



Panduan Pengguna

Elastic Load Balancing



Elastic Load Balancing: Panduan Pengguna

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

Table of Contents

Apa itu Elastic Load Balancing?	1
Manfaat load balancer	1
Fitur Elastic Load Balancing	1
Mengakses Elastic Load Balancing	2
Layanan terkait	2
Harga	3
Cara kerja Elastic Load Balancing	4
Availability Zone dan node Load Balancer	4
Penyeimbangan beban lintas-zona	5
Pergeseran zona	7
Meminta perutean	9
Algoritma perutean	9
Koneksi HTTP	10
Header HTTP	11
Batas header HTTP Header	12
Skema Load Balancer	12
MTU Jaringan	13
Memulai	14
Membuat Application Load Balancer	14
Membuat Network Load Balancer	14
Membuat Gateway Load Balancer	15
Buat Classic Load Balancer	15
Keamanan	16
Perlindungan data	17
Enkripsi diam	18
Enkripsi dalam bergerak	18
Manajemen Identitas dan akses	18
Audiens	19
Mengautentikasi dengan identitas	19
Mengelola akses menggunakan kebijakan	23
Bagaimana Elastic Load Balancing bekerja dengan IAM	26
Izin API:	40
Izin API penandaan sumber daya	42
Peran yang terhubung dengan layanan	45

AWS kebijakan terkelola	47
Validasi kepatuhan	50
Ketahanan	51
Keamanan infrastruktur	52
Isolasi jaringan	53
Pengontrolan lalu lintas jaringan	53
AWS PrivateLink	54
Buat titik akhir antarmuka untuk Keseimbangan Beban Elastis	54
Membuat kebijakan titik akhir VPC untuk Elastic Load Balancing	55
Migrasi Classic Load Balancer Anda	56
Manfaat Migrasi	56
Wisaya migrasi	57
Salin migrasi utilitas	59
Migrasi manual	59
.....	lxiii

Apa itu Elastic Load Balancing?

Elastic Load Balancing secara otomatis mendistribusikan lalu lintas aplikasi masuk di beberapa target, seperti instans EC2, wadah, dan IP addresses, dalam satu atau lebih Zona Ketersediaan. Ini memantau kesehatan target terdaftar, dan mengarahkan lalu lintas hanya ke target yang sehat. Elastic Load Balancing menskalakan kapasitas load balancer Anda secara otomatis sebagai respons terhadap perubahan lalu lintas yang masuk.

Manfaat load balancer

Load balancer mendistribusikan beban kerja ke beberapa sumber daya komputasi, seperti server virtual. Menggunakan load balancer meningkatkan ketersediaan dan toleransi kesalahan aplikasi Anda.

Anda dapat menambahkan dan menghapus sumber daya komputasi dari load balancer saat kebutuhan Anda berubah, tanpa mengganggu seluruh aliran permintaan ke aplikasi Anda.

Anda dapat mengkonfigurasi pemeriksaan kesehatan, yang memantau kesehatan sumber daya komputasi, sehingga load balancer mengirimkan permintaan hanya untuk yang sehat. Anda juga dapat offload pekerjaan enkripsi dan dekripsi ke load balancer Anda sehingga sumber daya komputasi Anda dapat fokus pada pekerjaan utamanya.

Fitur Elastic Load Balancing

Elastic Load Balancing mendukung load balancer berikut: Application Load Balancer, Network Load Balancer, Gateway Load Balancers, dan Classic Load Balancer. Anda dapat memilih jenis load balancer yang paling sesuai dengan kebutuhan Anda. Untuk informasi lebih lanjut, lihat [Perbandingan Produk](#).

Untuk informasi selengkapnya tentang penggunaan setiap penyeimbang beban, lihat dokumentasi berikut:

- [Panduan Pengguna untuk Penyeimbang Beban Aplikasi](#)
- [Panduan Pengguna untuk Network Load Balancers](#)
- [Panduan Pengguna untuk Gateway Load Balancers](#)
- [Panduan Pengguna untuk Penyeimbang Beban Klasik](#)

Mengakses Elastic Load Balancing

Anda dapat membuat, mengakses, dan mengelola load balancers Anda menggunakan salah satu antarmuka berikut:

- **AWS Management Console**— Menyediakan antarmuka web yang dapat Anda gunakan untuk mengakses Elastic Load Balancing.
- **AWS Command Line Interface (AWS CLI)** — Menyediakan perintah untuk serangkaian AWS layanan yang luas, termasuk Elastic Load Balancing. AWS CLI Ini didukung di Windows, macOS, dan Linux. Untuk informasi selengkapnya, lihat [AWS Command Line Interface](#).
- **AWS SDK** — Menyediakan API khusus bahasa dan mengurus banyak detail koneksi, seperti menghitung tanda tangan, menangani percobaan ulang permintaan, dan penanganan kesalahan. Untuk informasi selengkapnya, lihat [SDK AWS](#).
- **Query API**— Menyediakan tindakan API tingkat rendah yang Anda hubungi menggunakan permintaan HTTPS. Menggunakan Query API adalah cara paling langsung untuk mengakses Elastic Load Balancing. Namun, Query API mengharuskan aplikasi Anda menangani detail tingkat rendah seperti membuat hash untuk menandatangani permintaan, dan menangani kesalahan. Untuk informasi lebih lanjut, lihat berikut ini:
 - Aplikasi Load Balancers dan Load Balancers Jaringan —[Versi API 2015-12-01](#)
 - Classic Load Balancers[Versi API 2012-06-01](#)

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 mengkonfigurasi load balancer Anda untuk mengarahkan lalu lintas ke instans EC2 Anda. Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon EC2](#).
- **Amazon EC2 Auto Scaling**— Memastikan bahwa Anda menjalankan jumlah instans yang Anda inginkan, bahkan jika sebuah instans gagal. Amazon EC2 Auto Scaling juga memungkinkan Anda untuk secara otomatis meningkatkan 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 load balancer. Demikian pula, pendaftaran instans yang dihentikan oleh grup Auto Scaling Anda secara otomatis

dibatalkan pendaftarannya dari load balancer. Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon EC2 Auto Scaling](#).

- AWS Certificate Manager— Ketika Anda membuat pendengar HTTPS, Anda dapat menentukan sertifikat yang disediakan oleh ACM. Load balancer menggunakan sertifikat untuk mengakhiri koneksi dan mendekripsi permintaan dari klien.
- Amazon CloudWatch - Memungkinkan Anda memantau penyeimbang beban dan mengambil tindakan sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).
- Amazon ECS-Memungkinkan Anda untuk menjalankan, menghentikan, dan mengelola kontainer Docker pada kluster instans EC2. Anda dapat mengkonfigurasi load balancer Anda untuk mengarahkan lalu lintas ke wadah Anda. Untuk informasi lebih lanjut, lihat Panduan Pengembang layanan Amazon Elastic Container.
- AWS Global Accelerator— Meningkatkan ketersediaan dan kinerja aplikasi Anda. Gunakan akselerator untuk mendistribusikan lalu lintas di beberapa penyeimbang beban di satu atau beberapa Wilayah. AWS Untuk informasi selengkapnya, lihat [AWS Global Accelerator Panduan Pengembang](#).
- Route 53 — Menyediakan cara yang andal dan hemat biaya untuk mengarahkan pengunjung ke situs web dengan menerjemahkan nama domain ke alamat IP numerik yang digunakan komputer untuk terhubung satu sama lain. Misalnya, itu akan diterjemahkan `www.example.com` ke alamat `192.0.2.1` IP numerik. AWS menetapkan URL ke sumber daya Anda, seperti penyeimbang beban. Namun, Anda mungkin ingin sebuah URL yang mudah diingat pengguna. Misalnya, Anda dapat memetakan nama domain Anda ke sebuah load balancer. Untuk informasi lebih lanjut, lihat Panduan Pengembang Amazon Route 53.
- AWS WAF— Anda dapat menggunakan AWS WAF Application Load Balancer Anda untuk mengizinkan atau memblokir permintaan berdasarkan aturan dalam daftar kontrol akses web (web ACL). Lihat informasi selengkapnya di [Panduan Developer AWS WAF](#).

Harga

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

Cara kerja Elastic Load Balancing

Load Balancer menerima lalu lintas masuk dari klien dan merutekan permintaan untuk target yang telah terdaftar (seperti contoh EC2) dalam satu atau lebih Availability Zone. Load Balancer juga memantau kesehatan target yang terdaftar dan memastikan untuk mengarahkan lalu lintas hanya ke target yang sehat. Ketika Load Balancer mendeteksi target yang tidak sehat, maka routing lalu lintas ke target tersebut akan berhenti. Kemudian akan melanjutkan routing lalu lintas ke target ketika mendeteksi bahwa target sehat kembali.

Anda mengkonfigurasi load balancer Anda untuk menerima lalu lintas masuk dengan menentukan satu atau lebih pendengar. Seorang pendengar adalah proses yang memeriksa permintaan koneksi. Permintaan koneksi dikonfigurasi dengan protokol dan nomor port untuk koneksi dari klien ke Load Balancer. Demikian juga, dikonfigurasi dengan protokol dan nomor port untuk koneksi dari Load Balancer ke target.

Elastic Load Balancing mendukung jenis Load Balancer berikut:

- Application Load Balancer
- Network Load Balancers
- Gateway Load Balancer
- Classic Load Balancer

Ada perbedaan utama dalam bagaimana jenis Load Balancer dikonfigurasi. Dengan Application Load Balancer, Load Balancer Jaringan, dan Load Balancer Gateway, Anda daftarkan target-target dalam grup target, dan merutekan lalu lintas ke grup target. Dengan Classic Load Balancer, instans didaftarkan dengan Load Balancer.

Availability Zone dan node Load Balancer

Saat Anda mengaktifkan Availability Zone untuk Load Balancer, Elastic Load Balancing akan menciptakan node Load Balancer di Availability Zone. Jika Anda mendaftarkan target di Availability Zone tetapi tidak mengaktifkannya, target yang telah terdaftar ini tidak menerima lalu lintas. Load Balancer Anda paling efektif bila Anda memastikan bahwa setiap Availability Zone yang diaktifkan memiliki setidaknya satu target yang sudah terdaftar.

Disarankan untuk mengaktifkan beberapa Availability Zone untuk semua load balancer. Tapi, dengan Application Load Balancer, Anda harus mengaktifkan setidaknya dua atau lebih Availability Zone.

Konfigurasi ini membantu memastikan bahwa Load Balancer dapat terus merutekan lalu lintas. Jika satu Availability Zone menjadi tidak tersedia atau tidak memiliki target yang sehat, Load Balancer dapat mengarahkan lalu lintas ke target sehat di Availability Zone lain.

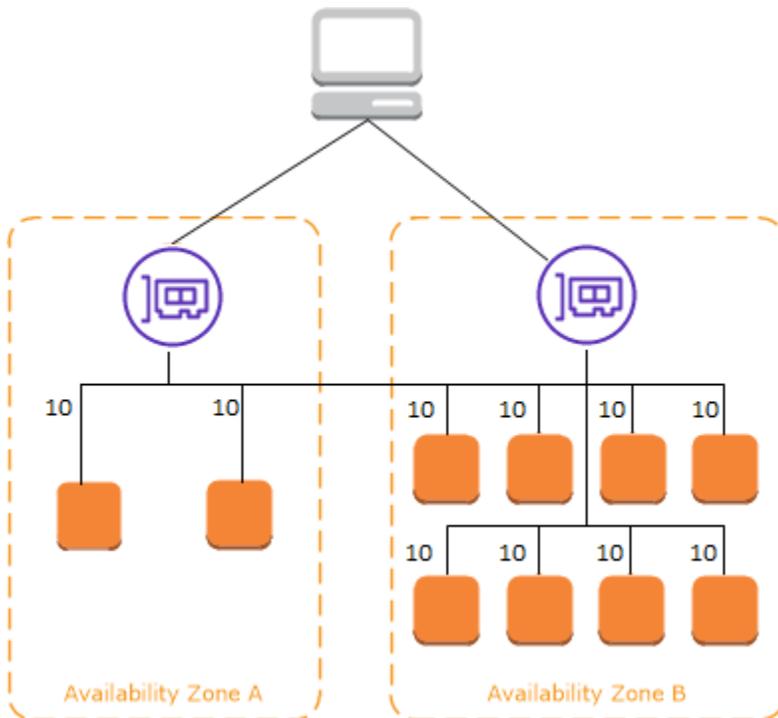
Setelah Anda menonaktifkan Availability Zone, target di Availability Zone tetap terdaftar dengan Load Balancer. Namun, meskipun target-target tersebut tetap terdaftar, Load Balancer tidak mengarahkan lalu lintas ke target-target itu.

Penyeimbangan beban lintas-zona

Node untuk Load Balancer Anda mendistribusikan permintaan dari klien ke target yang telah terdaftar. Ketika load balancing lintas zona diaktifkan, setiap node Load Balancer mendistribusikan lalu lintas di seluruh target yang terdaftar di semua Availability Zone yang telah diaktifkan. Ketika load balancing lintas zona dinonaktifkan, setiap node Load Balancer mendistribusikan lalu lintas hanya di target yang telah terdaftar di Availability Zonanya.

Diagram berikut menunjukkan efek penyeimbangan beban lintas zona dengan round robin sebagai algoritma routing default. Ada dua Availability Zone yang diaktifkan, dengan dua target di Availability Zone A dan delapan target di Availability Zone B. Klien mengirim permintaan, dan Amazon Route 53 menanggapi setiap permintaan dengan alamat IP dari salah satu node Load Balancer. Berdasarkan algoritma routing round robin, lalu lintas didistribusikan sedemikian rupa sehingga setiap node penyeimbang beban menerima 50% lalu lintas dari klien. Setiap node Load Balancer mendistribusikan pangsa lalu lintas di target yang sudah terdaftar dalam ruang lingkungannya.

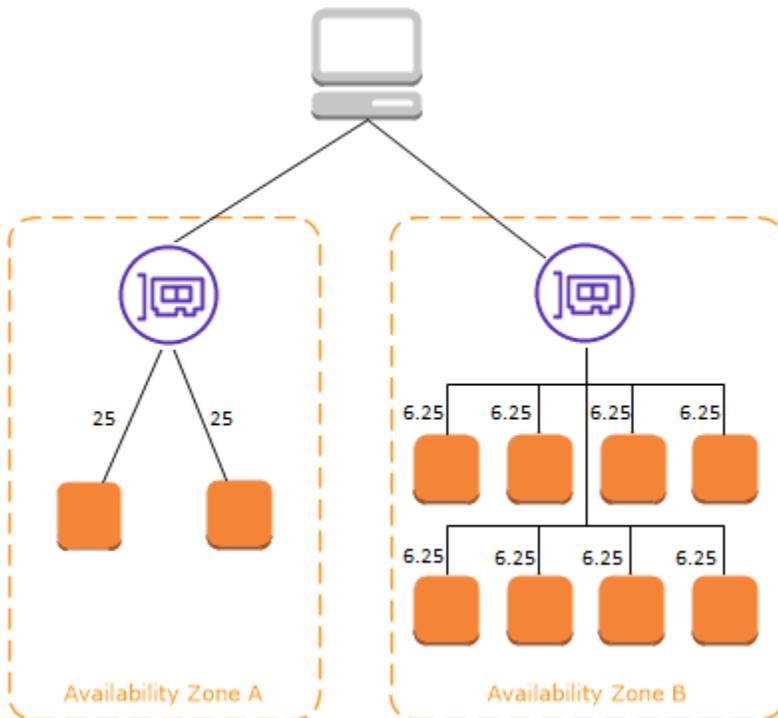
Jika load balancing lintas zona diaktifkan, masing-masing dari 10 target menerima 10% dari lalu lintas. Hal ini karena setiap node Load Balancer dapat merutekan 50% dari lalu lintas klien ke semua 10 target.



Jika load balancing lintas zona dinonaktifkan:

- Masing-masing dari dua target di Availability Zone A menerima 25% dari lalu lintas.
- Masing-masing dari delapan target di Availability Zone B menerima 6,25% dari lalu lintas.

Hal ini karena setiap node Load Balancer dapat merutekan 50% dari lalu lintas klien hanya untuk target di Availability Zonanya.



Dengan Application Load Balancers, penyeimbangan beban lintas zona selalu diaktifkan pada tingkat penyeimbang beban. Pada tingkat kelompok sasaran, penyeimbangan beban lintas zona dapat dinonaktifkan. Untuk informasi selengkapnya, lihat [Menonaktifkan penyeimbangan beban lintas zona](#) di Panduan Pengguna untuk Penyeimbang Beban Aplikasi.

Dengan Load Balancers Jaringan dan Load Balancers Gateway, load balancing lintas zona dinonaktifkan secara default. Setelah Anda membuat Load Balancer, Anda dapat mengaktifkan atau menonaktifkan load balancing lintas zona setiap saat.

Saat Anda membuat Classic Load Balancer, default untuk load balancing lintas zona tergantung pada cara Anda membuat Load Balancer. Dengan API atau CLI, load balancing lintas zona dinonaktifkan secara default. Dengan AWS Management Console, opsi untuk mengaktifkan penyeimbangan beban lintas zona dipilih secara default. Setelah membuat Classic Load Balancer, Anda dapat mengaktifkan atau menonaktifkan penyeimbangan beban lintas zona kapan saja. Untuk informasi selengkapnya, lihat [Aktifkan penyeimbangan beban lintas zona](#) di Panduan Pengguna untuk Classic Load Balancer.

Pergeseran zona

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

Saat Anda memulai pergeseran zona, penyeimbang beban Anda berhenti mengirimkan lalu lintas untuk sumber daya ke Availability Zone yang terpengaruh. Route 53 ARC segera menciptakan pergeseran zona. Namun, dibutuhkan 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 pergeseran zona: pemeriksaan kesehatan dan alamat IP zonal](#) di Panduan Pengembang Pengontrol Pemulihan Aplikasi Amazon Route 53.

Pergeseran zona hanya didukung pada Application Load Balancers dan Network Load Balancer dengan penyeimbangan beban lintas zona dimatikan. Jika Anda mengaktifkan penyeimbangan beban lintas zona, Anda tidak dapat memulai pergeseran zona. Untuk informasi selengkapnya, lihat [Sumber daya yang didukung untuk pergeseran zona di Panduan](#) Pengembang Pengontrol Pemulihan Aplikasi Amazon Route 53.

Sebelum Anda menggunakan pergeseran zona, tinjau hal-hal berikut:

- Penyeimbangan beban lintas zona tidak didukung dengan pergeseran zona. Anda harus mematikan penyeimbangan beban lintas zona untuk menggunakan kemampuan ini.
- Pergeseran zona tidak didukung saat Anda menggunakan Application Load Balancer sebagai titik akhir akselerator di AWS Global Accelerator
- Anda dapat memulai pergeseran zona untuk penyeimbang beban tertentu hanya untuk satu Availability Zone. Anda tidak dapat memulai pergeseran zona untuk beberapa Availability Zone.
- AWS secara proaktif menghapus alamat IP penyeimbang beban zonal dari DNS ketika beberapa masalah infrastruktur berdampak pada layanan. Selalu periksa kapasitas Availability Zone saat ini sebelum Anda memulai pergeseran zona. Jika penyeimbang beban Anda mematikan penyeimbang beban lintas zona dan Anda menggunakan pergeseran zona untuk menghapus alamat IP penyeimbang beban zonal, Availability Zone yang terpengaruh oleh pergeseran zona juga kehilangan kapasitas target.
- Ketika Application Load Balancer adalah target Network Load Balancer, selalu mulai pergeseran zona dari Network Load Balancer. Jika Anda memulai pergeseran zona 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 zona Route 53 ARC](#) di Panduan Pengembang Pengontrol Pemulihan Aplikasi Amazon Route 53.

Meminta perutean

Sebelum klien mengirim permintaan ke Load Balancer Anda, nama domain Load Balancer diselesaikan menggunakan server Domain Name System (DNS). Entri DNS dikendalikan oleh Amazon, karena Load Balancer Anda berada di domain `amazonaws.com`. Amazon DNS server mengembalikan satu atau lebih alamat IP untuk klien. Ini adalah alamat IP dari node Load Balancer untuk Load Balancer Anda. Dengan Network Load Balancers, Elastic Load Balancing menciptakan antarmuka jaringan untuk setiap Availability Zone yang Anda aktifkan, dan menggunakannya untuk mendapatkan alamat IP statis. Anda dapat secara opsional mengaitkan satu alamat IP Elastis dengan setiap antarmuka jaringan saat Anda membuat Network Load Balancer.

Ketika lalu lintas ke aplikasi Anda berubah dari waktu ke waktu, Elastis Load Balancing mengukur beban Anda dan memperbarui entri DNS. Entri DNS juga menentukan time-to-live (TTL) 60 detik. Hal ini membantu memastikan bahwa alamat IP dapat dipetakan kembali dengan cepat dalam menanggapi perubahan lalu lintas.

Klien menentukan alamat IP mana yang akan digunakan untuk mengirim permintaan ke Load Balancer. Simpul Load Balancer yang menerima permintaan memilih target sehat yang telah terdaftar dan mengirimkan permintaan ke target menggunakan alamat IP pribadi.

Untuk informasi selengkapnya, lihat [Merutekan lalu lintas ke penyeimbang beban ELB](#) di Panduan Pengembang Amazon Route 53.

Algoritma perutean

Dengan Application Load Balancer, simpul Load Balancer yang menerima permintaan menggunakan proses berikut:

1. Mengevaluasi aturan pendengar dalam rangka prioritas untuk menentukan aturan yang diterapkan.
2. Memilih target dari grup target untuk tindakan aturan, menggunakan algoritma routing yang dikonfigurasi untuk grup target. Algoritma routing default adalah round robin. Routing dilakukan secara mandiri untuk setiap grup target, bahkan ketika target telah terdaftar dengan beberapa kelompok target.

Dengan Load Balancer Jaringan, simpul Load Balancer yang menerima sambungan menggunakan proses berikut:

1. Memilih target dari kelompok target untuk aturan default menggunakan algoritma aliran hash. Ini mendasarkan algoritma pada:

- Protokol
 - Alamat IP sumber dan port sumber
 - Alamat IP tujuan dan port tujuan
 - Nomor urut TCP
2. Rute=rute setiap koneksi TCP individu untuk satu target untuk kehidupan koneksi. Koneksi-koneksi TCP dari klien memiliki port sumber yang berbeda dan nomor urut, dan dapat diarahkan ke target yang berbeda.

Dengan Classic Load Balancer, node Load Balancer yang menerima permintaan memilih instans yang telah terdaftar sebagai berikut:

- Menggunakan algoritma round robin routing untuk pendengar TCP
- Menggunakan algoritma routing permintaan yang paling tidak beredar untuk HTTP dan HTTPS pendengar

Koneksi HTTP

Classic Load Balancer menggunakan koneksi-koneksi pra-terbuka, tetapi Balancers Load Aplikasi tidak menggunakan koneksi-koneksi tersebut. Baik Classic Load Balancers dan Application Load Balancer menggunakan koneksi multiplexing. Ini berarti bahwa permintaan dari beberapa klien pada beberapa koneksi-koneksi front-end dapat dirutekan ke target yang diberikan melalui koneksi backend tunggal. Koneksi multiplexing meningkatkan latensi dan mengurangi beban pada aplikasi Anda. Untuk mencegah multiplexing koneksi, nonaktifkan keep-alive header HTTP dengan mengatur `Connection: close` header dalam respons HTTP Anda.

Application Load Balancer dan Balancers Beban Klasik mendukung HTTP pipelined pada koneksi front-end. Pipelined HTTP pada koneksi backend tidak didukung.

Application Load Balancers mendukung metode permintaan HTTP berikut: GET, HEAD, POST, PUT, DELETE, OPTIONS, dan PATCH.

Application Load Balancer mendukung protokol berikut pada koneksi front-end: HTTP/0.9, HTTP/1.0, HTTP/1.1, dan HTTP/2. Anda dapat menggunakan HTTP/2 hanya dengan pendengar HTTPS, dan dapat mengirim hingga 128 permintaan secara paralel menggunakan satu koneksi HTTP/2. Application Load Balancers juga mendukung peningkatan koneksi dari HTTP ke. WebSockets

Namun, jika ada peningkatan koneksi, aturan dan AWS WAF integrasi perutean pendengar Application Load Balancer tidak lagi berlaku.

Application Load Balancer menggunakan HTTP/1.1 pada koneksi backend (Load Balancer ke target terdaftar) secara default. Namun, Anda dapat menggunakan versi protokol untuk mengirim permintaan ke target menggunakan HTTP/2 atau gRPC. Untuk informasi selengkapnya, lihat [Versi protokol](#). `keep-alive` header didukung pada koneksi backend secara default. Untuk permintaan HTTP/1.0 dari klien yang tidak memiliki header host, Load Balancer menghasilkan header host untuk permintaan HTTP/1.1 yang dikirim pada koneksi backend. Header host berisi nama DNS dari Load Balancer.

Classic Load Balancer mendukung protokol berikut pada koneksi-koneksi front-end (klien ke Load Balancer): HTTP/0.9, HTTP/1.0, dan HTTP/1.1. Mereka menggunakan HTTP/1.1 pada koneksi backend (penyeimbang beban ke target terdaftar). `keep-alive` header didukung pada koneksi backend secara default. Untuk permintaan HTTP/1.0 dari klien yang tidak memiliki header host, Load Balancer menghasilkan header host untuk permintaan HTTP/1.1 yang dikirim pada koneksi backend. Header host berisi alamat IP dari node Load Balancer.

Header HTTP

Application Load Balancer dan Load Balancers Klasik secara otomatis ditambahkan `X-Forwarded-For`, `X-Forwarded-Proto`, dan `X-Forwarded-Port` header untuk permintaan.

Application Load Balancer mengkonversi nama host di header host HTTP menjadi huruf kecil sebelum mengirim mereka ke target.

Untuk koneksi front-end yang menggunakan HTTP/2, nama header menggunakan huruf kecil. Sebelum permintaan dikirim ke target menggunakan HTTP/1.1, nama header berikut dikonversi ke kasus campuran: `X-Forwarded-For`, `X-Forwarded-Proto`, dan `X-Forwarded-Port`, `host`, `id`, `X-Amzn-Trace-Id`, `Upgrade`, dan `Connection`. Semua nama header lainnya dalam huruf kecil.

Application Load Balancer dan Classic Load Balancer menerima header koneksi dari permintaan klien yang masuk setelah proxy respon kembali ke klien.

Ketika Application Load Balancers dan Classic Load Balancer menggunakan HTTP/1.1 menerima header `Expect: 100-Continue`, mereka segera merespons dengan HTTP/1.1 `100 Continue` tanpa menguji header panjang konten. Header permintaan `Expect: 100-Continue` tidak diteruskan ke targetnya.

Saat menggunakan HTTP/2, Application Load Balancers tidak mendukung header Expect: 100-Continue dari permintaan klien. Application Load Balancer tidak akan merespons dengan HTTP/2 100 Lanjutkan atau meneruskan header ini ke targetnya.

Batas header HTTP Header

Batas ukuran berikut untuk Application Load Balancer adalah batas keras yang tidak dapat diubah:

- Baris permintaan: 16 K
- Header tunggal: 16 K
- Seluruh header respons: 32 K
- Seluruh header permintaan: 64 K

Skema Load Balancer

Bila Anda membuat Load Balancer, Anda harus memilih apakah akan menjadikannya internal atau menghadap-internet.

Simpul Load Balancer menghadap-internet memiliki alamat IP publik. Nama DNS dari Load Balancer yang menghadap internet dapat dipecahkan secara publik ke alamat IP publik simpul tersebut. Oleh karena itu, Load Balancer yang menghadap internet dapat merutekan permintaan dari klien melalui internet.

Simpul penyeimbang beban internal hanya memiliki alamat IP privat. Nama DNS Load Balancer internal dapat dibuka secara publik ke alamat IP pribadi dari simpul. Oleh karena itu, Load Balancer internal hanya dapat merutekan permintaan dari klien dengan akses ke VPC untuk Load Balancer.

Baik Load Balancer yang menghadap-internet maupun internal merutekan permintaan ke target Anda menggunakan alamat IP pribadi. Oleh karena itu, target Anda tidak perlu alamat IP publik untuk menerima permintaan dari Load Balancer internal atau yang menghadap internet.

Jika aplikasi Anda memiliki beberapa tingkatan, Anda dapat merancang arsitektur yang menggunakan Load Balancer internal dan menghadap-internet. Misalnya, langkah ini berlaku jika aplikasi Anda menggunakan server web yang harus terhubung ke internet, dan server aplikasi yang hanya terhubung ke server web. Buat Load Balancer yang menghadap internet dan daftarkan server web dengannya. Buat Load Balancer internal dan daftarkan server aplikasi dengannya. Server web menerima permintaan dari Load Balancer menghadap internet dan mengirim permintaan untuk server aplikasi ke Load Balancer internal. Server aplikasi menerima permintaan dari Load Balancer internal.

MTU Jaringan untuk Load Balancer Anda

Unit transmisi maksimum (MTU) menentukan ukuran, dalam byte, untuk paket terbesar yang dapat dikirim melalui jaringan. Semakin besar MTU suatu koneksi, semakin banyak data yang dapat dilewatkan dalam satu paket tunggal. Frame Ethernet terdiri dari paket, atau data aktual yang Anda kirim, dan informasi overhead jaringan yang mengelilinginya. Lalu lintas yang dikirim melalui gateway internet memiliki MTU 1500. Ini berarti bahwa jika sebuah paket lebih dari 1500 byte, itu terfragmentasi untuk dikirim menggunakan beberapa frame, atau dijatuhkan jika Don't Fragment diatur dalam header IP.

Ukuran MTU pada node penyeimbang beban tidak dapat dikonfigurasi. Jumbo frame (9001 MTU) adalah standar di seluruh node penyeimbang beban untuk Application Load Balancers, Network Load Balancers, dan Classic Load Balancer. Gateway Load Balancers mendukung 8500 MTU. Untuk informasi selengkapnya, lihat [Unit transmisi maksimum \(MTU\)](#) di Panduan Pengguna untuk Penyeimbang Beban Gateway.

MTU jalur adalah ukuran paket maksimum yang didukung oleh jalur antara host asal dan host penerima. Penemuan MTU Jalur (PMTUD) digunakan untuk menentukan jalur MTU antara dua perangkat. Penemuan MTU Jalur sangat penting jika klien atau target tidak mendukung frame jumbo.

Jika suatu host mengirimkan paket yang lebih besar daripada MTU host penerima atau yang lebih besar daripada MTU perangkat di sepanjang jalur, host atau perangkat penerima akan mengembalikan pesan ICMP berikut: `Destination Unreachable: Fragmentation Needed and Don't Fragment was Set (Type 3, Code 4)`. Pesan ini menginstruksikan host pemancar untuk membagi muatan menjadi beberapa paket yang lebih kecil, dan mengirimkan ulang muatan tersebut.

Jika paket yang lebih besar dari ukuran MTU klien atau target antarmuka terus diturunkan, ada kemungkinan Penemuan Jalur MTU (PMTUD) tidak akan berfungsi. Untuk menghindari hal ini, pastikan Penemuan MTU Jalur berfungsi dari ujung ke ujung, dan bahwa Anda telah mengaktifkan frame jumbo pada klien dan target Anda. Untuk informasi lebih lanjut tentang Penemuan MTU Jalur dan mengaktifkan frame jumbo, lihat [Path MTU Discovery](#) di Panduan Pengguna Amazon EC2.

Memulai Elastic Load Balancing

Elastic Load Balancing mendukung Load Balancer berikut: Application Load Balancer, Network Load Balancer, Gateway Load Balancer, dan Classic Load Balancer. Anda dapat memilih jenis load balancer yang paling sesuai dengan kebutuhan Anda. Untuk informasi lebih lanjut, lihat [Perbandingan Produk](#).

Untuk demo konfigurasi load balancer umum, lihat [demo Elastic Load Balancing](#).

Jika Anda memiliki Classic Load Balancer yang sudah ada, Anda dapat bermigrasi ke Application Load Balancer atau Network Load Balancer. Untuk informasi selengkapnya, lihat [Migrasi Classic Load Balancer Anda](#).

Daftar Isi

- [Membuat Application Load Balancer](#)
- [Membuat Network Load Balancer](#)
- [Membuat Gateway Load Balancer](#)
- [Buat Classic Load Balancer](#)

Membuat Application Load Balancer

Untuk membuat Application Load Balancer menggunakan AWS Management Console, lihat [Memulai Application Load Balancers](#) dalam Panduan Pengguna untuk Application Load Balancers.

Untuk membuat Application Load Balancer menggunakan AWS CLI, lihat [Buat Application Load Balancer menggunakan AWS CLI](#) dalam Panduan Pengguna untuk Application Load Balancers.

Membuat Network Load Balancer

Untuk membuat Network Load Balancer menggunakan AWS Management Console, lihat [Memulai Network Load Balancers](#) dalam Panduan pengguna untuk Network Load Balancers.

Untuk membuat Network Load Balancer menggunakan AWS CLI, lihat [Buat Network Load Balancer menggunakan AWS CLI](#) dalam Panduan pengguna untuk Network Load Balancers.

Membuat Gateway Load Balancer

Untuk membuat Gateway Load Balancer menggunakan AWS Management Console, lihat [Memulai Gateway Load Balancers](#) dalam Panduan Pengguna untuk Gateway Load Balancers.

Untuk membuat Gateway Load Balancer menggunakan AWS CLI, lihat [Memulai dengan Gateway Load Balancers menggunakan AWS CLI](#) dalam Panduan Pengguna untuk Gateway Load Balancers.

Buat Classic Load Balancer

Untuk membuat Classic Load Balancer menggunakan AWS Management Console, lihat [Buat Classic Load Balancer](#) dalam Panduan Pengguna untuk Classic Load Balancers.

Keamanan dalam Elastic Load Balancing

Keamanan cloud di AWS merupakan prioritas tertinggi. Sebagai pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara AWS dan Anda. [Model tanggung jawab bersama](#) menjelaskan hal ini sebagai keamanan cloud dan keamanan dalam cloud:

- Keamanan cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di Cloud AWS. AWS juga menyediakan layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi secara rutin efektivitas keamanan kami sebagai bagian dari [program kepatuhan AWS](#). Untuk mempelajari program kepatuhan yang berlaku di Elastic Load Balancing, lihat [AWS layanan dalam lingkup dengan program kepatuhan](#).
- Keamanan dalam cloud – Tanggung jawab Anda ditentukan oleh layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Penyeimbangan Muatan Elastis. Dokumentasi ini menunjukkan kepada Anda cara mengonfigurasi Penyeimbangan Muatan Elastis untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan layanan AWS lain yang membantu Anda memantau dan mengamankan sumber daya Elastic Load Balancing.

Dengan [Gateway Load Balancer](#), Anda bertanggung jawab untuk memilih dan menerapkan kualifikasi perangkat lunak dari vendor alat. Anda harus mempercayai perangkat lunak alat untuk memeriksa atau memodifikasi lalu lintas dari penyeimbang beban, yang beroperasi pada lapisan 3 model Open Systems Interkoneksi (OSI), lapisan jaringan. Vendor alat yang terdaftar sebagai [Mitra Elastic Load Balancing](#) telah terintegrasi dan memenuhi syarat perangkat lunak alat mereka dengan AWS. Anda dapat menempatkan tingkat kepercayaan yang lebih tinggi pada perangkat lunak alat dari vendor dalam daftar ini. Namun, AWS tidak menjamin keamanan atau keandalan perangkat lunak dari vendor-vendor ini.

Daftar Isi

- [Perlindungan data dalam Elastic Load Balancing](#)
- [Manajemen identitas dan akses untuk Penyeimbangan Beban Elastis](#)

- [Validasi Kepatuhan untuk Elastic Load Balancing](#)
- [Ketahanan pada Elastic Load Balancing](#)
- [Keamanan Infrastruktur dalam Elastic Load Balancing](#)
- [Akses Elastic Load Balancing menggunakan endpoint antarmuka \(AWS PrivateLink\)](#)

Perlindungan data dalam Elastic Load Balancing

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data dalam Elastic Load Balancing. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam [Pertanyaan Umum Privasi Data](#). Lihat informasi tentang perlindungan data di Eropa di pos blog [Model Tanggung Jawab Bersama dan GDPR AWS](#) di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi yang lebih lengkap tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Elastic Load Balancing atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi diam

Jika Anda mengaktifkan enkripsi sisi server dengan kunci enkripsi terkelola Amazon S3 (SSE-S3) untuk bucket S3 Anda untuk log akses Elastic Load Balancing, Elastic Load Balancing secara otomatis mengenkripsi setiap file log akses sebelum disimpan dalam ember S3 Anda. Elastic Load Balancing juga mendekripsi file log akses ketika Anda mengaksesnya. Setiap file log dienkripsi dengan kunci unik, yang dienkripsi dengan kunci KMS yang diputar secara teratur.

Enkripsi dalam bergerak

Elastic Load Balancing menyederhanakan proses membangun aplikasi web yang aman dengan memangkas lalu lintas HTTPS dan TLS dari klien pada load balancer. Load Balancer melakukan pekerjaan mengenkripsi dan mendekripsi lalu lintas, alih-alih mengharuskan setiap instans EC2 untuk menangani pekerjaan untuk pemangkas TLS. Bila Anda mengkonfigurasi pendengar aman, Anda menentukan suite penyandian dan versi protokol yang didukung oleh aplikasi Anda, dan sertifikat server untuk menginstal pada Load Balancer Anda. Anda dapat menggunakan AWS Certificate Manager (ACM) atau AWS Identity and Access Management (IAM) untuk mengelola sertifikat server Anda. Application Load Balancer mendukung pendengar HTTPS. Network Load Balancer mendukung pendengar TLS. Classic Load Balancers mendukung pendengar HTTPS dan TLS.

Manajemen identitas dan akses untuk Penyeimbangan Beban Elastis

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya Elastic Load Balancing. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Daftar Isi

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana Elastic Load Balancing bekerja dengan IAM](#)
- [Izin API Penyeimbangan Beban Elastis](#)
- [Izin Elastic Load Balancing API untuk menandai sumber daya selama pembuatan](#)
- [Peran terkait layanan Penyeimbangan Beban Elastis](#)
- [AWS kebijakan terkelola untuk Elastic Load Balancing](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Elastic Load Balancing.

Pengguna layanan - Jika Anda menggunakan layanan Elastic Load Balancing untuk melakukan pekerjaan Anda, administrator Anda memberi Anda kredensi dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak fitur Elastic Load Balancing untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda.

Administrator layanan - Jika Anda bertanggung jawab atas sumber daya Elastic Load Balancing di perusahaan Anda, Anda mungkin memiliki akses penuh ke Elastic Load Balancing. Tugas Anda adalah menentukan fitur dan sumber daya Elastic Load Balancing mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM.

Administrator IAM - Jika Anda seorang administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke Elastic Load Balancing.

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensial identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensi Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani permintaan AWS API](#) di Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat [Autentikasi multi-faktor](#) dalam Panduan Pengguna AWS IAM Identity Center dan [Menggunakan autentikasi multi-faktor \(MFA\) dalam AWS](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Identitas gabungan

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat [Apakah itu Pusat Identitas IAM?](#) dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, kami merekomendasikan untuk mengandalkan kredensial sementara, bukan membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan tertentu yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami merekomendasikan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat [Merotasi kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang](#) dalam Panduan Pengguna IAM.

[Grup IAM](#) adalah identitas yang menentukan sekumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin bagi beberapa pengguna sekaligus. Grup mempermudah manajemen izin untuk sejumlah besar pengguna sekaligus. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin ke grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat [Kapan harus membuat pengguna IAM \(bukan peran\)](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan [beralih peran](#). Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat [Menggunakan peran IAM](#) dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi – Untuk menetapkan izin ke identitas terfederasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberi izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) dalam Panduan Pengguna IAM. Jika menggunakan Pusat Identitas IAM, Anda harus mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM akan mengorelasikan set izin ke peran dalam IAM. Untuk informasi tentang set izin, lihat [Set izin](#) dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (prinsipal tepercaya) di akun lain untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda memanggil suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.
 - Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan

AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses maju](#).

- Peran layanan – Peran layanan adalah [peran IAM](#) yang dijalankan oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan dalam instans Amazon EC2](#) dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat [Kapan harus membuat peran IAM \(bukan pengguna\)](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan menentukan apakah permintaan diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat [Gambaran umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk melakukan operasinya. Misalnya, anggaplah Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan yang dikelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam Akun AWS. Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan yang dikelola atau kebijakan inline, lihat [Memilih antara kebijakan yang dikelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus

[menentukan prinsipal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat [Gambaran umum daftar kontrol akses \(ACL\)](#) dalam Panduan Developer Amazon Simple Storage Service.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Jenis-jenis kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh jenis kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas ke entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya tentang batasan izin, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di organisasi, Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations .
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara programatis untuk peran atau pengguna terfederasi. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau

peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini akan menggantikan pemberian izin. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan Pengguna IAM.

Bagaimana Elastic Load Balancing bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Elastic Load Balancing, pelajari fitur IAM apa saja yang tersedia untuk digunakan dengan Elastic Load Balancing.

Fitur IAM yang dapat Anda gunakan dengan Elastic Load Balancing

Fitur IAM	Dukungan Elastic Load Balancing
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACL	Tidak
ABAC (tanda dalam kebijakan)	Ya
Kredensial sementara	Ya
Izin prinsipal	Ya
Peran layanan	Tidak
Peran terkait layanan	Ya

Kebijakan berbasis identitas untuk Elastic Load Balancing

Mendukung kebijakan berbasis identitas Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan JSON IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya dalam Elastic Load Balancing

Mendukung kebijakan berbasis sumber daya Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan prinsipal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai prinsipal dalam kebijakan berbasis sumber daya. Menambahkan prinsipal akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, administrator IAM di akun tepercaya juga harus memberikan izin entitas utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin dengan melampirkan kebijakan berbasis identitas kepada

entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk Elastic Load Balancing

Mendukung tindakan kebijakan

Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar tindakan Elastic Load Balancing, lihat [Tindakan yang ditentukan oleh Elastic Load Balancing](#) dalam Referensi Otorisasi Layanan.

Tindakan kebijakan dalam Elastic Load Balancing menggunakan awalan berikut sebelum tindakan:

```
elasticloadbalancing
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

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

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (*). Sebagai contoh, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut:

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

Untuk daftar lengkap API tindakan untuk Penyeimbang Beban Elastis, lihat dokumentasi berikut:

- [Aplikasi Load Balancers, Network Load Balancers, dan Gateway Load Balancers - Referensi API versi 2015-12-01](#)
- Penyeimbang Beban Klasik — [Versi referensi API 2012-06-01](#)

Untuk informasi lebih lanjut tentang izin yang diperlukan oleh setiap tindakan Penyeimbangan Beban Elastis, lihat [Izin API Penyeimbangan Beban Elastis](#).

Sumber daya kebijakan untuk Elastic Load Balancing

Mendukung sumber daya kebijakan Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Beberapa tindakan Elastic Load Balancing API mendukung beberapa sumber daya. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARN dengan koma.

```
"Resource": [  
  "resource1",
```

```
"resource2"  
]
```

Untuk melihat daftar jenis sumber daya Elastic Load Balancing dan ARNnya, lihat Sumber Daya yang [ditetapkan oleh Elastic Load Balancing](#) dalam Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditetapkan oleh Elastic Load Balancing](#).

Kunci kondisi kebijakan untuk Elastic Load Balancing

Mendukung kunci kondisi kebijakan khusus layanan	Ya
--	----

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen Condition (atau blok Condition) akan memungkinkan Anda menentukan kondisi yang menjadi dasar suatu pernyataan berlaku. Elemen Condition bersifat opsional. Anda dapat membuat ekspresi bersyarat yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen Condition dalam sebuah pernyataan, atau beberapa kunci dalam elemen Condition tunggal, maka AWS akan mengevaluasinya menggunakan operasi AND logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan OR operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat [Elemen kebijakan IAM: variabel dan tag](#) dalam Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

Untuk melihat daftar tombol kondisi Elastic Load Balancing, lihat tombol Kondisi [untuk Elastic Load Balancing](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang ditetapkan oleh Elastic Load Balancing](#).

Kunci syarat **elasticloadbalancing:ResourceTag**

Kunci ketentuan `elasticloadbalancing:ResourceTag`/*kunci* adalah ketentuan khusus untuk Penyeimbangan Beban Elastis. Tindakan berikut mendukung kunci ketentuan ini:

Versi API 2015-12-01

- AddTags
- CreateListener
- CreateLoadBalancer
- DeleteLoadBalancer
- DeleteTargetGroup
- DeregisterTargets
- ModifyLoadBalancerAttributes
- ModifyTargetGroup
- ModifyTargetGroupAttributes
- RegisterTargets
- RemoveTags
- SetIpAddressType
- SetSecurityGroups
- SetSubnets

Versi API 2012-06-01.

- AddTags
- ApplySecurityGroupsToLoadBalancer
- AttachLoadBalancersToSubnets
- ConfigureHealthCheck
- CreateAppCookieStickinessPolicy
- CreateLBCookieStickinessPolicy
- CreateLoadBalancer
- CreateLoadBalancerListeners

- `CreateLoadBalancerPolicy`
- `DeleteLoadBalancer`
- `DeleteLoadBalancerListeners`
- `DeleteLoadBalancerPolicy`
- `DeregisterInstancesFromLoadBalancer`
- `DetachLoadBalancersFromSubnets`
- `DisableAvailabilityZonesForLoadBalancer`
- `EnableAvailabilityZonesForLoadBalancer`
- `ModifyLoadBalancerAttributes`
- `RegisterInstancesWithLoadBalancer`
- `RemoveTags`
- `SetLoadBalancerListenerSSLCertificate`
- `SetLoadBalancerPoliciesForBackendServer`
- `SetLoadBalancerPoliciesOfListener`

Kunci syarat **`elasticloadbalancing:ListenerProtocol`**

Kunci `elasticloadbalancing:ListenerProtocol` kondisi dapat digunakan untuk kondisi yang menentukan jenis pendengar yang dapat dibuat dan digunakan. Tindakan berikut mendukung kunci ketentuan ini:

Versi API 2015-12-01

- `CreateListener`
- `ModifyListener`

Versi API 2012-06-01.

- `CreateLoadBalancer`
- `CreateLoadBalancerListeners`

Kebijakan ini tersedia untuk Application Load Balancers, Network Load Balancers, dan Classic Load Balancer. Berikut ini adalah contoh kebijakan yang hanya memungkinkan pengguna untuk memilih salah satu protokol yang ditentukan untuk listener mereka.

Protokol yang didukung:

- HTTPS
- HTTP
- TCP
- SSL
- TLS
- UDP
- TCP_UDP

```
"Version": "2015-12-01",
  "Statement": {"Effect": "Allow",
    "Action": [
      "elasticloadbalancing:CreateListener",
      "elasticloadbalancing:ModifyListener"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals":{
        "elasticloadbalancing:ListenerProtocol": [
          "HTTPS",
          "TLS"
        ]
      }
    }
  }
```

Kunci syarat **elasticloadbalancing:SecurityPolicy**

Kunci `elasticloadbalancing:SecurityPolicy` kondisi dapat digunakan untuk kondisi yang menentukan dan menegakkan kebijakan keamanan tertentu pada penyeimbang beban. Tindakan berikut mendukung kunci ketentuan ini:

Versi API 2015-12-01

- `CreateListener`
- `ModifyListener`

Versi API 2012-06-01.

- `CreateLoadBalancerPolicy`
- `SetLoadBalancerPoliciesOfListener`

Kebijakan ini tersedia untuk Application Load Balancers, Network Load Balancers dan Classic Load Balancer. Berikut ini adalah contoh kebijakan yang hanya memungkinkan pengguna untuk memilih salah satu kebijakan keamanan yang ditentukan untuk penyeimbang beban mereka.

```
"Resource": [
  "Version": "2015-12-01",
    "Statement": {"Effect": "Allow",
      "Action": [
        "elasticloadbalancing:CreateListener",
        "elasticloadbalancing:ModifyListener"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals":{
          "elasticloadbalancing:SecurityPolicy": [
            "ELBSecurityPolicy-TLS13-1-2-2021-06",
            "ELBSecurityPolicy-TLS13-1-2-Res-2021-06",
            "ELBSecurityPolicy-TLS13-1-1-2021-06"
          ]
        }
      }
    }
  ]
}
```

Kunci syarat `elasticloadbalancing:Scheme`

Kunci `elasticloadbalancing:Scheme` kondisi dapat digunakan untuk kondisi yang menentukan skema mana yang dapat dipilih selama pembuatan penyeimbang beban. Tindakan berikut mendukung kunci ketentuan ini:

Versi API 2015-12-01

- `CreateLoadBalancer`

Versi API 2012-06-01.

- `CreateLoadBalancer`

Kebijakan ini tersedia untuk Application Load Balancers, Network Load Balancers, dan Classic Load Balancer. Berikut ini adalah contoh kebijakan yang hanya memungkinkan pengguna untuk memilih salah satu skema yang ditentukan untuk penyeimbang beban mereka.

```
"Version": "2015-12-01",
  "Statement": [{"Effect": "Allow",
    "Action": "elasticloadbalancing:CreateLoadBalancer",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "elasticloadbalancing:Scheme": "internal"
      }
    }
  }
]
```

Kunci syarat **elasticloadbalancing:Subnet**

Important

Elastic Load Balancing menerima semua kapitalisasi ID Subnet. Namun, pastikan untuk menggunakan operator kondisi case insensitive yang sesuai, misalnya `StringEqualsIgnoreCase`.

Kunci `elasticloadbalancing:Subnet` kondisi dapat digunakan untuk kondisi yang menentukan subnet mana yang dapat dibuat dan dilampirkan ke penyeimbang beban. Tindakan berikut mendukung kunci ketentuan ini:

Versi API 2015-12-01

- `CreateLoadBalancer`
- `SetSubnets`

Versi API 2012-06-01.

- `CreateLoadBalancer`
- `AttachLoadBalancerToSubnets`

Kebijakan ini tersedia untuk Application Load Balancers, Network Load Balancers, Gateway Load Balancers dan Classic Load Balancer. Berikut ini adalah contoh kebijakan yang hanya

memungkinkan pengguna untuk memilih salah satu subnet yang ditentukan untuk penyeimbang beban mereka.

```
"Version": "2015-12-01",
  "Statement": {"Effect": "Allow",
    "Action": [
      "elasticloadbalancing:CreateLoadBalancer",
      "elasticloadbalancing:SetSubnets"
    ],
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEqualsIgnoreCase":{
        "elasticloadbalancing:Subnet": [
          "subnet-01234567890abcdef",
          "subnet-01234567890abcdeg "
        ]
      }
    }
  }
```

Kunci syarat **elasticloadbalancing:SecurityGroup**

Important

Elastic Load Balancing menerima semua kapitalisasi ID. SecurityGroup Namun, pastikan untuk menggunakan operator kondisi case insensitive yang sesuai, misalnya `StringEqualsIgnoreCase`.

Kunci `elasticloadbalancing:SecurityGroup` kondisi dapat digunakan untuk kondisi yang menentukan kelompok keamanan mana yang dapat diterapkan ke penyeimbang beban. Tindakan berikut mendukung kunci ketentuan ini:

Versi API 2015-12-01

- `CreateLoadBalancer`
- `SetSecurityGroups`

Versi API 2012-06-01.

- `CreateLoadBalancer`

- `ApplySecurityGroupsToLoadBalancer`

Kebijakan ini tersedia untuk Application Load Balancers, Network Load Balancers dan Classic Load Balancer. Berikut ini adalah contoh kebijakan yang hanya memungkinkan pengguna untuk memilih salah satu grup keamanan yang ditentukan untuk penyeimbang beban mereka.

```
"Version": "2015-12-01",
  "Statement": [{"Effect": "Allow",
    "Action": [
      "elasticloadbalancing:CreateLoadBalancer",
      "elasticloadbalancing:SetSecurityGroup"
    ],
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEqualsIgnoreCase": {
        "elasticloadbalancing:SecurityGroup": [
          "sg-51530134",
          "sg-51530144",
          "sg-51530139"
        ]
      }
    }
  ]
}
```

ACL dalam Elastic Load Balancing

Mendukung ACL

Tidak

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Elastic Load Balancing

Mendukung ABAC (tanda dalam kebijakan)

Ya

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna

atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tag milik prinsipal cocok dengan tag yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi saat manajemen kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di [elemen kondisi](#) dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat [Apa itu ABAC?](#) dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) dalam Panduan Pengguna IAM.

Menggunakan kredensial sementara dengan Elastic Load Balancing

Mendukung penggunaan kredensial sementara Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensial sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensi sementara, lihat [Layanan AWS yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Anda menggunakan kredensial sementara jika Anda masuk AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, ketika Anda mengakses AWS menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut secara otomatis membuat kredensi sementara. Anda juga akan secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai seorang pengguna lalu beralih peran. Untuk informasi selengkapnya tentang peralihan peran, lihat [Peralihan peran \(konsol\)](#) dalam Panduan Pengguna IAM.

Anda dapat membuat kredensial sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensial sementara tersebut untuk mengakses.

AWS AWS merekomendasikan agar Anda menghasilkan kredensial sementara secara dinamis alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat [Kredensial keamanan sementara di IAM](#).

Izin utama lintas layanan untuk Elastic Load Balancing

Mendukung sesi akses maju (FAS)	Ya
---------------------------------	----

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian menginisiasi tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses maju](#).

Peran servis untuk Elastic Load Balancing

Mendukung peran layanan	Tidak
-------------------------	-------

Peran layanan adalah sebuah [peran IAM](#) yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

Peran terkait layanan untuk Elastic Load Balancing

Mendukung peran terkait layanan	Ya
---------------------------------	----

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke Layanan AWS. Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Akun AWS Anda dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

Untuk detail tentang membuat atau mengelola peran terkait layanan Elastic Load Balancing, lihat [Peran terkait layanan Penyeimbangan Beban Elastis](#)

Izin API Penyeimbangan Beban Elastis

Anda harus memberikan izin kepada pengguna untuk memanggil tindakan Elastic Load Balancing API yang mereka butuhkan. Selain itu, untuk beberapa tindakan Elastic Load Balancing, Anda harus memberikan izin kepada pengguna untuk memanggil tindakan tertentu dari Amazon EC2 API.

Izin yang diperlukan untuk API 2015-12-01

Saat memanggil tindakan berikut dari API 2015-12-01, Anda harus memberikan izin kepada pengguna untuk memanggil tindakan yang ditentukan.

CreateLoadBalancer

- `elasticloadbalancing:CreateLoadBalancer`
- `ec2:DescribeAccountAttributes`
- `ec2:DescribeAddresses`
- `ec2:DescribeInternetGateways`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `iam:CreateServiceLinkedRole`

CreateTargetGroup

- `elasticloadbalancing:CreateTargetGroup`
- `ec2:DescribeInternetGateways`
- `ec2:DescribeVpcs`

RegisterTargets

- `elasticloadbalancing:RegisterTargets`
- `ec2:DescribeInstances`
- `ec2:DescribeInternetGateways`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`

SetIpAddressType

- `elasticloadbalancing:SetIpAddressType`
- `ec2:DescribeSubnets`

SetSubnets

- `elasticloadbalancing:SetSubnets`
- `ec2:DescribeSubnets`

Izin yang diperlukan untuk API 2012-06-01

Saat memanggil tindakan berikut dari API 2012-06-01, Anda harus memberikan izin kepada pengguna untuk memanggil tindakan yang ditentukan.

ApplySecurityGroupsToLoadBalancer

- `elasticloadbalancing:ApplySecurityGroupsToLoadBalancer`
- `ec2:DescribeAccountAttributes`
- `ec2:DescribeSecurityGroups`

AttachLoadBalancerToSubnets

- `elasticloadbalancing:AttachLoadBalancerToSubnets`
- `ec2:DescribeSubnets`

CreateLoadBalancer

- `elasticloadbalancing>CreateLoadBalancer`
- `ec2:CreateSecurityGroup`
- `ec2:DescribeAccountAttributes`
- `ec2:DescribeInternetGateways`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`
- `iam:CreateServiceLinkedRole`

DeregisterInstancesFromLoadBalancer

- `elasticloadbalancing:DeregisterInstancesFromLoadBalancer`
- `ec2:DescribeClassicLinkInstances`

- `ec2:DescribeInstances`

`DescribeInstanceHealth`

- `elasticloadbalancing:DescribeInstanceHealth`
- `ec2:DescribeClassicLinkInstances`
- `ec2:DescribeInstances`

`DescribeLoadBalancers`

- `elasticloadbalancing:DescribeLoadBalancers`
- `ec2:DescribeSecurityGroups`

`DisableAvailabilityZonesForLoadBalancer`

- `elasticloadbalancing:DisableAvailabilityZonesForLoadBalancer`
- `ec2:DescribeAccountAttributes`
- `ec2:DescribeInternetGateways`
- `ec2:DescribeVpcs`

`EnableAvailabilityZonesForLoadBalancer`

- `elasticloadbalancing:EnableAvailabilityZonesForLoadBalancer`
- `ec2:DescribeAccountAttributes`
- `ec2:DescribeInternetGateways`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcs`

`RegisterInstancesWithLoadBalancer`

- `elasticloadbalancing:RegisterInstancesWithLoadBalancer`
- `ec2:DescribeAccountAttributes`
- `ec2:DescribeClassicLinkInstances`
- `ec2:DescribeInstances`
- `ec2:DescribeVpcClassicLink`

Izin Elastic Load Balancing API untuk menandai sumber daya selama pembuatan

Agar pengguna dapat menandai sumber daya selama pembuatan, mereka harus memiliki izin untuk menggunakan tindakan yang membuat sumber

daya, seperti `elasticloadbalancing:CreateLoadBalancer` atau `elasticloadbalancing:CreateTargetGroup`. Jika tag ditentukan dalam tindakan pembuatan sumber daya, otorisasi tambahan diperlukan pada `elasticloadbalancing:AddTags` tindakan untuk memverifikasi apakah pengguna memiliki izin untuk menerapkan tag ke sumber daya yang sedang dibuat. Oleh karena itu, para pengguna juga harus memiliki izin eksplisit untuk menggunakan tindakan `elasticloadbalancing:AddTags`.

Dalam definisi kebijakan IAM untuk `elasticloadbalancing:AddTags` tindakan, Anda dapat menggunakan `Condition` elemen dengan kunci `elasticloadbalancing:CreateAction` kondisi untuk memberikan izin penandaan pada tindakan yang membuat sumber daya.

Contoh berikut menunjukkan kebijakan yang memungkinkan pengguna membuat grup target dan menerapkan tag apa pun pada mereka selama pembuatan. Pengguna tidak diizinkan untuk menandai sumber daya yang ada (mereka tidak dapat memanggil `elasticloadbalancing:AddTags` tindakan secara langsung).

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

Demikian pula, kebijakan berikut memungkinkan pengguna untuk membuat penyeimbang beban dan menerapkan tag selama pembuatan. Pengguna tidak diizinkan untuk menandai sumber daya yang ada (mereka tidak dapat memanggil `elasticloadbalancing:AddTags` tindakan secara langsung).

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

Tindakan `elasticloadbalancing:AddTags` akan dievaluasi hanya jika tanda diterapkan selama tindakan pembuatan sumber daya. Oleh karena itu, seorang pengguna yang memiliki izin untuk membuat sumber daya (dengan asumsi tidak ada syarat untuk pemberian tanda) tidak memerlukan izin untuk menggunakan tindakan `elasticloadbalancing:AddTags` jika tidak ada tanda yang ditentukan dalam permintaan. Akan tetapi, jika pengguna tersebut mencoba untuk membuat sumber daya dengan tanda, maka permintaan akan gagal jika pengguna tidak memiliki izin untuk menggunakan tindakan `elasticloadbalancing:AddTags`.

Peran terkait layanan Penyeimbangan Beban Elastis

Penyeimbangan Beban Elastis menggunakan peran terkait layanan untuk izin yang diperlukan untuk memanggil layanan AWS lain atas nama Anda. Untuk informasi lebih lanjut, lihat [Menggunakan peran tertaut layanan](#) dalam Panduan Pengguna IAM.

Izin yang diberikan oleh peran tertaut layanan

Elastic Load Balancing menggunakan peran terkait layanan bernama `AWSServiceRoleForElasticLoadBalancing` untuk memanggil tindakan berikut atas nama Anda:

- `ec2:AssignIpv6Addresses`
- `ec2:AssignPrivateIpAddresses`
- `ec2:AssociateAddress`
- `ec2:AttachNetworkInterface`
- `ec2:AuthorizeSecurityGroupIngress`
- `ec2:CreateNetworkInterface`
- `ec2:CreateSecurityGroup`
- `ec2>DeleteNetworkInterface`
- `ec2:DescribeAccountAttributes`
- `ec2:DescribeAddresses`
- `ec2:DescribeClassicLinkInstances`
- `ec2:DescribeCoipPools`
- `ec2:DescribeInstances`
- `ec2:DescribeInternetGateways`
- `ec2:DescribeNetworkInterfaces`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcClassicLink`
- `ec2:DescribeVpcPeeringConnections`
- `ec2:DescribeVpcs`
- `ec2:DetachNetworkInterface`
- `ec2:DisassociateAddress`

- `ec2:GetCoipPoolUsage`
- `ec2:ModifyNetworkInterfaceAttribute`
- `ec2:ReleaseAddress`
- `ec2:UnassignIpv6Addresses`
- `logs:CreateLogDelivery`
- `logs>DeleteLogDelivery`
- `logs:GetLogDelivery`
- `logs>ListLogDeliveries`
- `logs:UpdateLogDelivery`
- `outposts:GetOutpostInstanceTypes`

`AWSServiceRoleForElasticLoadBalancing` mempercayai `elasticloadbalancing.amazonaws.com` layanan untuk mengambil peran.

Membuat peran terkait layanan

Anda tidak perlu membuat `AWSServiceRoleForElasticLoadBalancing` peran secara manual. Elastic Load Balancing menciptakan peran ini untuk Anda saat Anda membuat penyeimbang beban atau grup target.

Agar Penyeimbangan Beban Elastis membuat peran yang terkait dengan layanan atas nama Anda, Anda harus memiliki izin yang diperlukan. Untuk informasi selengkapnya, lihat [Izin peran tertaut layanan](#) dalam Panduan Pengguna IAM.

Jika Anda membuat penyeimbang beban sebelum 11 Januari 2018, Elastic Load Balancing `AWSServiceRoleForElasticLoadBalancing` dibuat di AWS akun Anda. Untuk informasi selengkapnya, lihat [Peran baru muncul di AWS akun saya](#) di Panduan Pengguna IAM.

Mengedit peran terkait layanan

Anda dapat mengedit deskripsi `AWSServiceRoleForElasticLoadBalancing` menggunakan IAM. Untuk informasi lebih lanjut, lihat [Mengedit peran terkait layanan](#) dalam Panduan Pengguna IAM.

Hapus peran tertaut layanan

Jika Anda tidak perlu lagi menggunakan Elastic Load Balancing, kami sarankan Anda menghapusnya. `AWSServiceRoleForElasticLoadBalancing`

Anda dapat menghapus peran terkait layanan ini hanya setelah Anda menghapus semua penyeimbang beban di akun Anda. AWS Hal ini untuk memastikan bahwa Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses penyeimbang beban Anda. Untuk informasi selengkapnya, lihat [Menghapus Penyeimbang Beban Aplikasi](#), [Menghapus Penyeimbang Beban Jaringan](#), dan [Menghapus Penyeimbang Beban Klasik](#).

Anda dapat menggunakan konsol IAM, IAM CLI, atau IAM API untuk menghapus peran terkait-layanan. Untuk informasi selengkapnya, lihat [Menghapus peran tertaut layanan](#) dalam Panduan Pengguna IAM.

Setelah Anda menghapus `AWSServiceRoleForElasticLoadBalancing`, Elastic Load Balancing akan membuat peran lagi jika Anda membuat penyeimbang beban.

AWS kebijakan terkelola untuk Elastic Load Balancing

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan [kebijakan yang dikelola pelanggan](#) yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat [AWS kebijakan yang dikelola](#) dalam Panduan Pengguna IAM.

AWS kebijakan terkelola: `AWSElasticLoadBalancingClassicServiceRolePolicy`

Kebijakan ini mencakup semua izin yang diperlukan Elastic Load Balancing (Classic Load Balancer) untuk AWS menghubungi layanan lain atas nama Anda. Peran terkait layanan sudah ditentukan sebelumnya. Dengan peran yang telah ditentukan, Anda tidak perlu menambahkan izin yang diperlukan secara manual untuk Elastic Load Balancing untuk menyelesaikan tindakan atas nama Anda. Anda tidak dapat melampirkan, melepaskan, memodifikasi, atau menghapus kebijakan ini.

Untuk melihat izin kebijakan ini, lihat [AWSElasticLoadBalancingClassicServiceRolePolicy](#) di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSElasticLoadBalancingServiceRolePolicy

Kebijakan ini mencakup semua izin yang diperlukan Elastic Load Balancing untuk memanggil layanan AWS lain atas nama Anda. Peran terkait layanan sudah ditentukan sebelumnya. Dengan peran yang telah ditentukan, Anda tidak perlu menambahkan izin yang diperlukan secara manual untuk Elastic Load Balancing untuk menyelesaikan tindakan atas nama Anda. Anda tidak dapat melampirkan, melepaskan, memodifikasi, atau menghapus kebijakan ini.

Untuk melihat izin kebijakan ini, lihat [AWSElasticLoadBalancingServiceRolePolicy](#) di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: ElasticLoadBalancingFullAccess

Kebijakan ini memberikan akses penuh ke layanan Elastic Load Balancing dan akses terbatas ke layanan lain melalui AWS Management Console.

Untuk melihat izin kebijakan ini, lihat [ElasticLoadBalancingFullAccess](#) di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: ElasticLoadBalancingReadOnly

Kebijakan ini menyediakan akses read-only ke Elastic Load Balancing dan layanan dependen.

Untuk melihat izin kebijakan ini, lihat [ElasticLoadBalancingReadOnly](#) di Referensi Kebijakan AWS Terkelola.

Elastic Load Balancing memperbarui kebijakan terkelola AWS

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Elastic Load Balancing sejak layanan ini mulai melacak perubahan ini.

Perubahan	Deskripsi	Tanggal
AWS kebijakan terkelola: ElasticLoadBalancingFullAccess - Perbarui ke kebijakan yang ada.	Elastic Load Balancing menambahkan tindakan baru untuk memberikan izin menggunakan pergeseran zona. Tindakan ini ditambahkan ke kebijakan akses penuh Elastic Load Balancing	28 November 2022

Perubahan	Deskripsi	Tanggal
	. Hal ini terkait dengan operasi <code>arc-zonal-shift:*</code> API.	
AWS kebijakan terkelola: ElasticLoadBalancingReadOnly - Perbarui ke kebijakan yang ada.	Elastic Load Balancing menambahkan tindakan baru untuk memberikan izin menggunakan pergeseran zona. Tindakan ini ditambahkan ke kebijakan baca saja Elastic Load Balancing. Hal ini terkait dengan <code>arc-zonal-shift:GetManagedResource</code> , <code>arc-zonal-shift:ListManagedResources</code> dan operasi <code>arc-zonal-shift:ListZonalShifts</code> API.	28 November 2022
AWS kebijakan terkelola : AWSElasticLoadBalancingServiceRolePolicy - Perbarui ke kebijakan yang ada.	Elastic Load Balancing menambahkan tindakan baru untuk memberikan izin menggunakan koneksi peering. Tindakan ini ditambahkan ke kebijakan peran terkait layanan, untuk bidang kontrol Elastic Load Balancing. Hal ini terkait dengan operasi <code>ec2:DescribeVpcPeeringConnections</code> API.	11 Oktober 2021
AWS kebijakan terkelola: ElasticLoadBalancingFullAccess - Perbarui ke kebijakan yang ada.	Elastic Load Balancing menambahkan tindakan baru untuk memberikan izin menggunakan koneksi peering. Tindakan ini ditambahkan ke kebijakan akses penuh Elastic Load Balancing . Hal ini terkait dengan operasi <code>ec2:DescribeVpcPeeringConnections</code> API.	11 Oktober 2021
AWS kebijakan terkelola : AWSElasticLoadBalancingClassicServiceRolePolicy - Perbarui ke kebijakan yang ada.	Elastic Load Balancing menambahkan kebijakan peran terkait layanan (untuk bidang kontrol) untuk Classic Load Balancer. Pembaruan ini untuk versi 2 (default).	7 Oktober 2019

Perubahan	Deskripsi	Tanggal
AWS kebijakan terkelola: ElasticLoadBalancingReadOnly	Menyediakan akses read-only ke Elastic Load Balancing dan layanan dependen. Ini adalah versi 1 (default).	20 September 2018
Elastic Load Balancing mulai melacak perubahan	Elastic Load Balancing mulai melacak perubahan untuk kebijakan yang AWS dikelola.	23 Juli 2021

Validasi Kepatuhan untuk Elastic Load Balancing

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat [Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS](#) dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#) .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#) .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- [Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

Note

Tidak semua memenuhi Layanan AWS syarat HIPAA. Untuk informasi selengkapnya, lihat [Referensi Layanan yang Memenuhi Syarat HIPAA](#).

- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.

- [AWS Panduan Kepatuhan Pelanggan](#) - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda serta untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).
- [Amazon GuardDuty](#) — Ini Layanan AWS mendeteksi potensi ancaman terhadap beban kerja Akun AWS, kontainer, dan data Anda dengan memantau lingkungan Anda untuk aktivitas mencurigakan dan berbahaya. GuardDuty dapat membantu Anda mengatasi berbagai persyaratan kepatuhan, seperti PCI DSS, dengan memenuhi persyaratan deteksi intrusi yang diamanatkan oleh kerangka kerja kepatuhan tertentu.
- [AWS Audit Manager](#)Ini Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan pada Elastic Load Balancing

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone Wilayah menyediakan beberapa Availability Zone yang terpisah dan terisolasi secara fisik, yang terhubung melalui jaringan latensi rendah, throughput tinggi, dan sangat redundan. Dengan Availability Zone, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis mengalami failover antar zona tanpa gangguan. Availability Zone lebih tersedia, memiliki toleransi kesalahan, dan dapat diskalakan dibandingkan dengan satu atau beberapa infrastruktur pusat data tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat [AWS Infrastruktur Global](#).

Selain Infrastruktur Global AWS, Elastic Load Balancing menyediakan beberapa fitur untuk membantu mendukung ketahanan data Anda.

- Pendistribusian lalu lintas yang masuk pada berbagai instans dalam satu Availability Zone atau beberapa Availability Zone.
- Anda dapat menggunakan AWS Global Accelerator dengan Load Balancers Aplikasi Anda untuk mendistribusikan lalu lintas yang masuk di beberapa load balancers dalam satu atau lebih Wilayah AWS. Untuk informasi selengkapnya, lihat [AWS Global Accelerator Panduan Pengembang](#)
- Amazon ECS memungkinkan Anda untuk menjalankan, menghentikan, dan mengelola kontainer Docker pada kluster instans EC2. Anda dapat mengkonfigurasi layanan Amazon ECS untuk menggunakan load balancer untuk mendistribusikan lalu lintas yang masuk di layanan dalam sebuah kluster. Untuk informasi lebih lanjut, lihat [Panduan Pengembang Amazon Elastic Container Service](#).

Keamanan Infrastruktur dalam Elastic Load Balancing

Sebagai layanan terkelola, Elastic Load Balancing dilindungi oleh keamanan jaringan AWS global. Lihat informasi tentang layanan keamanan AWS dan cara AWS melindungi infrastruktur di [Keamanan Cloud AWS](#). Untuk mendesain lingkungan AWS Anda dengan menggunakan praktik terbaik bagi keamanan infrastruktur, lihat [Perlindungan Infrastruktur](#) dalam Pilar Keamanan Kerangka Kerja Berarsitektur Baik AWS.

Anda menggunakan panggilan API yang dipublikasikan AWS untuk mengakses Elastic Load Balancing melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan pengguna utama IAM. Atau, Anda bisa menggunakan [AWS Security Token Service](#) (AWS STS) untuk membuat kredensial keamanan sementara untuk menandatangani permintaan.

Isolasi jaringan

Virtual Private Cloud (VPC) adalah jaringan virtual dalam area Anda yang diisolasi secara logis dalam AWS Cloud. Subnet adalah serangkaian alamat IP di VPC. Bila Anda membuat penyeimbang beban, Anda dapat menentukan satu atau lebih subnet untuk simpul penyeimbang beban. Anda dapat menggunakan contoh EC2 di subnet VPC Anda dan mendaftarkannya dengan penyeimbang beban Anda. Untuk informasi selengkapnya tentang membuat VPC, lihat [Panduan Pengguna Amazon VPC](#).

Bila Anda membuat penyeimbang beban di VPC, bisa dibuat menghadap-internal atau pun internal. Penyeimbang beban internal hanya dapat merutekan permintaan yang berasal dari klien dengan akses ke VPC untuk penyeimbang beban.

Penyeimbang beban Anda mengirimkan permintaan ke target yang terdaftar menggunakan alamat IP pribadi. Oleh karena itu, target Anda tidak perlu alamat IP publik untuk menerima permintaan dari penyeimbang beban.

Untuk memanggil Elastic Load Balancing API dari VPC Anda menggunakan alamat IP pribadi, gunakan [AWS PrivateLink](#). Untuk informasi selengkapnya, lihat [Akses Elastic Load Balancing menggunakan endpoint antarmuka \(AWS PrivateLink\)](#).

Pengontrolan lalu lintas jaringan

Pertimbangkan opsi berikut untuk mengamankan lalu lintas jaringan saat Anda menggunakan penyeimbang beban:

- Gunakan pendengar yang aman untuk mendukung komunikasi terenkripsi antara klien dan penyeimbang beban Anda. Application Load Balancer mendukung pendengar HTTPS. Network Load Balancer mendukung pendengar TLS. Classic Load Balancer mendukung pendengar HTTPS dan TLS. Anda dapat memilih dari kebijakan keamanan yang telah ditetapkan untuk penyeimbang beban Anda untuk menentukan rangkaian penyandian dan versi protokol yang didukung oleh aplikasi Anda. Anda dapat menggunakan AWS Certificate Manager (ACM) atau AWS Identity and Access Management (IAM) untuk mengelola sertifikat server yang diinstal pada penyeimbang beban Anda. Anda dapat menggunakan protokol Indikasi Nama Server (SNI) untuk melayani beberapa situs web aman menggunakan satu pendengar aman. SNI diaktifkan secara otomatis untuk penyeimbang beban Anda ketika Anda mengaitkan lebih dari satu sertifikat server dengan pendengar yang aman.
- Mengonfigurasi grup keamanan untuk Application Load Balancer dan Classic Load Balancer untuk menerima lalu lintas hanya dari klien tertentu. Kelompok-kelompok keamanan ini harus memungkinkan lalu lintas masuk dari klien pada port pendengar dan lalu lintas keluar ke klien.

- Mengonfigurasi grup keamanan untuk instans Amazon EC2 Anda untuk menerima lalu lintas hanya dari penyeimbang beban. Kelompok-kelompok keamanan ini harus memungkinkan lalu lintas masuk dari penyeimbang beban pada port pendengar dan port pemeriksaan kesehatan.
- Konfigurasi Application Load Balancer Anda untuk mengotentikasi pengguna dengan aman melalui penyedia identitas atau menggunakan identitas perusahaan. Untuk informasi lebih lanjut, lihat [Mengautentikasi pengguna menggunakan Application Load Balancer](#).
- Gunakan [AWS WAF](#) dengan Application Load Balancer Anda untuk mengizinkan atau memblokir permintaan berdasarkan aturan dalam daftar kontrol akses web (web ACL).

Akses Elastic Load Balancing menggunakan endpoint antarmuka (AWS PrivateLink)

Anda dapat membuat koneksi pribadi antara cloud pribadi virtual (Virtual Private Cloud/VPC) dan Elastic Load Balancing API dengan membuat titik akhir antarmuka VPC. Anda dapat menggunakan koneksi ini untuk menghubungi Elastic Load Balancing dari VPC Anda tanpa mengharuskan Anda melampirkan gateway internet, instans NAT, atau koneksi VPN ke VPC Anda. Endpoint menyediakan konektivitas yang dapat diandalkan dan terukur ke Elastic Load Balancing API, versi 2015-12-01 dan 2012-06-01, yang Anda gunakan untuk membuat dan mengelola penyeimbang beban Anda.

Titik akhir VPC antarmuka didukung oleh AWS PrivateLink, fitur yang memungkinkan komunikasi antara aplikasi Anda dan Layanan AWS menggunakan alamat IP pribadi. Untuk informasi selengkapnya, lihat [AWS PrivateLink](#).

Kuota

AWS PrivateLink tidak mendukung Penyeimbang Beban Jaringan dengan lebih dari 50 pendengar.

Buat titik akhir antarmuka untuk Keseimbangan Beban Elastis

Buat sebuah titik akhir untuk Keseimbangan Beban Elastis dengan menggunakan nama layanan berikut:

```
com.amazonaws.region.elasticloadbalancing
```

Untuk informasi selengkapnya, lihat [Membuat sebuah titik akhir antarmuka di AWS PrivateLink](#) Panduan.

Membuat kebijakan titik akhir VPC untuk Elastic Load Balancing

Anda dapat menyematkan kebijakan ke titik akhir VPC Anda untuk mengontrol akses ke Elastic Load Balancing API. Kebijakan menentukan:

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.
- Sumber daya untuk melakukan tindakan.

Contoh berikut menunjukkan kebijakan titik akhir VPC yang menolak izin setiap orang untuk membuat kebijakan penskalaan melalui titik akhir. Kebijakan contoh tersebut juga memberikan izin kepada semua orang untuk melakukan semua tindakan lainnya.

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

Untuk informasi selengkapnya, lihat [Mengontrol akses ke layanan menggunakan kebijakan endpoint](#) di AWS PrivateLink Panduan.

Migrasi Classic Load Balancer Anda

Elastic Load Balancing mendukung jenis load balancer berikut: Application Load Balancers, Network Load Balancers, Gateway Load Balancers, dan Classic Load Balancers. Untuk informasi tentang fitur yang berbeda dari setiap jenis penyeimbang beban, lihat [Perbandingan produk Elastic Load Balancing](#).

Anda juga dapat memilih untuk memigrasikan Classic Load Balancer yang ada di VPC, ke Application Load Balancer atau Network Load Balancer.

Manfaat migrasi dari Classic Load Balancer

Setiap jenis load balancer memiliki fitur, fungsi, dan konfigurasi uniknya sendiri. Tinjau manfaat masing-masing penyeimbang beban untuk membantu memutuskan mana yang terbaik untuk Anda.

Application Load Balancer

Menggunakan Application Load Balancer bukan Classic Load Balancer memiliki manfaat sebagai berikut:

Support untuk:

- [Kondisi jalur](#), [kondisi Host](#), dan [kondisi header HTTP](#).
- Mengarahkan permintaan dari satu URL ke URL lainnya, dan merutekan permintaan ke beberapa aplikasi pada satu instans EC2.
- Mengembalikan respons HTTP kustom.
- Mendaftarkan target berdasarkan alamat IP, dan mendaftar fungsi Lambda sebagai target. Termasuk target di luar VPC untuk load balancer.
- Mengotentikasi pengguna melalui identitas perusahaan atau sosial.
- Aplikasi kontainer Amazon Elastic Container Service (Amazon ECS).
- Memantau kesehatan setiap layanan secara independen.

Log akses berisi informasi tambahan dan disimpan dalam format terkompresi.

Peningkatan kinerja penyeimbang beban secara keseluruhan.

Network Load Balancer

Menggunakan Network Load Balancer sebagai pengganti Classic Load Balancer memiliki manfaat sebagai berikut:

Support untuk:

- Alamat IP statis, yang memungkinkan penetapan satu alamat IP Elastis per subnet yang diaktifkan untuk penyeimbang beban.
- Mendaftarkan target berdasarkan alamat IP, termasuk target di luar VPC untuk penyeimbang beban.
- Perutean permintaan ke beberapa aplikasi pada satu instans EC2.
- Aplikasi kontainer Amazon Elastic Container Service (Amazon ECS).
- Memantau kesehatan setiap layanan secara independen.

Kemampuan untuk menangani beban kerja yang mudah menguap dan skala untuk jutaan permintaan per detik.

Migrasi menggunakan wizard migrasi

Wisaya migrasi menggunakan konfigurasi Classic Load Balancer Anda untuk membuat Application Load Balancer atau Network Load Balancer yang setara. Ini mengurangi waktu dan upaya yang diperlukan untuk memigrasikan Classic Load Balancer dibandingkan dengan metode lain.

Note

Wizard menciptakan penyeimbang beban baru. Wizard tidak mengonversi Classic Load Balancer yang ada menjadi Application Load Balancer atau Network Load Balancer. Anda harus mengarahkan lalu lintas secara manual ke penyeimbang beban yang baru dibuat.

Batasan

- Nama penyeimbang beban baru tidak bisa sama dengan penyeimbang beban yang ada dengan tipe yang sama, di wilayah yang sama.
- Jika Classic Load Balancer memiliki tag yang berisi `aws :` awalan di kuncinya, tag tersebut tidak akan dimigrasikan.

Saat bermigrasi ke Application Load Balancer

- Jika Classic Load Balancer hanya memiliki satu subnet, Anda harus menentukan subnet kedua.
- Jika Classic Load Balancer memiliki pendengar HTTP/HTTPS yang menggunakan pemeriksaan kesehatan TCP, protokol pemeriksaan kesehatan diperbarui ke HTTP dan jalur disetel ke “/”.
- Jika Classic Load Balancer memiliki pendengar HTTPS menggunakan kebijakan keamanan khusus atau tidak didukung, wizard migrasi menggunakan kebijakan keamanan default untuk jenis penyeimbang beban baru.

Saat bermigrasi ke Network Load Balancer

- Jenis contoh berikut tidak akan terdaftar dengan kelompok target baru: C1, CC1, CC2, CG1, CG2, CR1, CS1, G1, G2, HI1, HS1, M1, M1, M2, M3, T1
- Pengaturan pemeriksaan kesehatan tertentu dari Classic Load Balancer Anda mungkin tidak dapat ditransfer ke grup target baru. Kasus-kasus ini akan ditunjukkan sebagai perubahan di bagian ringkasan wizard migrasi.
- Jika Classic Load Balancer memiliki pendengar SSL, wizard migrasi akan membuat pendengar TLS menggunakan sertifikat dan kebijakan keamanan dari pendengar SSL.

Proses wizard migrasi

Untuk memigrasikan Classic Load Balancer menggunakan wizard migrasi

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 Classic Load Balancer yang ingin Anda migrasikan.
4. Di bagian Detail penyeimbang beban, pilih Luncurkan panduan migrasi.
5. Pilih Migrasi ke Application Load Balancer, atau Migrasi ke Network Load Balancer, untuk membuka panduan migrasi.
6. Di bawah Nama penyeimbang beban baru, untuk nama Load balancer masukkan nama untuk penyeimbang beban baru Anda.
7. Di bawah Nama grup target baru dan tinjau target, untuk nama grup Target masukkan nama untuk grup target baru Anda.
8. (Opsional) Di bawah Target, Anda dapat meninjau instance target yang akan didaftarkan pada grup target baru.

9. (Opsional) Di bawah tag Ulasan, Anda dapat meninjau tag yang akan diterapkan ke penyeimbang beban baru Anda
10. Di bawah Ringkasan untuk Application Load Balancer, atau Ringkasan untuk Network Load Balancer, tinjau dan verifikasi opsi konfigurasi yang ditetapkan oleh wizard migrasi.
11. Setelah Anda puas dengan ringkasan konfigurasi, pilih Create Application Load Balancer, atau Buat Network Load Balancer, untuk memulai migrasi.

Migrasi menggunakan utilitas salinan penyeimbang beban

Utilitas salinan penyeimbang beban tersedia dalam repositori Elastic Load Balancing Tools, di halaman. [AWS GitHub](#)

Sumber daya

- [Alat Elastic Load Balancing](#)
- [Classic Load Balancer ke utilitas salinan Application Load Balancer](#)
- [Classic Load Balancer ke utilitas salinan Network Load Balancer](#)

Migrasikan penyeimbang beban Anda secara manual

Informasi berikut memberikan petunjuk umum untuk secara manual membuat Application Load Balancer atau Network Load Balancer berdasarkan Classic Load Balancer yang ada di VPC. Anda dapat bermigrasi menggunakan AWS Management Console, AWS CLI, atau AWS SDK. Untuk informasi selengkapnya, lihat [Memulai Elastic Load Balancing](#).

Setelah menyelesaikan proses migrasi, Anda dapat memanfaatkan fitur penyeimbang beban baru Anda.

Proses migrasi manual

Langkah 1: Membuat penyeimbang beban baru

Buat penyeimbang beban dengan konfigurasi yang setara dengan Classic Load Balancer untuk bermigrasi.

1. Buat penyeimbang beban baru, dengan skema yang sama (menghadap-internet atau internal), subnet, dan grup keamanan sebagai Classic Load Balancer.

2. Buat satu grup target untuk penyeimbang beban Anda, dengan pengaturan pemeriksaan kesehatan yang sama dengan yang Anda miliki untuk Classic Load Balancer Anda.
3. Lakukan salah satu dari berikut ini:
 - Jika Classic Load Balancer dilampirkan ke grup Auto Scaling, lampirkan grup target Anda ke grup Auto Scaling. Ini juga mendaftarkan instans Auto Scaling dengan kelompok target.
 - Daftarkan instans EC2 Anda dengan kelompok target Anda.
4. Buat satu atau lebih pendengar, masing-masing dengan aturan default yang meneruskan permintaan ke grup target. Jika Anda membuat pendengar HTTPS, Anda dapat menentukan sertifikat yang sama yang Anda tentukan untuk Classic Load Balancer Anda. Kami sarankan Anda menggunakan kebijakan keamanan default.
5. Jika Classic Load Balancer Anda memiliki tag, tinjau dan tambahkan tag yang relevan ke penyeimbang beban baru Anda.

Langkah 2: Secara bertahap mengalihkan lalu lintas ke penyeimbang beban baru Anda

Setelah instans Anda terdaftar dengan penyeimbang beban baru Anda, Anda dapat memulai proses mengalihkan lalu lintas dari penyeimbang beban lama ke penyeimbang beban baru. Hal ini memungkinkan Anda untuk menguji penyeimbang beban baru Anda sambil meminimalkan risiko pada ketersediaan aplikasi Anda.

Untuk mengalihkan lalu lintas secara bertahap ke penyeimbang beban baru

1. Tempelkan nama DNS penyeimbang beban baru Anda ke bidang alamat browser web yang terhubung ke internet. Jika semuanya bekerja, browser menampilkan halaman default aplikasi Anda.
2. Buat rekaman DNS baru yang menghubungkan nama domain Anda dengan penyeimbang beban baru Anda. Jika layanan DNS Anda mendukung pembobotan, tentukan berat 1 dalam catatan DNS baru dan berat 9 dalam catatan DNS yang ada untuk penyeimbang beban lama Anda. Ini mengarahkan 10% lalu lintas ke penyeimbang beban baru dan 90% lalu lintas ke penyeimbang beban lama.
3. Pantau penyeimbang beban baru Anda untuk memverifikasi bahwa penyeimbang beban baru menerima permintaan lalu lintas dan merutekan ke instans Anda.

⚠ Important

time-to-live (TTL) dalam catatan DNS adalah 60 detik. Artinya setiap server DNS yang menyelesaikan nama domain Anda menyimpan informasi catatan dalam cache selama 60 detik, sementara perubahannya menyebar. Oleh karena itu, server DNS ini masih dapat merutekan lalu lintas ke penyeimbang beban lama Anda hingga 60 detik setelah Anda menyelesaikan langkah sebelumnya. Selama penyebaran, lalu lintas dapat diarahkan ke salah satu penyeimbang beban.

4. Lanjutkan untuk memperbarui berat rekaman DNS Anda sampai semua lalu lintas diarahkan ke penyeimbang beban baru Anda. Setelah selesai, Anda dapat menghapus rekaman DNS untuk penyeimbang beban lama Anda.

Langkah 3: Memperbarui kebijakan, skrip, dan kode

Jika Anda memindahkan Classic Load Balancer ke Application Load Balancer atau Network Load Balancer, pastikan untuk melakukan hal berikut:

- Memperbarui kebijakan IAM yang menggunakan API versi 2012-06-01 untuk menggunakan versi 2015-12-01.
- Perbarui proses yang menggunakan CloudWatch metrik di AWS/ELB namespace untuk menggunakan metrik dari atau namespace. `AWS/ApplicationELB` `AWS/NetworkELB`
- Perbarui skrip yang menggunakan `aws elb` AWS CLI perintah untuk menggunakan `aws elbv2` AWS CLI perintah.
- Perbarui AWS CloudFormation template yang menggunakan `AWS::ElasticLoadBalancing::LoadBalancer` sumber daya untuk menggunakan `AWS::ElasticLoadBalancingV2` sumber daya.
- Perbarui kode yang menggunakan API Penyeimbangan Beban Elastis versi 2012-06-01 untuk menggunakan versi 2015-12-01.

Sumber daya

- [elbv2](#) dalam AWS CLI Referensi Perintah
- [Referensi API Elastic Load Balancing Versi 2015-12-01](#)
- [Manajemen identitas dan akses untuk Penyeimbangan Beban Elastis](#)

- [Metrik Application Load Balancer](#) dalam Panduan Pengguna untuk Application Load Balancer
- [Metrik Network Load Balancer](#) dalam Panduan pengguna untuk Network Load Balancer
- [AWS::ElasticLoadBalancingV2::LoadBalancer](#) di Panduan Pengguna AWS CloudFormation

Langkah 4: Hapus penyeimbang beban lama

Anda dapat menghapus Classic Load Balancer lama setelah:

- Anda telah mengalihkan semua lalu lintas dari penyeimbang beban lama ke penyeimbang beban baru.
- Semua permintaan yang ada yang diarahkan ke penyeimbang beban lama telah selesai.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.