` | Ukuran sertifikat klien maksimum telah terlampaui | |
| `ClientCertCrlHit` | Sertifikat klien telah dicabut oleh CA | |
| `ClientCertCrlProcessingError` | Kesalahan pemrosesan CRL | |
| `ClientCertUntrusted` | Sertifikat klien tidak dipercaya | |
| `ClientCertNotYetValid` | Sertifikat klien belum valid | |
| `ClientCertExpired` | Sertifikat klien kedaluwarsa | |
| `ClientCertTypeUnsupported` | Jenis sertifikat klien tidak didukung | |
| `ClientCertInvalid` | Sertifikat klien tidak valid | |

| Kode | Deskripsi | |
|---|---|---|
| `ClientCertRejected` | Sertifikat klien ditolak oleh validasi server kustom | |
| `UnmappedConnectionError` | Kesalahan koneksi runtime yang tidak dipetakan | |

## Contoh Entri log

Berikut ini adalah contoh entri log koneksi.

Berikut ini adalah contoh entri log untuk koneksi yang berhasil dengan pendengar HTTPS dengan mode 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,O=endEntity,L=Seattle,ST=Washington,C=US"
 NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z FEF257372D5C14D4 Success
```

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,O=endEntity,L=Seattle,ST=Washington,C=US"
 NotBefore=2023-09-21T22:43:21Z;NotAfter=2026-06-17T22:43:21Z FEF257372D5C14D4
 Failed:ClientCertUntrusted
```

## 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: Konfigurasikan 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 Layanan Penyimpanan Sederhana Amazon.
    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 yang tersedia

Kebijakan bucket yang akan Anda gunakan bergantung pada Wilayah AWS dan jenis zona.

Wilayah tersedia per Agustus 2022 atau lebih baru

Kebijakan ini memberikan izin ke layanan pengiriman log yang ditentukan. Gunakan kebijakan ini untuk penyeimbang beban di Availability Zone dan Local Zones di Wilayah berikut:

- Asia Pasifik (Hyderabad)
- Asia Pasifik (Melbourne)
- Eropa (Spanyol)
- Eropa (Zürich)
- Israel (Tel Aviv)
- Timur Tengah (UEA)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "logdelivery.elasticloadbalancing.amazonaws.com"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*"
    }
  ]
}
```

Wilayah tersedia sebelum Agustus 2022

Kebijakan ini memberikan izin ke ID akun Elastic Load Balancing yang ditentukan. Gunakan kebijakan ini untuk load balancers di Availability Zones atau Local Zones di Daerah pada daftar di bawah ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::elb-account-id:root"
      },
      "Action": "s3:PutObject",
       "Resource": "arn:aws:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*"
    }
  ]
}
```

Ganti *elb-account-id dengan ID* Elastic Load Akun AWS Balancing untuk Wilayah Anda:

- 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-Barat) — 048591011584
- AWS GovCloud (AS-Timur) — 190560391635

Ganti *my-s3-arn* dengan ARN lokasi untuk log koneksi Anda. ARN yang Anda tentukan tergantung pada apakah Anda berencana untuk menentukan awalan saat Anda mengaktifkan log koneksi di langkah 3.

- Contoh ARN dengan awalan

```
arn:aws:s3:::bucket-name/prefix/AWSLogs/aws-account-id/*
```

- Contoh ARN tanpa awalan

```
arn:aws:s3:::bucket-name/AWSLogs/aws-account-id/*
```

Menggunakan **`NotPrincipal`** kapan **`EffectDeny`**.

Jika kebijakan bucket Amazon S3 digunakan `Effect` dengan nilai `Deny` dan menyertakan `NotPrincipal` seperti yang ditunjukkan pada contoh di bawah ini, pastikan kebijakan tersebut `logdelivery.elasticloadbalancing.amazonaws.com` disertakan dalam daftar. `Service`

```
{
"Effect": "Deny",
"NotPrincipal": {
    "Service": [
       "logdelivery.elasticloadbalancing.amazonaws.com",
       "example.com"
},
```

Untuk melampirkan kebijakan bucket untuk log koneksi ke bucket Anda menggunakan konsol Amazon 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.

## Langkah 3: Konfigurasikan 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 mengaktifkan log koneksi untuk penyeimbang beban Anda menggunakan konsol

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.

Untuk mengaktifkan log koneksi menggunakan AWS CLI

Gunakan perintah modify-load-balancer-attributes.

Untuk mengelola bucket S3 untuk log koneksi Anda

Pastikan untuk menonaktifkan log koneksi sebelum Anda menghapus bucket yang Anda konfigurasikan 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.

## 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: `my-bucket/prefix/AWSLogs/123456789012/ELBConnectionLogTestFile`
   - Lokasi tanpa awalan: `my-bucket/AWSLogs/123456789012/ELBConnectionLogTestFile`

### 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 Bekerja dengan bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Untuk menonaktifkan log koneksi menggunakan konsol

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.

Untuk menonaktifkan log koneksi menggunakan AWS CLI

Gunakan perintah modify-load-balancer-attributes.

# 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.

## Sintaks

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.

waktu

Jangka waktu dalam detik.

id

Pengidentifikasi jejak.

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.

## Pengelogan panggilan API untuk Application Load Balancer Anda menggunakan AWS CloudTrail

Elastic Load Balancing terintegrasi dengan AWS CloudTrail layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan dalam Elastic Load Balancing. CloudTrail menangkap semua panggilan API untuk Elastic Load Balancing sebagai peristiwa. Panggilan yang diambil mencakup panggilan dari panggilan AWS Management Console dan kode ke operasi Elastic Load Balancing API. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail peristiwa secara terus menerus ke bucket Amazon S3, termasuk peristiwa untuk Elastic Load Balancing. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk Elastic Load Balancing, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Untuk memantau tindakan lain untuk penyeimbang beban Anda, seperti saat klien membuat permintaan ke penyeimbang beban Anda, gunakan log akses. Untuk informasi selengkapnya, lihat Log akses untuk Application Load Balancer Anda.

## Informasi Elastic Load Balancing di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di Elastic Load Balancing, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Peristiwa. Anda dapat melihat, mencari, dan mengunduh acara terbaru di AWS akun Anda. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan riwayat CloudTrail acara.

Untuk catatan peristiwa yang sedang berlangsung di AWS akun Anda, termasuk acara untuk Elastic Load Balancing, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua AWS Wilayah. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua tindakan Elastic Load Balancing untuk Application Load Balancer dicatat oleh CloudTrail dan didokumentasikan dalam Referensi API Elastic Load Balancing versi 2015-12-01. Misalnya, panggilan ke `CreateLoadBalancer` dan `DeleteLoadBalancer` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

- Apakah permintaan dibuat dengan kredensyal root.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna terfederasi.

- Apakah permintaan tersebut dibuat oleh Layanan AWS lain.

Untuk informasi selengkapnya, lihat elemen CloudTrailUserIdentity.

## Memahami entri berkas log Elastic Load Balancing

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa merepresentasikan satu permintaan dari sumber apa pun dan menyertakan informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

File log menyertakan peristiwa untuk semua panggilan AWS API untuk Anda Akun AWS, bukan hanya panggilan Elastic Load Balancing API. Anda dapat menemukan panggilan ke API Elastic Load Balancing dengan memeriksa elemen `eventSource` dengan nilai `elasticloadbalancing.amazonaws.com`. Untuk melihat catatan tindakan tertentu, seperti `CreateLoadBalancer`, periksa elemen `eventName` dengan nama tindakan.

Berikut ini adalah contoh catatan CloudTrail log untuk Elastic Load Balancing untuk pengguna yang membuat Application Load Balancer dan kemudian menghapusnya menggunakan. AWS CLI Anda dapat mengidentifikasi CLI menggunakan elemen `userAgent`. Anda dapat mengidentifikasi panggilan API yang diminta menggunakan elemen `eventName`. Informasi tentang pengguna (`Alice`) dapat ditemukan di elemen `userIdentity`.

Example Contoh: CreateLoadBalancer

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "CreateLoadBalancer",
```

```
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "subnets": ["subnet-8360a9e7","subnet-b7d581c0"],
        "securityGroups": ["sg-5943793c"],
        "name": "my-load-balancer",
        "scheme": "internet-facing"
    },
    "responseElements": {
        "loadBalancers":[{
            "type": "application",
            "loadBalancerName": "my-load-balancer",
            "vpcId": "vpc-3ac0fb5f",
            "securityGroups": ["sg-5943793c"],
            "state": {"code":"provisioning"},
            "availabilityZones": [
               {"subnetId":"subnet-8360a9e7","zoneName":"us-west-2a"},
               {"subnetId":"subnet-b7d581c0","zoneName":"us-west-2b"}
            ],
            "dNSName": "my-load-balancer-1836718677.us-west-2.elb.amazonaws.com",
            "canonicalHostedZoneId": "Z2P70J7HTTTPLU",
            "createdTime": "Apr 11, 2016 5:23:50 PM",
            "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/ffcddace1759e1d0",
            "scheme": "internet-facing"
        }]
    },
    "requestID": "b9960276-b9b2-11e3-8a13-f1ef1EXAMPLE",
    "eventID": "6f4ab5bd-2daa-4d00-be14-d92efEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-12-01",
    "recipientAccountId": "123456789012"
}
```

### Example Contoh: DeleteLoadBalancer

```
{
    "eventVersion": "1.03",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "123456789012",
        "arn": "arn:aws:iam::123456789012:user/Alice",
```

```
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "userName": "Alice"
    },
    "eventTime": "2016-04-01T15:31:48Z",
    "eventSource": "elasticloadbalancing.amazonaws.com",
    "eventName": "DeleteLoadBalancer",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "198.51.100.1",
    "userAgent": "aws-cli/1.10.10 Python/2.7.9 Windows/7 botocore/1.4.1",
    "requestParameters": {
        "loadBalancerArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/ffcddace1759e1d0"
    },
    "responseElements": null,
    "requestID": "349598b3-000e-11e6-a82b-298133eEXAMPLE",
    "eventID": "75e81c95-4012-421f-a0cf-babdaEXAMPLE",
    "eventType": "AwsApiCall",
    "apiVersion": "2015-12-01",
    "recipientAccountId": "123456789012"
}
```

# Memecahkan masalah Application Load Balancer

Informasi berikut dapat membantu Anda memecahkan masalah pada Application Load Balancer.

Masalah



## 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 Anda.

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

Kelompok keamanan untuk Load Balancer dan jaringan ACL apapun untuk subnet Load Balancer harus memungkinkan lalu lintas masuk dari klien dan lalu lintas keluar untuk klien pada 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 panjang terkompresi dari salah satu header melebihi 8 K byte atau jika jumlah permintaan yang disajikan 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 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.

## 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-Untuk 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 Load Balancer dan ACL jaringan untuk VPC Anda memungkinkan 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.

## HTTP 501: Tidak diimplementasikan

Load Balancer menerima header Transfer-Pengodeandengan nilai yang tidak didukung. Nilai-nilai yang didukung untuk Transfer Pengodean adalah `chunked`dan `identity`. Sebagai alternatif, Anda dapat menggunakan Header Pengodean-Konten.

## 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

Kelompok target untuk Load Balancer tidak memiliki target yang terdaftar.

## 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.

# 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 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 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 konfirmasikan 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 pendengar HTTPS Anda menggunakan kebijakan keamanan TLS 1.3 untuk koneksi front-end, kebijakan `ELBSecurityPolicy-TLS13-1-0-2021-06` keamanan akan digunakan untuk koneksi back-end.
    - Jika pendengar HTTPS Anda tidak menggunakan kebijakan keamanan TLS 1.3 untuk koneksi front-end, kebijakan `ELBSecurityPolicy-2016-08` keamanan akan digunakan untuk koneksi back-end.

    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.

# Kuota untuk Application Load Balancer Anda

Akun AWS Anda memiliki kuota default, yang sebelumnya disebut sebagai batasan, untuk setiap AWS layanan. Kecuali dinyatakan sebaliknya, setiap kuota unik untuk suatu 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 peningkatan kuota, lihat Meminta peningkatan kuota di Panduan Pengguna Service Quotas. Jika kuota belum tersedia dalam Service Quotas, gunakan formulir peningkatan batas Elastic Load Balancing.

Penyeimbang beban

Akun AWS Anda memiliki kuota berikut yang terkait dengan Application Load Balancer.

| Nama | Default | Dapat disesuaikan |
|---|---|---|
| Application Load Balancer per Wilayah | 50 | Ya |
| Sertifikat per Application Load Balancer (tidak termasuk sertifikat default) | 25 | Ya |
| Listener per Application Load Balancer | 50 | Ya |
| 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 | Ya |

Kelompok-kelompok target

Kuota berikut adalah untuk kelompok sasaran.

| Nama | Default | Dapat disesuaikan |
| --- | --- | --- |
| Grup Target per Wilayah | 3.000 * | Ya |
| Target per Grup Target per Wilayah (contoh atau alamat IP) | 1.000 | Ya |
| 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 | Ya |
| Nilai Syarat per Aturan | 5 | Tidak |
| Wildcard Syarat per Aturan | 5 | Tidak |
| Evaluasi kecocokan per aturan | 5 | Tidak |

Toko kepercayaan

Kuota berikut adalah untuk toko kepercayaan.

| Nama | Default | Dapat disesuaikan |
| --- | --- | --- |
| Toko kepercayaan per akun | 20 | Ya |
| Jumlah pendengar yang menggunakan mTL dalam mode verifikasi, per penyeimbang beban. | 2 | Tidak |

### Sertifikat otoritas sertifikat

Kuota berikut adalah untuk sertifikat CA.

| Nama | Default | Dapat disesuaikan |
| --- | --- | --- |
| Sertifikat CA per toko kepercayaan | 25 | Ya |
| Ukuran sertifikat CA | 16KB | Tidak |
| Kedalaman rantai sertifikat maksimum | 4 | Tidak |

### Daftar pencabutan sertifikat

Kuota berikut adalah untuk daftar pencabutan sertifikat.

| Nama | Default | Dapat disesuaikan |
| --- | --- | --- |
| Daftar pencabutan per toko kepercayaan | 30 | Ya |
| Entri pencabutan per toko kepercayaan | 500.000 | Ya |
| Ukuran file daftar pencabutan | 50MB | 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 respon | 32 K | Tidak |
| Seluruh header permintaan | 64 K | Tidak |

# Riwayat dokumen untuk Application Load Balancer

Tabel berikut menjelaskan rilis untuk Application Load Balancer.

| Perubahan | Deskripsi | Tanggal |
| --- | --- | --- |
| 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 mengonfig urasi 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 kripotogr afi FIPS 140-3 saat mengakhir i koneksi TLS. | 20 November 2023 |
| Daftarkan target menggunak an IPv6 | Rilis ini menambahkan dukungan untuk mendaftarkan instance sebagai target saat ditangani oleh IPv6. | 2 Oktober 2023 |

| | | |
|---|---|---|
| Kebijakan keamanan yang mendukung TLS 1.3 | Rilis ini menambahkan dukungan untuk kebijakan keamanan standar TLS 1.3. | 22 Maret 2023 |
| Pergeseran zona | Rilis ini menambahkan dukungan untuk merutekan lalu lintas dari satu Zona Ketersediaan yang terganggu melalui integrasi dengan Amazon Route 53 Application Recovery Controller. | 28 November 2022 |
| Matikan penyeimbangan beban lintas zona | Rilis ini menambahkan dukungan untuk mematikan penyeimbangan beban lintas zona. | 28 November 2022 |
| Kesehatan kelompok sasaran | Rilis ini menambahkan dukungan untuk mengonfig urasi jumlah minimum atau persentase target yang harus sehat, dan tindakan apa yang dilakukan penyeimbang beban ketika ambang batas tidak terpenuhi. | 28 November 2022 |
| Penyeimbangan beban lintas zona | Rilis ini menambahkan dukungan untuk mengonfig urasi penyeimbangan beban lintas zona di tingkat grup target. | 17 November 2022 |
| Kelompok sasaran IPv6 | Rilis ini menambahkan dukungan untuk mengkonfi gurasi grup target IPv6 untuk Application Load Balancers. | 23 November 2021 |

| | | |
|---|---|---|
| Penyeimbang beban internal IPv6 | Rilis ini menambahkan dukungan untuk mengkonfi gurasi grup target IPv6 untuk Application Load Balancers. | 23 November 2021 |
| AWS PrivateLink dan alamat IP statis | Rilis ini menambahkan dukungan untuk menggunak an AWS PrivateLink dan mengekspos alamat IP statis dengan meneruskan lalu lintas langsung dari Network Load Balancers ke Application Load Balancers. | 27 September 2021 |
| Pelestarian port klien | Rilis ini menambahkan atribut untuk mempertahankan port sumber yang digunakan klien untuk terhubung ke penyeimba ng beban. | 29 Juli 2021 |
| Header TLS | 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 |
| Sertifikat ACM tambahan | Rilis ini mendukung sertifika t RSA dengan panjang kunci 2048, 3072, dan 4096-bit, dan semua sertifikat ECDSA. | 14 Juli 2021 |

| | | |
|---|---|---|
| Kelengketan berbasis aplikasi | Rilis ini menambahkan cookie berbasis aplikasi untuk mendukung sesi lekat untuk penyeimbang beban Anda. | 8 Februari 2021 |
| Kebijakan keamanan untuk FS yang mendukung TLS versi 1.2 | Rilis ini menambahkan kebijakan keamanan untuk Forward Secrecy (FS) yang mendukung TLS versi 1.2. | 24 November 2020 |
| WAF gagal membuka dukungan | Rilis ini menambahkan dukungan untuk mengonfig urasi perilaku penyeimbang beban Anda jika terintegrasi dengan. AWS WAF | 13 November 2020 |
| dukungan gRPC dan HTTP/2 | Rilis ini menambahkan dukungan untuk beban kerja gRPC dan HTTP/2. end-to-end | 29 Oktober 2020 |
| Dukungan pos terdepan | Anda dapat menyediakan Application Load Balancer pada Anda. AWS Outposts | 8 September 2020 |
| Mode mitigasi desync | Rilis ini menambahkan dukungan untuk mode mitigasi desinkronisasi. | 17 Agustus 2020 |
| Permintaan yang paling tidak beredar | Rilis ini menambahkan dukungan untuk algoritme permintaan paling tidak tertunda. | 25 November 2019 |

| | | |
|---|---|---|
| Kelompok sasaran tertimbang | Rilis ini menambahkan dukungan untuk tindakan maju dengan beberapa kelompok target. Permintaan didistrib usikan ke grup target ini berdasarkan berat yang Anda tentukan untuk setiap kelompok target. | 19 November 2019 |
| Atribut baru | Rilis ini menambahkan dukungan untuk atribut routing.http.drop_invalid_h eader_fields.enabled. | 15 November 2019 |
| Kebijakan keamanan untuk FS | Rilis ini menambahkan dukungan untuk tiga kebijakan keamanan kerahasiaan lanjutan yang telah ditentukan sebelumnya. | 8 Oktober 2019 |
| Perutean permintaan lanjutan | Rilis ini menambahkan dukungan untuk jenis syarat tambahan untuk aturan listener Anda. | 27 Maret 2019 |
| Lambda berfungsi sebagai target | Rilis ini menambahkan dukungan untuk mendaftarkan fungsi Lambda sebagai target. | 29 November 2018 |
| Tindakan pengalihan | Rilis ini menambahkan dukungan untuk penyeimba ng beban untuk mengalihk an permintaan ke URL yang berbeda. | 25 Juli 2018 |

| | | |
|---|---|---|
| Tindakan respons tetap | Rilis ini menambahkan dukungan untuk penyeimbang beban untuk mengembalikan respons HTTP kustom. | 25 Juli 2018 |
| Kebijakan keamanan untuk FS dan TLS 1.2 | Rilis ini menambahkan dukungan untuk dua kebijakan keamanan standar tambahan. | Selasa, 06 Juni 2018 |
| Otentikasi pengguna | Rilis ini menambahkan dukungan bagi penyeimba ng beban untuk mengotent ikasi pengguna aplikasi Anda menggunakan identitas perusahaan atau sosial mereka sebelum merutekan permintaan. | 30 Mei 2018 |
| Izin tingkat sumber daya | Rilis ini menambahkan dukungan untuk izin tingkat sumber daya dan penandaan syarat kunci. | 10 Mei 2018 |
| Mode mulai lambat | Rilis ini menambahkan dukungan untuk mode mulai lambat, yang secara bertahap meningkatkan pangsa permintaan penyeimba ng beban mengirimkan ke target yang baru terdaftar saat pemanasan. | 24 Maret 2018 |
| Dukungan SNI | Rilis ini menambahkan dukungan untuk Indikasi Nama Server (SNI). | 10 Oktober 2017 |

| | | |
|---|---|---|
| Alamat IP sebagai target | Rilis ini menambahkan dukungan untuk mendaftarkan alamat IP sebagai target. | 31 Agustus 2017 |
| Perutean berbasis host | Rilis ini menambahkan dukungan untuk permintaan perutean berdasarkan nama host di header host. | 5 April 2017 |
| Kebijakan keamanan untuk TLS 1.1 dan TLS 1.2 | Rilis ini menambahkan kebijakan keamanan untuk TLS 1.1 dan TLS 1.2. | 6 Februari 2017 |
| Dukungan IPv6 | Rilis ini menambahkan dukungan untuk alamat IPv6. | 25 Januari 2017 |
| Minta penelusuran | Rilis ini menambahkan dukungan untuk permintaan pelacakan. | 22 November 2016 |
| Dukungan persentil untuk metrik TargetResponseTime | Rilis ini menambahkan dukungan untuk statistik persentil baru yang didukung oleh Amazon. CloudWatch | 17 November 2016 |
| Jenis penyeimbang beban baru | 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.