



Panduan Pengguna

Amazon EC2 Auto Scaling



Amazon EC2 Auto Scaling: 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 Amazon EC2 Auto Scaling?	1
Fitur Auto Scaling Amazon EC2	1
Harga untuk Amazon EC2 Auto Scaling	3
Memulai	4
Bekerja dengan grup Auto Scaling	4
Manfaat Auto Scaling	5
Contoh: Menutupi permintaan variabel	5
Contoh: Arsitektur aplikasi web	7
Contoh: Mendistribusikan instance di seluruh Availability Zone	9
Siklus aktif nyata	12
Meningkatkan skala	13
Instance dalam layanan	13
Penurunan skala	14
Lepaskan satu instance	15
Melampirkan instance	15
Hook siklus aktif	15
Masukkan dan keluar dari siaga	16
Kuota	16
Minta pembatasan untuk API Auto Scaling Amazon EC2	18
Tingkat penghentian EC2	18
Layanan lainnya	18
Mengatur	20
Bersiap menggunakan Amazon EC2	20
Bersiap menggunakan alatAWS CLI	20
Memulai	21
Tutorial: Buat grup Auto Scaling pertama Anda	22
Bersiaplah untuk penelusuran	22
Langkah 1: Buat templat peluncuran	23
Langkah 2: Buat grup Auto Scaling satu contoh	24
Langkah 3: Verifikasi grup Auto Scaling Anda	25
Langkah 4: Menghentikan instans di grup Auto Scaling	26
Langkah 5: Langkah selanjutnya	27
Langkah 6: Bersihkan	27
Tutorial: Siapkan aplikasi yang diskalakan dan diseimbangkan beban	28

Prasyarat	30
Langkah 1: Siapkan templat peluncuran atau konfigurasi peluncuran	31
Langkah 2: Buat grup Auto Scaling	34
Langkah 3: Verifikasi bahwa penyeimbang beban Anda terpasang	35
Langkah 4: Langkah selanjutnya	36
Langkah 5: Bersihkan	37
Sumber daya terkait	38
Templat peluncuran	39
Izin untuk bekerja dengan template peluncuran	40
Operasi API yang didukung oleh template peluncuran	40
Membuat template peluncuran untuk grup Auto Scaling	40
Buat template peluncuran Anda (konsol)	41
Ubah pengaturan antarmuka jaringan default (konsol)	44
Ubah konfigurasi penyimpanan (konsol)	46
Buat template peluncuran dari instance yang ada (konsol)	49
Sumber daya terkait	49
Batasan	50
Buat template peluncuran menggunakan pengaturan lanjutan	50
Pengaturan yang diperlukan	50
Pengaturan lanjutan	51
Minta Instans Spot	56
Blok Kapasitas untuk ML	58
Migrasi grup Auto Scaling Anda untuk meluncurkan template	63
Langkah 1: Temukan grup Auto Scaling yang menggunakan konfigurasi peluncuran	63
Langkah 2: Salin konfigurasi peluncuran ke template peluncuran	66
Langkah 3: Perbarui grup Auto Scaling untuk menggunakan template peluncuran	67
Langkah 4: Ganti instans Anda	68
Informasi tambahan	68
Migrasikan CloudFormation tumpukan untuk meluncurkan templat	68
Temukan grup Auto Scaling yang menggunakan konfigurasi peluncuran	69
Perbarui tumpukan untuk menggunakan template peluncuran	70
Memahami perilaku pembaruan sumber daya tumpukan	74
Lacak migrasi	74
Luncurkan referensi pemetaan konfigurasi	75
AWS CLI contoh untuk bekerja dengan template peluncuran	77
Contoh penggunaan	77

Buat template peluncuran dasar	78
Tentukan tag yang menandai instance saat diluncurkan	79
Tentukan peran IAM untuk diteruskan ke instance	79
Tetapkan alamat IP publik	80
Tentukan skrip data pengguna yang mengonfigurasi instance saat peluncuran	80
Tentukan pemetaan perangkat blok	80
Tentukan Host Khusus untuk membawa lisensi perangkat lunak dari vendor eksternal	81
Tentukan antarmuka jaringan yang ada	81
Buat beberapa antarmuka jaringan	81
Kelola templat peluncuran Anda	82
Memperbarui grup Auto Scaling untuk menggunakan template peluncuran	85
Gunakan parameter Systems Manager alih-alih ID AMI	86
Buat template peluncuran yang menentukan parameter untuk AMI	86
Verifikasi template peluncuran mendapatkan ID AMI yang benar	91
Sumber daya terkait	92
Batasan	92
Konfigurasi peluncuran	94
Buat konfigurasi peluncuran	95
Buat konfigurasi peluncuran	95
Konfigurasi IMDS	98
Buat konfigurasi peluncuran menggunakan instans EC2	100
Ubah konfigurasi peluncuran	105
Grup Auto Scaling	107
Buat grup Auto Scaling menggunakan template peluncuran	108
Buat grup menggunakan template peluncuran	109
Buat grup menggunakan wizard peluncuran EC2	112
Gunakan beberapa jenis instans dan opsi pembelian	116
Buat grup Auto Scaling menggunakan konfigurasi peluncuran	162
Buat grup menggunakan konfigurasi peluncuran	163
Buat grup menggunakan instans EC2	166
Memperbarui grup Auto Scaling	172
Perbarui instans Auto Scaling	173
Beri tag grup dan instans	174
Membatasi penamaan dan penggunaan tag	175
Siklus hidup penandaan instans EC2	176
Tandai grup Auto Scaling Anda	176

Hapus tag	179
Tag untuk keamanan	180
Kontrol akses ke tag	181
Gunakan tag untuk memfilter grup Auto Scaling	182
Kebijakan pemeliharaan instans	186
Gambaran Umum	187
Menetapkan kebijakan pemeliharaan instans pada grup Anda	195
Hook siklus aktif	199
Ketersediaan kait siklus hidup	201
Pertimbangan dan batasan	201
Sumber daya terkait	203
Cara kerja hook siklus aktif	204
Bersiaplah untuk menambahkan kait siklus hidup	205
Mengambil status siklus hidup target	213
Tambahkan hook siklus aktif	215
Selesaikan tindakan siklus hidup	219
Tutorial: Konfigurasi data pengguna untuk mengambil status siklus hidup target melalui metadata instance	221
Tutorial: Konfigurasi kait siklus hidup yang memanggil fungsi Lambda	230
Kolam hangat	239
Konsep inti	239
Prasyarat	242
Perbarui contoh di kolam yang hangat	243
Sumber daya terkait	244
Batasan	244
Gunakan kait siklus hidup	245
Buat kolam hangat untuk grup Auto Scaling	249
Lihat status pemeriksaan kesehatan	251
AWS CLI contoh untuk bekerja dengan kolam hangat	254
Instans lepas-lampirkan	257
Pertimbangan untuk melepaskan contoh	257
Pertimbangan untuk melampirkan instance	258
Pindahkan instance ke grup yang berbeda menggunakan detach and attach	259
Hapus instans untuk sementara	264
Cara kerja status siaga	265
Pertimbangan	265

Status kesehatan suatu instance dalam status siaga	266
Hapus sementara instance dengan menyetelnya ke standby	265
Hapus infrastruktur Auto Scaling Anda	271
Hapus grup Auto Scaling Anda	271
(Opsional) Hapus konfigurasi peluncuran	272
(Opsional) Hapus templat peluncuran	273
(Opsional) Hapus load balancer dan grup target	274
(Opsional) Hapus CloudWatch alarm	275
AWS Contoh SDK untuk bekerja dengan grup Auto Scaling	275
Membuat grup Auto Scaling	276
Memperbarui grup Auto Scaling	291
Jelaskan grup Auto Scaling	301
Menghapus grup Auto Scaling	315
Daur ulang instans Anda	329
Refresh instans	329
Cara kerja penyegaran instance	330
Memahami nilai default	336
Mulai penyegaran instance	339
Pantau penyegaran instance	351
Batalkan penyegaran instance	354
Batalkan perubahan dengan rollback	355
Gunakan pencocokan lewati	360
Tambahkan pos pemeriksaan	369
Masa aktif instans maksimum	375
Pertimbangan	376
Mengatur masa pakai instans maksimum	376
Batasan	378
Skala grup Anda	379
Pilih metode penskalaan Anda	379
Tetapkan batas penskalaan	381
Mengatur contoh pemanasan default	383
Pertimbangan kinerja penskalaan	384
Pilih waktu pemanasan instance default	385
Aktifkan pemanasan instance default untuk grup	386
Verifikasi pemanasan instance default untuk grup	387

Temukan kebijakan penskalaan dengan waktu pemanasan instance yang ditetapkan sebelumnya	388
Hapus pemanasan instance yang disetel sebelumnya untuk kebijakan penskalaan	389
Penskalaan manual	390
Ubah kapasitas yang diinginkan dari grup Auto Scaling Anda	391
Mengakhiri instance di grup Auto Scaling (AWS CLI)	394
Penskalaan terjadwal	395
Cara kerja penskalaan terjadwal	396
Jadwal berulang	397
Zona waktu	398
Pertimbangan	398
Buat tindakan terjadwal	399
Lihat detail tindakan terjadwal	401
Verifikasi aktivitas penskalaan	402
Hapus tindakan terjadwal	402
Batasan	403
Penskalaan dinamis	403
Cara kerja kebijakan penskalaan dinamis	404
Beberapa kebijakan penskalaan dinamis	405
Kebijakan penskalaan pelacakan target	407
Kebijakan penskalaan langkah dan sederhana	421
Menskalakan jeda pakai	438
Penskalaan berdasarkan Amazon SQS	441
Verifikasi aktivitas penskalaan	449
Nonaktifkan kebijakan penskalaan	451
Menghapus kebijakan penskalaan	454
AWS CLI contoh untuk kebijakan penskalaan	456
Penskalaan prediktif	459
Cara kerja penskalaan prediktif	460
Buat kebijakan penskalaan prediktif	464
Evaluasi kebijakan penskalaan prediktif Anda	472
Ganti perkiraan	481
Gunakan metrik khusus	486
Pengakhiran instans kontrol	498
Skenario kebijakan penghentian	498
Konfigurasi kebijakan penghentian	502

Membuat kebijakan penghentian kustom dengan Lambda	508
Gunakan perlindungan skala dalam instance	515
Desain untuk penghentian instance yang anggun	519
Proses resume ditangguhkan	523
Jenis proses	524
Pertimbangan	525
Menangguhkan proses	525
Lanjutkan proses	526
Bagaimana proses yang ditangguhkan memengaruhi proses lain	527
Memantau	531
Pemeriksaan kondisi	533
Tentang pemeriksaan kesehatan	534
Lihat alasan kegagalan pemeriksaan kesehatan	542
Atur masa tenggang pemeriksaan kesehatan	543
Monitor dengan AWS Health Dashboard	545
Monitor CloudWatch metrik	547
Lihat grafik pemantauan di konsol Auto Scaling Amazon EC2	547
CloudWatch metrik untuk Auto Scaling Amazon EC2	552
Konfigurasi pemantauan untuk instans Auto Scaling	559
Log panggilan API dengan AWS CloudTrail	562
Informasi Auto Scaling Amazon EC2 di CloudTrail	562
Memahami entri file log Amazon EC2 Auto Scaling	563
Sumber daya terkait	565
Opsi notifikasi Amazon SNS	565
Amazon SNS dan Amazon EC2 Auto Scaling	566
Bekerja dengan layanan lain	573
Penyeimbangan Ulang Kapasitas	573
Gambaran Umum	574
Perilaku penyeimbangan kembali kapasitas	575
Pertimbangan	576
Aktifkan Rebalancing Kapasitas (konsol)	578
Aktifkan Rebalancing Kapasitas (AWS CLI)	579
Sumber daya terkait	584
Batasan	584
Reservasi Kapasitas	584
Langkah 1: Buat Reservasi Kapasitas	585

Langkah 2: Buat grup Reservasi Kapasitas	588
Langkah 3: Buat template peluncuran	590
Langkah 4: Buat grup Auto Scaling	591
Sumber daya terkait	593
AWS CloudShell	594
AWS CloudFormation	594
Auto Scaling dan template Amazon EC2 AWS CloudFormation	595
Pelajari lebih lanjut tentang AWS CloudFormation	595
Compute Optimizer	596
Batasan	596
Temuan	597
Melihat rekomendasi	597
Pertimbangan untuk mengevaluasi rekomendasi	598
Penyeimbang Beban Elastis	599
Jenis Elastic Load Balancing	600
Bersiaplah untuk memasang penyeimbang beban	602
Pasang penyeimbang beban	604
Konfigurasi penyeimbang beban dari konsol Auto Scaling Amazon EC2	608
Verifikasi status lampiran	609
Tambahkan Availability Zone	610
AWS CLI contoh untuk bekerja dengan Elastic Load Balancing	614
Kisi VPC	622
Bersiaplah untuk melampirkan grup sasaran	624
Lampirkan grup target VPC Lattice	627
Verifikasi status lampiran	632
EventBridge	633
Referensi acara Auto Scaling Amazon EC2	634
Acara dan pola contoh kolam hangat	643
Buat EventBridge aturan	649
Amazon VPC	654
VPC default	654
VPC Non-default	655
Pertimbangan saat memilih subnet VPC	655
Pembuatan alamat IP di VPC	656
Antarmuka jaringan dalam VPC	656
Tenansi penempatan instans	657

AWS Outposts	657
Lebih banyak sumber daya untuk mempelajari VPC	657
Keamanan	659
Keamanan infrastruktur	660
Sumber daya terkait	660
Ketangguhan	660
Sumber daya terkait	662
Perlindungan data	662
Gunakan AWS KMS keys untuk mengenkripsi volume Amazon EBS	663
Sumber daya terkait	664
AWS KMS kebijakan kunci untuk digunakan dengan volume terenkripsi	664
Manajemen Identitas dan Akses	671
Kontrol akses	671
Cara kerja Amazon EC2 Auto Scaling dengan IAM	672
Izin API:	682
Kebijakan terkelola	683
Peran terkait layanan	688
Contoh kebijakan berbasis identitas	696
Pencegahan confused deputy lintas layanan	705
Luncurkan dukungan templat	707
Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2	715
Validasi kepatuhan	718
Kepatuhan PCI DSS	719
Gunakan titik akhir VPC untuk konektivitas pribadi	720
Buat endpoint VPC antarmuka	720
Buat kebijakan endpoint VPC	721
Pemecahan Masalah	722
Mengambil pesan kesalahan	722
Matikan aktivitas penskalaan	724
Sumber daya pemecahan masalah tambahan	725
Kegagalan peluncuran Instance	726
Konfigurasi yang diminta saat ini tidak didukung.	727
Grup keamanan <name of the security group> tidak ada. Peluncuran instance EC2 gagal. ..	728
Pasangan kunci <key pair associated with your EC2 instance> tidak ada. Peluncuran instance EC2 gagal.	728

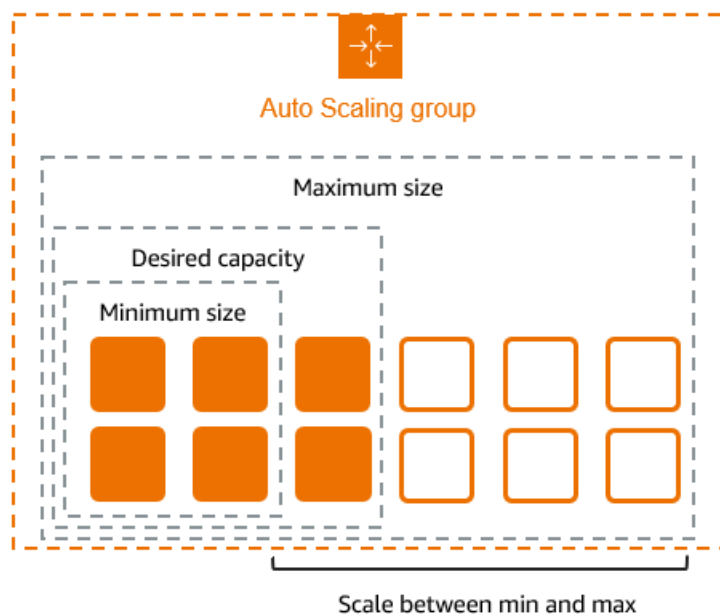
Jenis instance yang Anda minta (<instance type>) tidak didukung dalam Zona Ketersediaan yang Anda minta (<instance Availability Zone>)... ..	728
Harga permintaan Spot Anda sebesar 0,015 lebih rendah dari harga pemenuhan permintaan Spot minimum 0,0735... ..	729
Nama perangkat tidak <device name>valid/Unggahan nama perangkat tidak valid.	
Peluncuran instance EC2 gagal.	729
Value (<name associated with the instance storage device>) untuk parameter VirtualName tidak valid... Peluncuran instance EC2 gagal.	730
EBS memblokir pemetaan perangkat tidak didukung untuk AMI instance-store.	730
Grup penempatan tidak boleh digunakan dengan contoh tipe '<instance type>'. Peluncuran instance EC2 gagal.	731
Klien. InternalError: Kesalahan klien saat diluncurkan.	731
Saat ini kami tidak memiliki <instance type>kapasitas yang cukup di Availability Zone yang Anda minta... Peluncuran instance EC2 gagal.	732
Reservasi yang diminta tidak memiliki kapasitas yang kompatibel dan tersedia untuk permintaan ini. Peluncuran instance EC2 gagal.	733
Reservasi Blok Kapasitas Anda <reservation id>belum aktif. Peluncuran instance EC2 gagal.	734
Tidak ada kapasitas Spot yang tersedia yang sesuai dengan permintaan Anda. Peluncuran instance EC2 gagal.	734
<number of instances> instance telah berjalan. Peluncuran instance EC2 gagal.	734
Masalah AMI	735
ID AMI <ID of your AMI> tidak ada. Peluncuran instance EC2 gagal.	736
AMI <AMI ID> tertunda, dan tidak dapat dijalankan. Peluncuran instance EC2 gagal.	736
<device name>Nama perangkat tidak valid. Peluncuran instance EC2 gagal.	736
Arsitektur 'arm64 'dari jenis instance yang ditentukan tidak cocok dengan arsitektur' x86_64 'dari AMI yang ditentukan... Peluncuran instans EC2 gagal.	737
AMI '<AMI ID>' dinonaktifkan, dan tidak dapat dijalankan. Peluncuran instance EC2 gagal. .	738
Masalah load balancer	739
Satu atau lebih kelompok sasaran tidak ditemukan. Validasi konfigurasi load balancer gagal.	739
<your load balancer>Tidak dapat menemukan Load Balancer. Validasi konfigurasi load balancer gagal.	740
Tidak ada Load Balancer AKTIF yang bernama <load balancer name>. Pembaruan konfigurasi load balancer gagal.	740

Instance EC2 <instance ID> tidak ada dalam VPC. Pembaruan konfigurasi load balancer gagal.	741
Luncurkan masalah template	741
Anda harus menggunakan templat peluncuran yang valid sepenuhnya (nilai tidak valid)	741
Anda tidak diizinkan untuk menggunakan template peluncuran (izin tidak memadai)	742
Pemeriksaan kondisi	743
Sebuah instance dikeluarkan dari layanan sebagai tanggapan atas kegagalan pemeriksaan status instans EC2	744
Sebuah instance dikeluarkan dari layanan sebagai tanggapan terhadap reboot terjadwal EC2	745
Sebuah contoh dikeluarkan dari layanan sebagai tanggapan terhadap pemeriksaan kesehatan EC2 yang mengindikasikan telah dihentikan atau dihentikan	746
Sebuah contoh diambil dari layanan sebagai tanggapan atas kegagalan pemeriksaan kesehatan sistem ELB	747
Informasi terkait	749
Riwayat dokumen	752
.....	dcxciii

Apa itu Amazon EC2 Auto Scaling?

Amazon EC2 Auto Scaling membantu Anda memastikan bahwa Anda memiliki jumlah instance Amazon EC2 yang tepat tersedia untuk menangani beban untuk aplikasi Anda. Anda membuat koleksi instance EC2, yang disebut Grup Auto Scaling. Anda dapat menentukan jumlah minimum instance di setiap grup Auto Scaling, dan Amazon EC2 Auto Scaling memastikan bahwa grup Anda tidak pernah berada di bawah ukuran ini. Anda dapat menentukan jumlah maksimum instance di setiap grup Auto Scaling, dan Amazon EC2 Auto Scaling memastikan grup Anda tidak pernah berada di atas ukuran ini. Jika Anda menentukan kapasitas yang diinginkan, baik ketika Anda membuat grup atau kapan saja setelahnya, Amazon EC2 Auto Scaling memastikan grup Anda memiliki banyak instance ini. Jika Anda menentukan kebijakan penskalaan, maka Amazon EC2 Auto Scaling dapat meluncurkan atau menghentikan instance saat permintaan pada aplikasi Anda meningkat atau menurun.

Misalnya, grup Auto Scaling berikut memiliki ukuran minimum empat instans, kapasitas enam instans yang diinginkan, dan ukuran maksimum dua belas instans. Kebijakan penskalaan yang Anda tentukan menyesuaikan jumlah instance, dalam jumlah instance minimum dan maksimum, berdasarkan kriteria yang Anda tentukan.



Fitur Auto Scaling Amazon EC2

Dengan Auto Scaling Amazon EC2, instans EC2 Anda diatur ke dalam grup Auto Scaling sehingga dapat diperlakukan sebagai unit logis untuk keperluan penskalaan dan pengelolaan. Grup Auto

Scaling menggunakan templat peluncuran (atau konfigurasi peluncuran) sebagai templat konfigurasi untuk instans EC2 mereka.

Berikut ini adalah fitur utama dari Amazon EC2 Auto Scaling:

Memantau kesehatan instance yang sedang berjalan

Auto Scaling Amazon EC2 secara otomatis memantau kesehatan dan ketersediaan instans Anda menggunakan pemeriksaan kesehatan EC2 dan menggantikan instans yang dihentikan atau terganggu untuk mempertahankan kapasitas yang Anda inginkan.

Pemeriksaan kesehatan khusus

Selain pemeriksaan kesehatan bawaan, Anda dapat menentukan pemeriksaan kesehatan khusus yang khusus untuk aplikasi Anda untuk memverifikasi bahwa itu merespons seperti yang diharapkan. Jika sebuah instans gagal pemeriksaan kesehatan kustom Anda, itu secara otomatis diganti untuk mempertahankan kapasitas yang Anda inginkan.

Menyeimbangkan kapasitas di seluruh Availability Zone

Anda dapat menentukan beberapa Availability Zone untuk grup Auto Scaling, dan Auto Scaling Amazon EC2 menyeimbangkan instans Anda secara merata di seluruh Availability Zone saat grup menskalakan. Ini memberikan ketersediaan dan ketahanan yang tinggi dengan melindungi aplikasi Anda dari kegagalan di satu lokasi.

Berbagai tipe instans dan opsi pembelian

Dalam satu grup Auto Scaling, Anda dapat meluncurkan beberapa jenis instans dan opsi pembelian (Instans Spot dan Sesuai Permintaan), sehingga Anda dapat mengoptimalkan biaya melalui penggunaan Instans Spot. Anda juga dapat memanfaatkan diskon Instans Cadangan dan Savings Plan dengan menggunakannya bersama dengan Instans Sesuai Permintaan di grup.

Penggantian otomatis Instans Spot

Jika grup Anda menyertakan Instans Spot, Auto Scaling Amazon EC2 dapat secara otomatis meminta penggantian kapasitas Spot jika Instans Spot Anda terganggu. Melalui Penyeimbangan Kembali Kapasitas, Auto Scaling Amazon EC2 juga dapat memantau dan secara proaktif mengganti Instans Spot Anda yang berisiko tinggi mengalami gangguan.

Penyeimbangan beban

Anda dapat menggunakan penyeimbangan beban Elastic Load Balancing dan pemeriksaan kesehatan untuk memastikan distribusi lalu lintas aplikasi yang merata ke instans sehat Anda.

Kapan pun instans diluncurkan atau dihentikan, Auto Scaling Amazon EC2 secara otomatis mendaftarkan dan membatalkan pendaftaran instans dari penyeimbang beban.

Skalabilitas

Auto Scaling Amazon EC2 juga menyediakan beberapa cara bagi Anda untuk menskalakan grup Auto Scaling Anda. Menggunakan penskalaan otomatis memungkinkan Anda mempertahankan ketersediaan aplikasi dan mengurangi biaya dengan menambahkan kapasitas untuk menangani beban puncak dan menghilangkan kapasitas saat permintaan lebih rendah. Anda juga dapat menyesuaikan ukuran grup Auto Scaling secara manual sesuai kebutuhan.

Refresh instans

Fitur penyegaran instans menyediakan mekanisme untuk memperbarui instance secara bergilir saat Anda memperbarui AMI atau meluncurkan template. Anda juga dapat menggunakan pendekatan bertahap, yang dikenal sebagai penerapan kenari, untuk menguji AMI baru atau meluncurkan template pada serangkaian kecil instance sebelum meluncurkannya ke seluruh grup.

Hook siklus aktif

Kait siklus hidup berguna untuk mendefinisikan tindakan kustom yang dipanggil saat instance baru diluncurkan atau sebelum instance dihentikan. Fitur ini sangat berguna untuk membangun arsitektur berbasis peristiwa, tetapi juga membantu Anda mengelola instance melalui siklus hidupnya.

Support untuk beban kerja stateful

Kait siklus hidup juga menawarkan mekanisme untuk mempertahankan status saat dimatikan. Untuk memastikan kontinuitas aplikasi stateful, Anda juga dapat menggunakan perlindungan skala dalam atau kebijakan penghentian khusus untuk mencegah instance dengan proses yang berjalan lama berhenti lebih awal.

Untuk informasi lebih lanjut tentang batas Amazon EC2 Auto Scaling, lihat [Manfaat Amazon EC2 Auto Scaling](#).

Harga untuk Amazon EC2 Auto Scaling

Tidak ada biaya tambahan dengan Amazon EC2 Auto Scaling, jadi mudah untuk mencobanya dan melihat bagaimana hal itu dapat bermanfaat bagi arsitektur Anda. AWS Anda hanya membayar AWS sumber daya (misalnya, instans EC2, volume EBS, dan CloudWatch alarm) yang Anda gunakan.

Memulai

Untuk memulai, selesaikan tutorial [Create your first Auto Scaling group](#) untuk membuat grup Auto Scaling dan lihat bagaimana responsnya ketika sebuah instance dalam grup tersebut berakhir.

Bekerja dengan grup Auto Scaling

Anda dapat membuat, mengakses, dan mengelola grup Auto Scaling menggunakan salah satu antarmuka berikut:

- **AWS Management Console**— Menyediakan antarmuka web yang dapat Anda gunakan untuk mengakses grup Auto Scaling Anda. Jika Anda telah mendaftarkan Akun AWS, Anda dapat mengakses grup Auto Scaling dengan masuk ke AWS Management Console, menggunakan kotak pencarian di bilah navigasi untuk mencari grup Auto Scaling, lalu memilih grup Auto Scaling.
- **AWS Command Line Interface (AWS CLI)** — Menyediakan perintah untuk serangkaian luas Layanan AWS, dan didukung pada Windows, macOS, dan Linux. Untuk memulai, lihat [Bersiap menggunakan alat AWS CLI](#). Untuk informasi selengkapnya, lihat [penskalaan otomatis](#) di Referensi Perintah AWS CLI
- **AWS Tools for Windows PowerShell**— Menyediakan perintah untuk serangkaian produk AWS yang luas bagi mereka yang membuat skrip di PowerShell lingkungan. Untuk memulai, lihat [AWS Tools for Windows PowerShell Panduan Pengguna](#). Untuk informasi lebih lanjut, lihat [AWS Tools for PowerShell Referensi Cmdlet](#).
- **AWS SDK** — Menyediakan operasi API khusus bahasa dan menangani banyak detail koneksi, seperti menghitung tanda tangan, menangani percobaan ulang permintaan, dan menangani kesalahan. Untuk informasi selengkapnya, lihat [AWS SDK](#).
- **Query API** - Menyediakan tindakan API tingkat rendah yang Anda panggil menggunakan permintaan HTTPS. Menggunakan Query API adalah cara paling langsung untuk mengakses Layanan AWS. Namun, aplikasi Anda harus menangani detail tingkat rendah seperti membuat hash untuk menandatangani permintaan, dan menangani kesalahan. Untuk informasi selengkapnya, lihat Referensi API [Auto Scaling Amazon EC2](#).
- **AWS CloudFormation**— Mendukung pembuatan grup Auto Scaling menggunakan CloudFormation template. Untuk informasi selengkapnya, lihat [Buat grup Auto Scaling dengan AWS CloudFormation](#).

Untuk terhubung secara terprogram ke sebuah Layanan AWS, Anda menggunakan endpoint. .

Manfaat Amazon EC2 Auto Scaling

Menambahkan Amazon EC2 Auto Scaling ke arsitektur aplikasi Anda adalah salah satu cara untuk memaksimalkan manfaat Cloud. AWS Ketika Anda menggunakan Amazon EC2 Auto Scaling, aplikasi Anda mendapatkan manfaat sebagai berikut:

- Toleransi kesalahan yang lebih baik. Amazon EC2 Auto Scaling dapat mendeteksi ketika suatu instance tidak sehat, mengakhirinya, dan meluncurkan instance untuk menggantikannya. Anda juga dapat mengonfigurasi Amazon EC2 Auto Scaling untuk menggunakan beberapa Zona Ketersediaan. Jika satu Zona Ketersediaan tidak tersedia, Amazon EC2 Auto Scaling dapat meluncurkan instance di zona lain untuk mengompensasi.
- Ketersediaan yang lebih baik. Amazon EC2 Auto Scaling membantu memastikan bahwa aplikasi Anda selalu memiliki kapasitas yang tepat untuk menangani permintaan lalu lintas saat ini.
- Manajemen biaya yang lebih baik. Amazon EC2 Auto Scaling dapat secara dinamis meningkatkan dan mengurangi kapasitas sesuai kebutuhan. Karena Anda membayar instance EC2 yang Anda gunakan, Anda menghemat uang dengan meluncurkan instance saat diperlukan dan menghentikannya ketika tidak membutuhkannya.

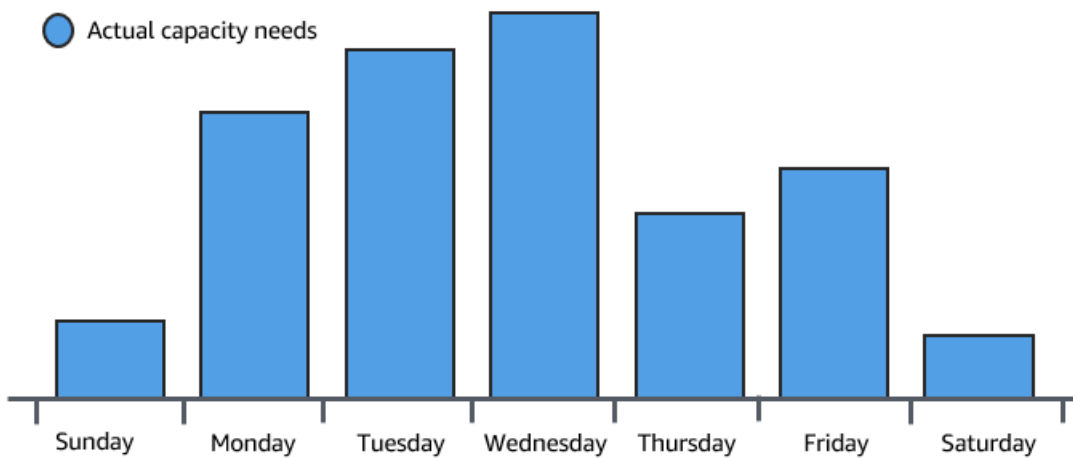
Daftar Isi

- [Contoh: Menutupi permintaan variabel](#)
- [Contoh: Arsitektur aplikasi web](#)
- [Contoh: Mendistribusikan instance di seluruh Availability Zone](#)
 - [Distribusi instance](#)
 - [Aktivitas rebalancing](#)

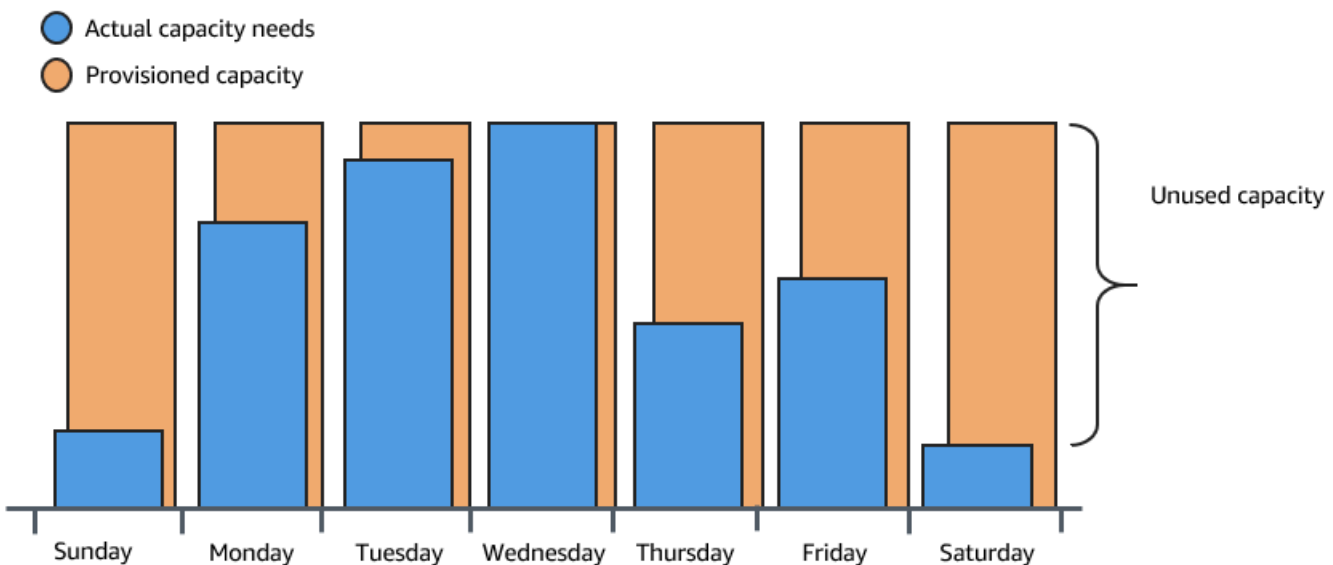
Contoh: Menutupi permintaan variabel

Untuk mendemonstrasikan beberapa manfaat Auto Scaling Amazon EC2, pertimbangkan aplikasi web dasar yang berjalan. AWS Aplikasi ini memungkinkan karyawan mencari ruang konferensi yang mungkin ingin mereka gunakan untuk rapat. Selama awal dan akhir minggu, penggunaan aplikasi ini adalah minimal. Selama pertengahan minggu, lebih banyak karyawan yang menjadwalkan pertemuan, sehingga permintaan pada aplikasi meningkat secara signifikan.

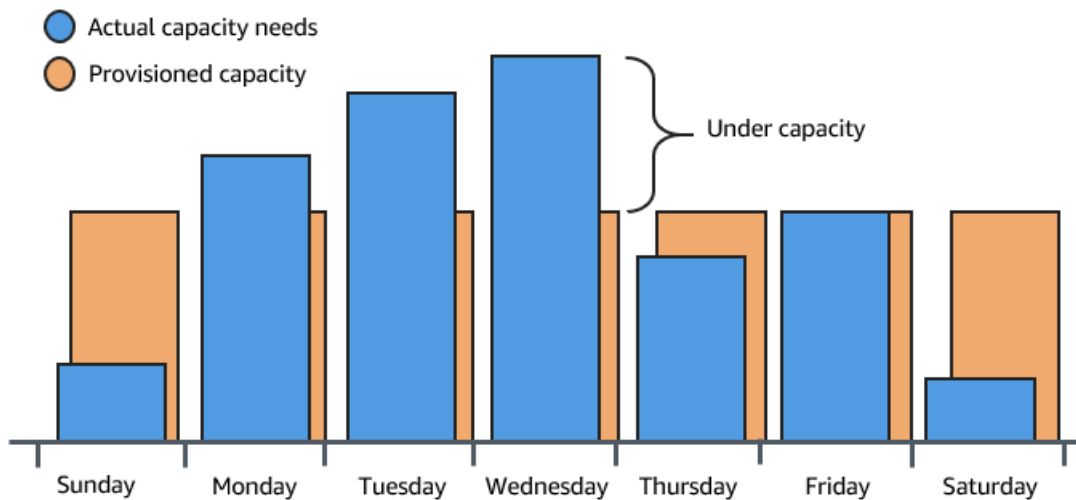
Grafik berikut menunjukkan jumlah kapasitas aplikasi yang digunakan selama seminggu.



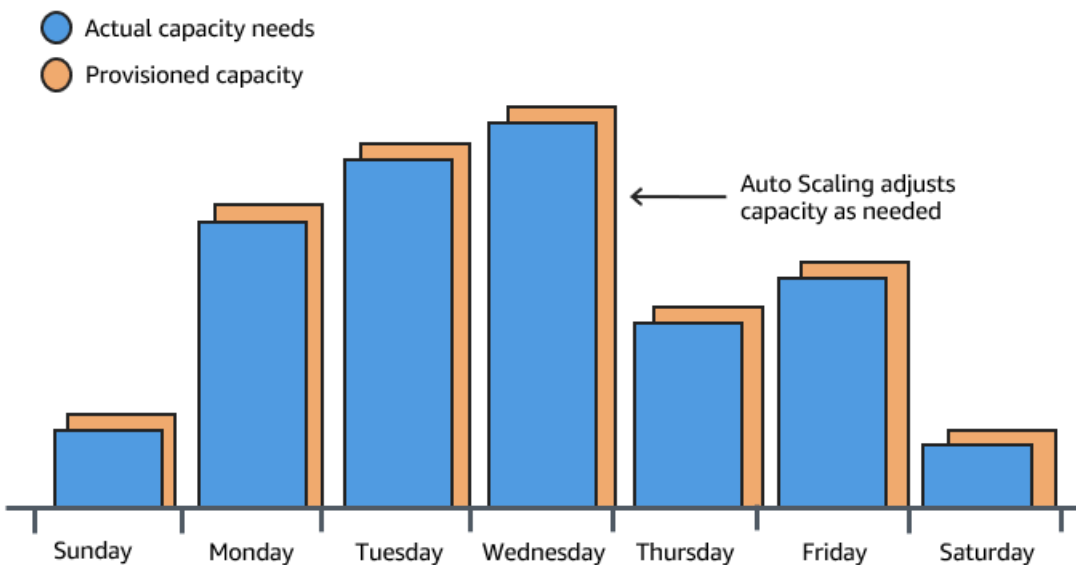
Secara tradisional, ada dua cara untuk merencanakan perubahan kapasitas ini. Opsi pertama adalah menambahkan cukup server agar aplikasi selalu memiliki kapasitas yang cukup untuk memenuhi permintaan. Namun, sisi negatif dari opsi ini adalah ada hari-hari ketika aplikasi tidak membutuhkan kapasitas yang besar. Kapasitas ekstra tetap tidak terpakai dan, pada dasarnya, meningkatkan biaya untuk menjaga aplikasi tetap berjalan.



Opsi kedua adalah memiliki kapasitas yang cukup untuk menangani permintaan rata-rata pada aplikasi. Opsi ini lebih murah, karena Anda tidak membeli peralatan yang hanya akan Anda gunakan sesekali. Namun, Anda berisiko menciptakan pengalaman pelanggan yang buruk ketika permintaan pada aplikasi melebihi kapasitasnya.



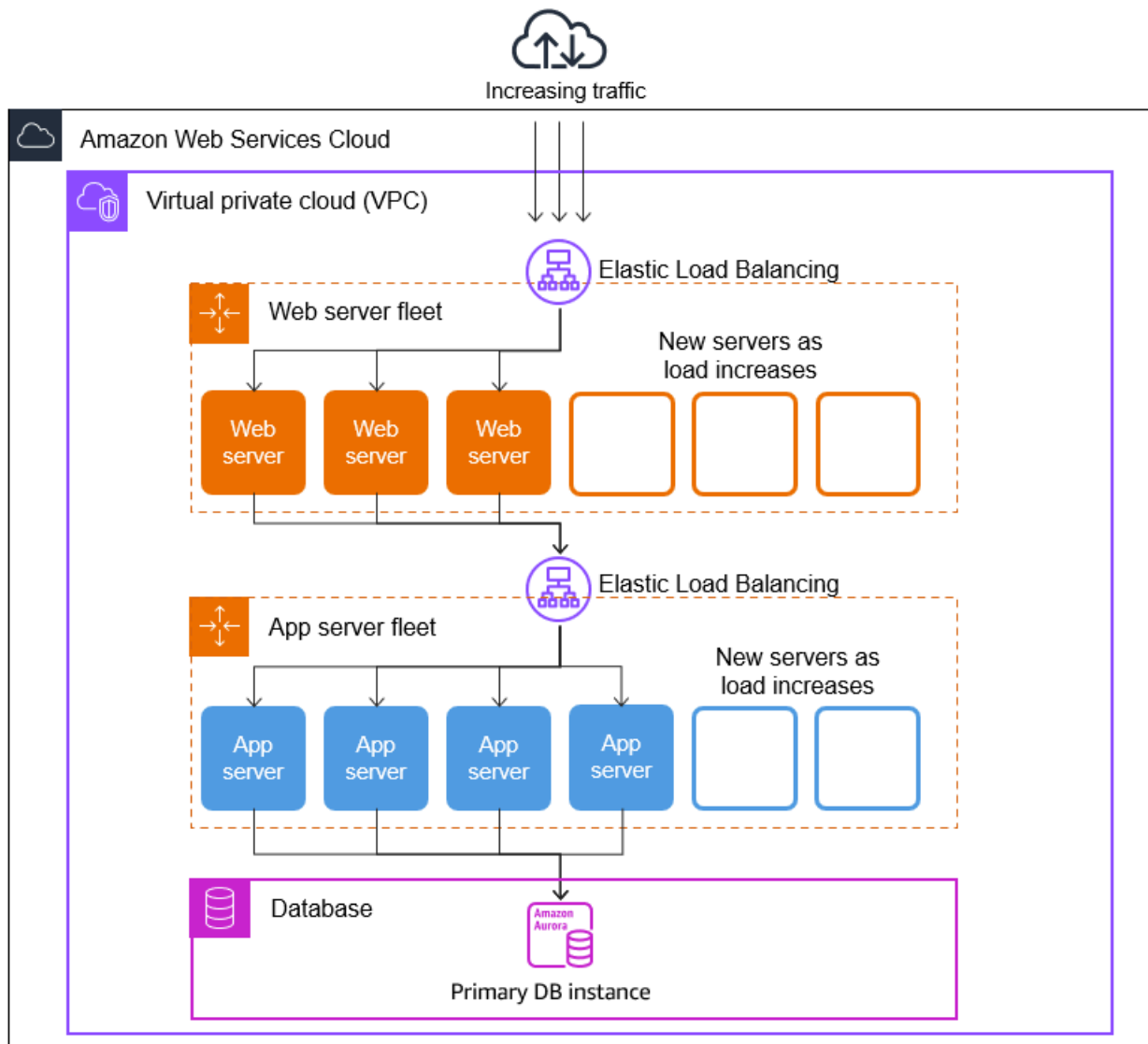
Dengan menambahkan Amazon EC2 Auto Scaling ke aplikasi ini, Anda memiliki opsi ketiga. Anda dapat menambahkan instance baru ke aplikasi hanya jika diperlukan, dan menghentikannya jika tidak lagi diperlukan. Karena Amazon EC2 Auto Scaling menggunakan instance EC2, Anda hanya perlu membayar untuk instance yang Anda gunakan, saat Anda menggunakannya. Sekarang Anda memiliki arsitektur hemat biaya yang memberikan pengalaman pelanggan terbaik dengan tetap meminimalkan biaya.



Contoh: Arsitektur aplikasi web

Dalam skenario aplikasi web yang umum, Anda menjalankan beberapa salinan aplikasi secara bersamaan untuk mencakup volume lalu lintas pelanggan Anda. Beberapa salinan aplikasi ini dihosting di instance EC2 (server cloud) yang identik, masing-masing menangani permintaan pelanggan.

Amazon EC2 Auto Scaling mengelola peluncuran dan penghentian instance EC2 ini atas nama Anda. Anda menentukan serangkaian kriteria (seperti CloudWatch alarm Amazon) yang menentukan kapan grup Auto Scaling meluncurkan atau menghentikan instans EC2. Penambahan grup Auto Scaling ke arsitektur jaringan Anda membantu membuat aplikasi Anda lebih tersedia dan toleran kegagalan.



Anda dapat membuat grup Auto Scaling sebanyak yang Anda perlukan. Misalnya, Anda dapat membuat grup Auto Scaling untuk setiap tingkat.

Untuk mendistribusikan lalu lintas di antara berbagai grup Auto Scaling Anda, Anda dapat memperkenalkan load balancer ke arsitektur Anda. Untuk informasi selengkapnya, lihat [Penyeimbang Beban Elastis](#).

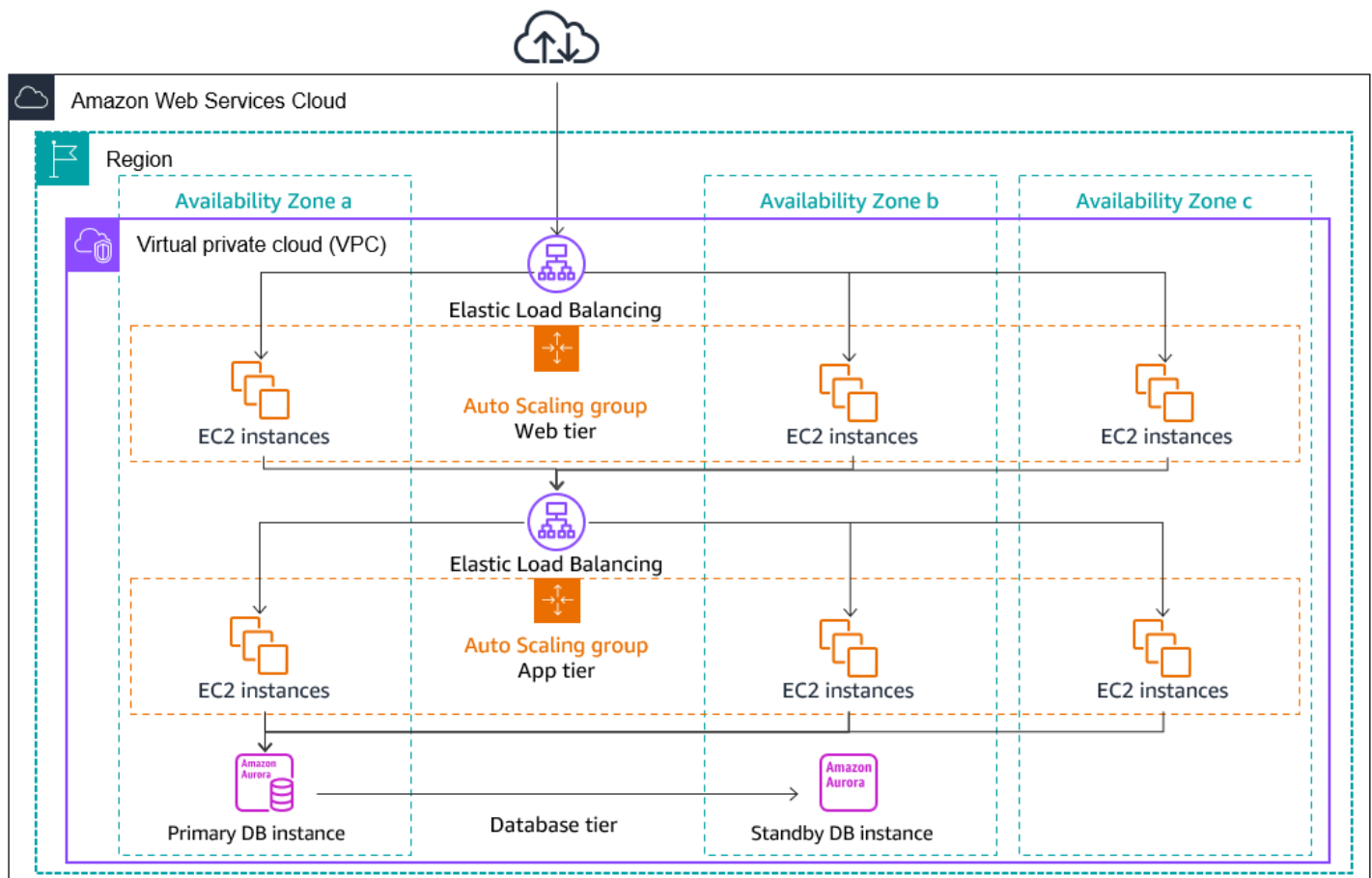
Contoh: Mendistribusikan instance di seluruh Availability Zone

Availability Zones adalah lokasi terisolasi di tempat tertentu Wilayah AWS. Setiap Wilayah memiliki beberapa Availability Zone yang dirancang untuk menyediakan ketersediaan tinggi untuk Wilayah. Availability Zones bersifat independen, dan karenanya Anda meningkatkan ketersediaan aplikasi saat mendesain aplikasi untuk menggunakan beberapa zona. Untuk informasi selengkapnya, lihat [Ketahanan di Amazon EC2 Auto Scaling](#).

Availability Zone diidentifikasi oleh Wilayah AWS kode diikuti oleh pengidentifikasi huruf (misalnya, us-east-1a). Jika Anda membuat VPC dan subnet daripada menggunakan VPC default, Anda dapat menentukan satu atau lebih subnet di setiap Availability Zone. Setiap subnet harus berada sepenuhnya dalam satu Availability Zone dan tidak dapat memperluas zona. Untuk informasi selengkapnya, lihat [Cara kerja Amazon VPC](#) di Panduan Pengguna Amazon VPC.

Saat Anda membuat grup Auto Scaling, Anda harus memilih VPC dan subnet tempat Anda akan menggunakan grup Auto Scaling. Auto Scaling Amazon EC2 membuat instans Anda di subnet pilihan Anda. Setiap instans dikaitkan dengan Availability Zone tertentu yang dipilih oleh Auto Scaling Amazon EC2. Saat instans diluncurkan, Amazon EC2 Auto Scaling mencoba mendistribusikannya secara merata di antara zona untuk ketersediaan dan keandalan yang tinggi.

Gambar berikut menunjukkan ikhtisar arsitektur multi-tier yang diterapkan di tiga Availability Zone.



Distribusi instance

Auto Scaling Amazon EC2 secara otomatis mencoba mempertahankan jumlah instans yang setara di setiap Availability Zone yang diaktifkan. Amazon EC2 Auto Scaling melakukan ini dengan mencoba meluncurkan instance baru di Zona Ketersediaan dengan beberapa instance paling sedikit. Jika ada beberapa subnet yang dipilih untuk Availability Zone, Auto Scaling Amazon EC2 memilih subnet dari Availability Zone secara acak. Namun, jika upaya tersebut gagal, Amazon EC2 Auto Scaling mencoba meluncurkan instance di Zona Ketersediaan lain hingga berhasil.

Dalam keadaan di mana Availability Zone menjadi tidak sehat atau tidak tersedia, distribusi instance mungkin menjadi tidak merata di seluruh Availability Zone. Saat Availability Zone pulih, Auto Scaling Amazon EC2 secara otomatis menyeimbangkan kembali grup Auto Scaling. Ini dilakukan dengan meluncurkan instance di Availability Zones yang diaktifkan dengan instans paling sedikit dan menghentikan instance di tempat lain.

Aktivitas rebalancing

Kegiatan penyeimbangan kembali terbagi dalam dua kategori: Penyeimbangan kembali Zona Ketersediaan dan penyeimbangan kembali kapasitas.

Penyeimbangan kembali Zona Ketersediaan

Setelah tindakan tertentu terjadi, grup Auto Scaling Anda dapat menjadi tidak seimbang antara Zona Ketersediaan. Amazon EC2 Auto Scaling memberikan kompensasi dengan rebalancing Zona Ketersediaan. Tindakan berikut dapat menyebabkan aktivitas rebalancing:

- Anda mengubah Availability Zone yang terkait dengan grup Auto Scaling Anda.
- Anda secara eksplisit menghentikan atau melepaskan instance atau menempatkan instance dalam keadaan siaga, dan kemudian grup menjadi tidak seimbang.
- Availability Zone yang sebelumnya memiliki kapasitas tidak mencukupi pulih dan sekarang memiliki kapasitas tambahan.
- Zona Ketersediaan yang sebelumnya memiliki harga Spot di atas harga maksimum Anda sekarang memiliki harga Spot di bawah harga maksimum Anda.

Saat menyeimbangkan kembali, Auto Scaling Amazon EC2 meluncurkan instans baru sebelum menghentikan instans sebelumnya. Dengan cara ini, penyeimbangan kembali tidak mengganggu kinerja atau ketersediaan aplikasi Anda.

Karena Auto Scaling Amazon EC2 mencoba meluncurkan instans baru sebelum menghentikan instans sebelumnya, berada pada atau mendekati kapasitas maksimum yang ditentukan dapat menghambat atau sepenuhnya menghentikan aktivitas penyeimbangan kembali.

Untuk menghindari masalah ini, sistem sementara dapat melebihi kapasitas maksimum yang ditentukan dari suatu kelompok selama aktivitas penyeimbangan kembali. Secara default, ia dapat melakukannya dengan margin 10 persen atau satu contoh, mana yang lebih besar. Margin diperpanjang hanya jika kelompok berada pada atau mendekati kapasitas maksimum dan perlu penyeimbangan kembali. Perpanjangan hanya berlangsung selama diperlukan untuk menyeimbangkan kembali kelompok (biasanya beberapa menit).

Atau, Anda dapat menetapkan ambang batas untuk grup Auto Scaling dengan menggunakan kebijakan pemeliharaan instans, dan grup hanya dapat menambah atau mengurangi kapasitas dalam rentang ambang tersebut. Dengan cara ini, Anda dapat mengontrol seberapa cepat grup Anda menyeimbangkan kembali dirinya sendiri. Untuk informasi selengkapnya, lihat [Kebijakan pemeliharaan instans](#).

Penyeimbangan Ulang Kapasitas

Anda dapat mengaktifkan Penyeimbangan Kembali Kapasitas untuk grup Auto Scaling saat menggunakan Instans Spot. Hal ini memungkinkan Amazon EC2 Auto Scaling mencoba meluncurkan Instans Spot setiap kali Amazon EC2 melaporkan bahwa Instans Spot berisiko tinggi mengalami gangguan. Setelah meluncurkan instance baru, itu kemudian mengakhiri instance sebelumnya. Untuk informasi selengkapnya, lihat [Gunakan Rebalancing Kapasitas untuk menangani interupsi Spot Amazon EC2](#).

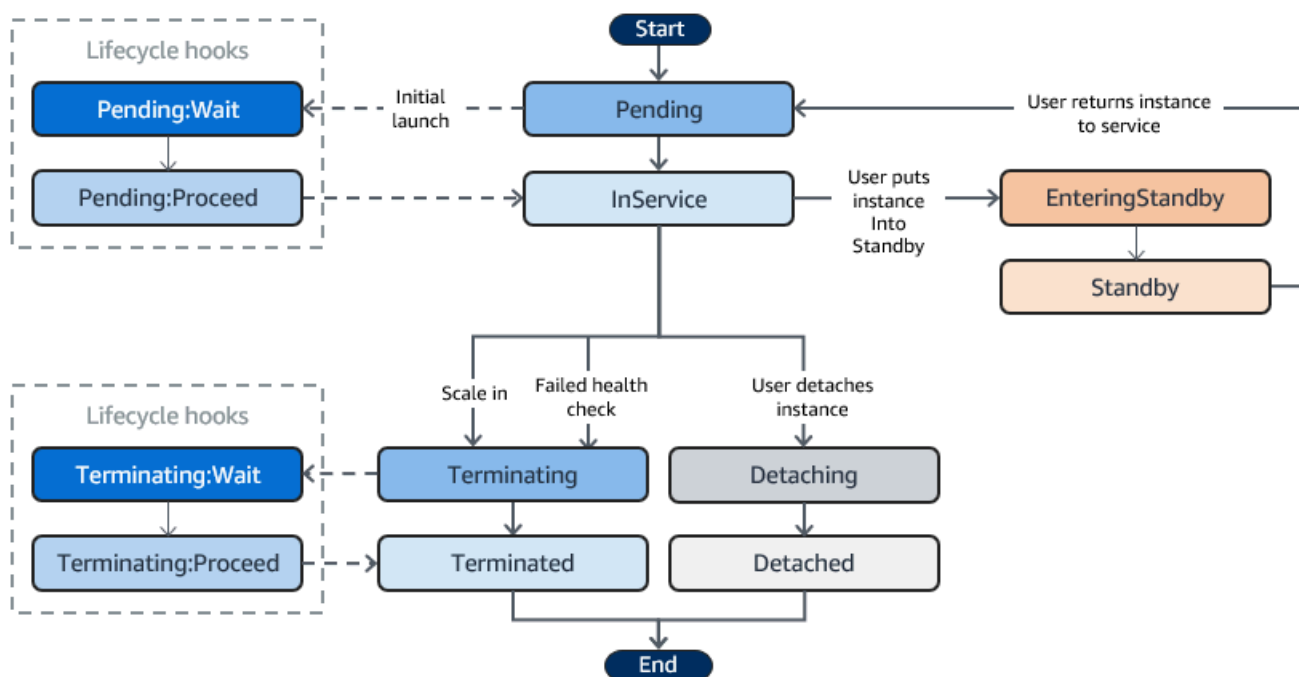
Siklus aktif instance Amazon EC2 Auto Scaling

Instance EC2 dalam grup Auto Scaling memiliki alur, atau siklus aktif, yang berbeda dari instance EC2 lainnya. Siklus aktif dimulai ketika grup Auto Scaling meluncurkan suatu instance dan menempatkannya dalam layanan. Siklus aktif berakhir saat Anda menghentikan instance, atau grup Auto Scaling menghentikan layanan dan menghentikannya.

Note

Anda ditagih untuk instance segera setelah diluncurkan, termasuk saat instance tersebut belum dalam layanan.

Ilustrasi berikut menunjukkan transisi antara status instance di siklus aktif Amazon EC2 Auto Scaling.



Meningkatkan skala

Peristiwa peningkatan skala berikut mengarahkan grup Auto Scaling untuk meluncurkan instance EC2 dan melampirkannya ke grup:

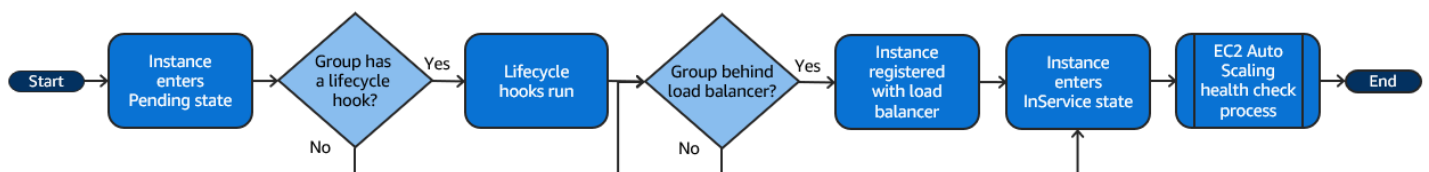
- Anda meningkatkan ukuran grup secara manual. Untuk informasi selengkapnya, lihat [Ubah kapasitas yang diinginkan dari grup Auto Scaling yang ada](#).
- Anda membuat kebijakan penskalaan untuk secara otomatis meningkatkan ukuran grup berdasarkan peningkatan permintaan yang telah ditentukan. Untuk informasi selengkapnya, lihat [Penskalaan dinamis untuk Amazon EC2 Auto Scaling](#).
- Anda menyiapkan penskalaan berdasarkan jadwal untuk meningkatkan ukuran grup pada waktu tertentu. Untuk informasi selengkapnya, lihat [Skala terjadwal untuk Amazon EC2 Auto Scaling](#).

Ketika peristiwa penskalaan terjadi, grup Auto Scaling meluncurkan jumlah instans EC2 yang diperlukan, menggunakan templat peluncuran yang ditetapkan. Instance ini dimulai dari status Pending. Jika Anda menambahkan hook siklus aktif ke grup Auto Scaling, Anda dapat melakukan tindakan khusus di sini. Untuk informasi selengkapnya, lihat [Hook siklus aktif](#).

Jika setiap instance sudah dikonfigurasi sepenuhnya dan lolos pemeriksaan kesehatan Amazon EC2, instance tersebut dilampirkan pada grup Auto Scaling dan masuk ke status InService. Instance dihitung terhadap kapasitas yang diinginkan dari grup Auto Scaling.

Jika grup Auto Scaling Anda dikonfigurasi untuk menerima lalu lintas dari penyeimbang beban Elastic Load Balancing, Auto Scaling Amazon EC2 secara otomatis mendaftarkan instans Anda dengan penyeimbang beban sebelum menandai instance sebagai InService

Berikut ini merangkum langkah-langkah untuk mendaftarkan instance dengan penyeimbang beban untuk acara scale-out.



Instance dalam layanan

Instance tetap dalam status InService hingga salah satu hal berikut terjadi:

- Peristiwa scale-in terjadi, dan Amazon EC2 Auto Scaling memilih untuk mengakhiri instance ini untuk mengurangi ukuran grup Auto Scaling. Untuk informasi selengkapnya, lihat [Kontrol instans Auto Scaling mana yang akan dihentikan selama penskalaan](#).
- Anda memasukkan instance tersebut ke dalam status Standby. Untuk informasi selengkapnya, lihat [Masukkan dan keluar dari siaga](#).
- Anda memisahkan instance dari grup Auto Scaling. Untuk informasi selengkapnya, lihat [Lepaskan atau lampirkan instance](#).
- Instance ini gagal dalam sejumlah pemeriksaan kesehatan yang diperlukan, sehingga dihilangkan dari grup Auto Scaling, dihentikan, dan diganti. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

Penurunan skala

Peristiwa penurunan skala berikut mengarahkan grup Auto Scaling untuk memisahkan instance EC2 dari grup dan menghentikannya:

- Anda mengurangi ukuran grup secara manual. Untuk informasi selengkapnya, lihat [Ubah kapasitas yang diinginkan dari grup Auto Scaling yang ada](#).
- Anda membuat kebijakan penskalaan untuk secara otomatis mengurangi ukuran grup berdasarkan penurunan permintaan yang telah ditentukan. Untuk informasi selengkapnya, lihat [Penskalaan dinamis untuk Amazon EC2 Auto Scaling](#).
- Anda mengatur penskalaan berdasarkan jadwal untuk mengurangi ukuran grup pada waktu tertentu. Untuk informasi selengkapnya, lihat [Skala terjadwal untuk Amazon EC2 Auto Scaling](#).

Penting bagi Anda untuk membuat peristiwa penurunan skala yang berhubungan untuk setiap peristiwa peningkatan skala yang Anda buat. Ini membantu memastikan bahwa sumber daya yang ditetapkan untuk aplikasi Anda sesuai dengan permintaan sumber daya tersebut sedekat mungkin.

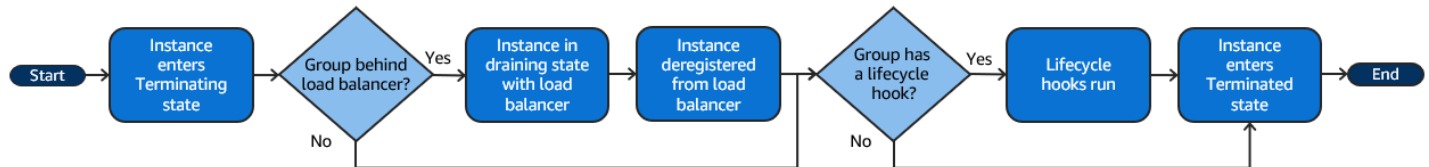
Ketika peristiwa scale-in terjadi, grup Auto Scaling mengakhiri satu atau beberapa instance. Auto Scaling group menggunakan kebijakan pemutusan hubungan kerja untuk menentukan situasi mana yang harus dihentikan. Instance yang sedang dalam proses penghentian dari grup Auto Scaling memasuki Terminating status, dan tidak dapat dimasukkan kembali ke layanan.

Jika grup Auto Scaling Anda dikonfigurasi untuk menerima lalu lintas dari penyeimbang beban Elastic Load Balancing, Auto Scaling Amazon EC2 otomatis membatalkan pendaftaran instance penghentian dari penyeimbang beban. Membatalkan pendaftaran instance memastikan bahwa semua permintaan

baru dialihkan ke instance lain di grup target penyeimbang beban sementara koneksi yang ada ke instance diizinkan untuk dilanjutkan hingga penundaan deregistrasi berakhir.

Jika menambahkan hook siklus hidup ke grup Auto Scaling, Anda dapat melakukan tindakan kustom pada instance penghentian. Untuk informasi selengkapnya, lihat [Hook siklus aktif](#). Akhirnya, instance sepenuhnya dihentikan dan memasuki Terminated negara bagian.

Berikut ini merangkum langkah-langkah untuk membatalkan pendaftaran instance dengan penyeimbang beban untuk acara scale-in.



Lepaskan satu instance

Anda dapat melepaskan satu instance dari grup Auto Scaling Anda. Setelah instance terlepas, Anda dapat mengelolanya secara terpisah dari grup Auto Scaling atau melampirkannya ke grup Auto Scaling yang berbeda.

Untuk informasi selengkapnya, lihat [Lepaskan atau lampirkan instance](#).

Melampirkan instance

Anda dapat melampirkan instance EC2 yang berjalan yang memenuhi kriteria tertentu ke grup Auto Scaling Anda. Setelah instance dilampirkan, hal ini dikelola sebagai bagian dari grup Auto Scaling.

Untuk informasi selengkapnya, lihat [Lepaskan atau lampirkan instance](#).

Hook siklus aktif

Anda dapat menambahkan pengait siklus hidup ke grup Auto Scaling Anda sehingga Anda dapat melakukan tindakan khusus saat peluncuran atau pengakhiran instance.

Ketika Amazon EC2 Auto Scaling menanggapi peristiwa peningkatan skala, maka akan diluncurkan satu atau lebih instance. Instance ini dimulai dari status Pending. Jika Anda menambahkan `autoscaling:EC2_INSTANCE_LAUNCHING` ke grup Auto Scaling Anda, instance beralih dari status Pending ke status Pending:Wait. Setelah Anda menyelesaikan tindakan siklus aktif, instance masuk ke status Pending:Proceed. Saat instance dikonfigurasi penuh, maka akan dilampirkan ke grup Auto Scaling dan akan masuk ke status InService.

Ketika Amazon EC2 Auto Scaling berhubungan dengan peristiwa penurunan skala, ini akan mengakhiri satu atau lebih instance. Instance-instance ini dipisahkan dari grup Auto Scaling dan masuk ke status `Terminating`. Jika Anda menambahkan `autoscaling:EC2_INSTANCE_TERMINATING` ke grup Auto Scaling Anda, instance beralih dari status `Terminating` ke status `Terminating:Wait`. Setelah Anda menyelesaikan tindakan siklus aktif, instance masuk ke status `Terminating:Proceed`. Saat instance benar-benar berakhir, instance tersebut masuk ke status `Terminated`.

Untuk informasi selengkapnya, lihat [Hook siklus aktif Amazon EC2 Auto Scaling](#).

Masukkan dan keluar dari siaga

Anda dapat memasukkan instance apa pun yang berada dalam status `InService` menjadi status `Standby`. Hal ini memungkinkan Anda untuk menghapus instance dari layanan, memecahkan masalah atau membuat perubahan untuk layanan, lalu menggunakannya kembali.

Instance dalam status `Standby` tetap dikelola oleh grup Auto Scaling. Namun demikian, semua itu bukan bagian aktif dari aplikasi Anda hingga Anda memasangnya kembali ke layanan.

Untuk informasi selengkapnya, lihat [Hapus instans sementara dari grup Auto Scaling](#).

Kuota untuk Auto Scaling Amazon EC2

Anda Akun AWS memiliki kuota default, sebelumnya disebut sebagai batas, untuk setiap layanan. AWS Kecuali dinyatakan sebaliknya, setiap kuota unik untuk suatu Wilayah. Anda dapat meminta penambahan untuk beberapa kuota, sementara kuota lainnya tidak dapat ditambah.

[Untuk melihat kuota untuk Amazon EC2 Auto Scaling, buka konsol Service Quotas](#). Di panel navigasi, pilih AWS layanan dan pilih Auto Scaling Amazon EC2.

Untuk meminta penambahan kuota, lihat [Meminta Penambahan Kuota](#) dalam Panduan Pengguna Kuota Layanan. Jika kuota belum tersedia di Service Quotas, gunakan formulir [Auto Scaling Limits](#). Kenaikan kuota terkait dengan Wilayah yang diminta.

Semua permintaan dikirimkan ke AWS Support. Anda dapat melacak kasus permintaan Anda di AWS Support konsol.

Sumber daya Amazon EC2 Auto Scaling

Anda Akun AWS memiliki kuota berikut yang terkait dengan jumlah grup Auto Scaling dan konfigurasi peluncuran yang dapat Anda buat.

Sumber daya	Kuota bawaan
Grup Auto Scaling per wilayah	500
Konfigurasi peluncuran per wilayah	200

Konfigurasi grup Auto Scaling

Anda Akun AWS memiliki kuota berikut yang terkait dengan konfigurasi grup Auto Scaling. Hal ini tidak dapat diubah.

Sumber Daya	Kuota
Kebijakan penskalaan per grup Auto Scaling	50
Tindakan terjadwal per grup Auto Scaling	125
Penyesuaian langkah per kebijakan penskalaan langkah	20
Pengait siklus hidup per grup Auto Scaling	50
Topik SNS per grup Auto Scaling	10
Classic Load Balancer per grup Auto Scaling	50
Kelompok target Elastic Load Balancing per grup Auto Scaling	50
Grup target VPC Lattice untuk grup Auto Scaling	5

Operasi API grup Auto Scaling

Auto Scaling Amazon EC2 menyediakan operasi API untuk membuat perubahan pada grup Auto Scaling Anda dalam batch. Berikut ini adalah batas API pada jumlah maksimum item (anggota array maksimum) yang diizinkan dalam satu operasi. Hal ini tidak dapat diubah.

Operasi	Anggota array maksimum
AttachInstances	20 ID contoh

Operasi	Anggota array maksimum
AttachLoadBalancers	10 penyeimbang beban
AttachLoadBalancerTargetGroups	10 kelompok sasaran
BatchDeleteScheduledAction	50 tindakan terjadwal
BatchPutScheduledUpdateGroupAction	50 tindakan terjadwal
DetachInstances	20 ID contoh
DetachLoadBalancers	10 penyeimbang beban
DetachLoadBalancerTargetGroups	10 kelompok sasaran
EnterStandby	20 ID contoh
ExitStandby	20 ID contoh
SetInstanceProtection	50 ID contoh

Minta pembatasan untuk API Auto Scaling Amazon EC2

Permintaan API Auto Scaling Amazon EC2 dibatasi menggunakan skema token bucket untuk mempertahankan bandwidth layanan. Untuk informasi selengkapnya, lihat [Rasio permintaan API](#) di Referensi API Auto Scaling Amazon EC2.

Tingkat penghentian EC2

Auto Scaling Amazon EC2 secara dinamis menentukan jumlah operasi penghentian instans EC2 yang dapat dilakukannya pada saat grup Auto Scaling Anda masuk. Ini berarti Anda mungkin melihat variasi dalam jumlah instance yang dihentikan pada satu waktu di seluruh grup Auto Scaling. Variasi ini disebabkan oleh pertimbangan eksternal, seperti apakah Auto Scaling Amazon EC2 harus membatalkan pendaftaran instans dengan penyeimbang beban.

Layanan lainnya

Kuota untuk layanan lain, seperti Amazon EC2 dan Amazon VPC, dapat memengaruhi grup Auto Scaling Anda. Anda dapat menggunakan Service Quotas untuk memperbarui kuota untuk

instans EC2 dan sumber daya lainnya di Anda. Akun AWS Di Service Quotas konsol, Anda dapat melihat semua kuota layanan yang tersedia dan meminta kenaikan untuk mereka. Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota](#) di Service Quotas Panduan Pengguna.

Untuk kuota yang khusus untuk meluncurkan template, lihat [Meluncurkan pembatasan template](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Menyiapkan Amazon EC2 Auto Scaling

Sebelum mulai menggunakan Amazon EC2 Auto Scaling, selesaikan tugas berikut.

Tugas

- [Bersiap menggunakan Amazon EC2](#)
- [Bersiap menggunakan alatAWS CLI](#)

Bersiap menggunakan Amazon EC2

Jika Anda belum pernah menggunakan Amazon EC2, selesaikan tugas yang dijelaskan dalam dokumentasi Amazon EC2. Untuk informasi lebih lanjut, lihat [Menyiapkan dengan Amazon EC2](#) dalam Panduan Pengguna Amazon EC2 untuk Instance Linux atau [Menyiapkan dengan Amazon EC2](#) dalam Panduan Pengguna Amazon EC2 untuk Windows Instances.

Bersiap menggunakan alatAWS CLI

Anda dapat menggunakan alat barisAWS perintah untuk mengeluarkan perintah pada baris perintah sistem Anda guna menjalankan Amazon EC2 Auto Scaling danAWS tugas lain.

Untuk menggunakanAWS Command Line Interface (AWS CLI), unduh, instal, dan konfigurasi versi 1 atau 2 dari fileAWS CLI. Fungsionalitas Amazon EC2 Auto Scaling yang sama tersedia dalam versi 1 dan 2. Untuk menginstalAWS CLI versi 1, lihat [Menginstal, memperbarui, dan menghapus instalasiAWS CLI di](#) Panduan PenggunaAWS CLI Versi 1. Untuk menginstalAWS CLI versi 2, lihat [Menginstal atau memperbarui versi terbaru dariAWS CLI](#) dalam Panduan PenggunaAWS CLI Versi 2.

AWS CloudShellmemungkinkan Anda melewati instalasiAWS CLI di lingkungan pengembangan Anda, dan menggunakannya diAWS Management Console sebagai gantinya. Selain menghindari penginstalan, Anda juga tidak perlu mengonfigurasi kredensi, dan Anda tidak perlu menentukan wilayah. AWS Management ConsoleSesi Anda memberikan konteks ini keAWS CLI. Anda dapat menggunakanAWS CloudShell di didukungWilayah AWS. Untuk informasi selengkapnya, lihat [Buat grup Auto Scaling dari baris perintah menggunakan AWS CloudShell](#).

Untuk informasi selengkapnya, lihat [penskalaan otomatis](#) di ReferensiAWS CLI Perintah.

Memulai dengan Auto Scaling Amazon EC2

Untuk memulai dengan Amazon EC2 Auto Scaling, Anda dapat mengikuti tutorial yang memperkenalkan Anda pada layanan ini.

Topik

- [Tutorial: Buat grup Auto Scaling pertama Anda](#)
- [Tutorial: Siapkan aplikasi yang diskalakan dan diseimbangkan beban](#)

Untuk tutorial tambahan yang berfokus pada alat khusus untuk mengelola siklus hidup instance dalam grup Auto Scaling, lihat topik berikut:

- [Tutorial: Konfigurasi kait siklus hidup yang memanggil fungsi Lambda](#). Tutorial ini menunjukkan cara menggunakan Amazon EventBridge untuk membuat aturan yang menjalankan fungsi Lambda berdasarkan peristiwa yang terjadi pada instance di grup Auto Scaling Anda.
- [Tutorial: Konfigurasi data pengguna untuk mengambil status siklus hidup target melalui metadata instance](#). Tutorial ini menunjukkan cara menggunakan Layanan Metadata Instance (IMDS) untuk menjalankan tindakan dari dalam instance itu sendiri.

Sebelum Anda membuat grup Auto Scaling untuk digunakan dengan aplikasi Anda, tinjau aplikasi Anda secara menyeluruh saat berjalan di AWS Cloud. Pertimbangkan hal berikut:

- Jumlah Availability Zone yang harus dicakup oleh grup Auto Scaling.
- Sumber daya yang sudah ada seperti apa yang dapat digunakan, seperti grup keamanan atau Amazon Machine Images (AMI).
- Apakah Anda ingin meningkatkan atau mengurangi kapasitas, atau hanya ingin memastikan bahwa jumlah server tertentu selalu berjalan. Harap diingat bahwa Amazon EC2 Auto Scaling dapat melakukan keduanya secara bersamaan.
- Metrik apa yang paling sesuai dengan kinerja aplikasi Anda.
- Berapa lama waktu yang dibutuhkan untuk meluncurkan dan menyediakan server.

Makin baik Anda memahami aplikasi, Anda akan makin efektif untuk membuat arsitektur Auto Scaling.

Tutorial: Buat grup Auto Scaling pertama Anda

Tutorial ini memberikan pengenalan langsung untuk Amazon EC2 Auto Scaling melalui AWS Management Console. Anda akan membuat template peluncuran yang mendefinisikan instans EC2 Anda dan grup Auto Scaling dengan satu instance di dalamnya. Setelah meluncurkan grup Auto Scaling, Anda akan menghentikan instance dan memverifikasi bahwa instance telah dihapus dari layanan dan diganti. Untuk mempertahankan jumlah instans yang konstan, Auto Scaling Amazon EC2 mendeteksi dan merespons pemeriksaan kesehatan dan jangkauan Amazon EC2 secara otomatis.

[Saat mendaftar AWS, Anda dapat memulai dengan Amazon EC2 Auto Scaling secara gratis menggunakan AWS Tingkat Gratis.](#) Anda dapat menggunakan tingkat gratis untuk meluncurkan dan menggunakan instans t2.micro secara gratis selama 12 bulan (di Wilayah yang tidak menyediakan t2.micro, Anda dapat menggunakan instans t3.micro pada tingkat gratis). Jika Anda meluncurkan instans yang tidak termasuk dalam tingkat gratis, Anda dikenai biaya penggunaan Amazon EC2 standar untuk instans tersebut. Untuk informasi selengkapnya, lihat [Harga Amazon EC2](#).

Tugas

- [Bersiaplah untuk penelusuran](#)
- [Langkah 1: Buat templat peluncuran](#)
- [Langkah 2: Buat grup Auto Scaling satu contoh](#)
- [Langkah 3: Verifikasi grup Auto Scaling Anda](#)
- [Langkah 4: Menghentikan instans di grup Auto Scaling](#)
- [Langkah 5: Langkah selanjutnya](#)
- [Langkah 6: Bersihkan](#)

Bersiaplah untuk penelusuran

Panduan ini mengasumsikan bahwa Anda terbiasa meluncurkan instans EC2 dan bahwa Anda telah membuat key pair dan grup keamanan. Untuk informasi selengkapnya, lihat [Menyiapkan dengan Amazon EC2](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk mulai menggunakan Amazon EC2 Auto Scaling, Anda dapat menggunakan VPC default untuk Anda. Akun AWS VPC default menyertakan subnet publik default di setiap Availability Zone dan

gateway internet yang dilampirkan ke VPC Anda. Anda dapat melihat VPC Anda di [halaman VPC Anda](#) di konsol Amazon Virtual Private Cloud (Amazon VPC).

Langkah 1: Buat templat peluncuran

Pada langkah ini, Anda membuat template peluncuran yang menentukan jenis instans EC2 yang dibuat oleh Amazon EC2 Auto Scaling untuk Anda. Sertakan informasi seperti ID Amazon Machine Image (AMI) untuk digunakan, jenis instans, pasangan kunci, dan grup keamanan.

Untuk membuat templat peluncuran

1. Buka konsol Amazon EC2 dan buka halaman [Peluncuran templat](#).
2. Di bilah navigasi atas, pilih file Wilayah AWS. Template peluncuran dan grup Auto Scaling yang Anda buat terikat dengan Wilayah yang Anda tentukan.
3. Pilih Buat templat peluncuran.
4. Untuk nama template Luncurkan, masukkan **my-template-for-auto-scaling**.
5. Di bawah panduan Auto Scaling, pilih kotak centang.
6. Untuk Aplikasi dan Gambar OS (Amazon Machine Image), pilih versi Amazon Linux 2 (HVM) dari daftar Mulai Cepat. AMI berfungsi sebagai template konfigurasi dasar untuk instans Anda.
7. Untuk Jenis instans, pilih konfigurasi perangkat keras yang kompatibel dengan AMI yang Anda tentukan.
8. (Opsional) Untuk Key pair (login), pilih key pair yang ada. Anda menggunakan pasangan kunci untuk terhubung ke instans Amazon EC2 dengan SSH. Koneksi ke suatu instans tidak disertakan sebagai bagian dari tutorial ini. Oleh karena itu, Anda tidak perlu menentukan key pair kecuali Anda berniat untuk terhubung ke instance Anda menggunakan SSH.
9. Untuk pengaturan Jaringan, perluas Konfigurasi jaringan lanjutan dan lakukan hal berikut:
 - a. Pilih Tambahkan antarmuka jaringan untuk mengkonfigurasi antarmuka jaringan utama.
 - b. Untuk Auto-assign IP publik, tentukan apakah instans Anda menerima alamat IPv4 publik. Secara default, Amazon EC2 menetapkan alamat IPv4 publik jika instans EC2 diluncurkan ke subnet default atau jika instans diluncurkan ke subnet yang telah dikonfigurasi untuk secara otomatis menetapkan alamat IPv4 publik. Jika Anda tidak perlu terhubung ke instans Anda, pilih Nonaktifkan.
 - c. Untuk ID grup Keamanan, pilih grup keamanan di VPC yang sama dengan yang Anda rencanakan untuk digunakan sebagai VPC untuk grup Auto Scaling Anda. Jika Anda tidak

menentukan grup keamanan, instans Anda secara otomatis terkait dengan grup keamanan default untuk VPC.

- d. Untuk Hapus saat penghentian, pilih Ya untuk menghapus antarmuka jaringan saat instance dihapus.

10. Pilih Buat templat peluncuran.

11. Pada halaman konfirmasi, pilih Buat grup Auto Scaling.

Langkah 2: Buat grup Auto Scaling satu contoh

Gunakan prosedur berikut untuk melanjutkan di mana Anda tinggalkan setelah membuat template peluncuran.

Untuk membuat grup Auto Scaling

1. Pada halaman Pilih template peluncuran atau konfigurasi, untuk nama grup Auto Scaling, masukkan. **my-first-asg**
2. Pilih Berikutnya.

Halaman opsi peluncuran instans Pilih muncul, memungkinkan Anda memilih pengaturan jaringan VPC yang ingin digunakan oleh grup Auto Scaling dan memberi Anda opsi untuk meluncurkan Instans Sesuai Permintaan dan Spot.

3. Di bagian Jaringan, tetapkan VPC ke VPC default untuk pilihan Anda Wilayah AWS, atau pilih VPC Anda sendiri. VPC default secara otomatis dikonfigurasi untuk menyediakan konektivitas internet untuk instans Anda. VPC ini mencakup subnet publik di setiap Availability Zone di Wilayah tersebut.
4. Untuk Availability Zones dan subnet, pilih subnet dari setiap Availability Zone yang ingin Anda sertakan. Gunakan subnet di beberapa Availability Zone untuk ketersediaan tinggi. Untuk informasi selengkapnya, lihat [Pertimbangan saat memilih subnet VPC](#).
5. Di bagian Persyaratan tipe instans, gunakan pengaturan default untuk menyederhanakan langkah ini. (Jangan mengganti template peluncuran.) Untuk tutorial ini, Anda hanya akan meluncurkan satu On-Demand Instance menggunakan jenis instance yang ditentukan dalam template peluncuran Anda.
6. Simpan sisa default untuk tutorial ini, lalu pilih Lewati untuk meninjau.

Note

Ukuran awal grup ditentukan oleh kapasitas yang diinginkan. Nilai default adalah instans 1.

7. Di halaman Peninjauan, tinjau informasi untuk grup, lalu pilih Buat grup Auto Scaling.

Langkah 3: Verifikasi grup Auto Scaling Anda

Setelah Anda membuat grup Auto Scaling, Anda siap untuk memverifikasi bahwa grup telah meluncurkan instans EC2.

Tip

Dalam prosedur berikut, Anda melihat bagian Riwayat Aktivitas dan Instans untuk grup Auto Scaling. Di keduanya, kolom bernama seharusnya sudah ditampilkan. Untuk menampilkan kolom tersembunyi atau mengubah jumlah baris yang ditampilkan, pilih ikon roda gigi di sudut kanan atas setiap bagian untuk membuka modal preferensi, perbarui pengaturan sesuai kebutuhan, dan pilih Konfirmasi.

Untuk memverifikasi bahwa grup Auto Scaling Anda telah meluncurkan instans EC2

1. Buka [halaman grup Auto Scaling](#) di konsol Amazon EC2.
2. Pilih kotak centang di samping grup Auto Scaling yang baru saja Anda buat.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling. Tab pertama yang tersedia adalah tab Detail, menampilkan informasi tentang grup Auto Scaling.

3. Pilih tab kedua, Aktivitas. Di bawah Riwayat aktivitas, Anda dapat melihat kemajuan aktivitas yang terkait dengan grup Auto Scaling. Kolom Status menunjukkan status instans Anda saat ini. Saat instance Anda diluncurkan, kolom status akan ditampilkan `Not yet in service`. Status berubah menjadi `Successful` setelah instance diluncurkan. Anda juga dapat menggunakan tombol refresh untuk melihat status instans Anda saat ini.
4. Di tab Manajemen instans, di bawah Instans, Anda dapat melihat status instans.
5. Verifikasi bahwa instans Anda berhasil diluncurkan. Butuh waktu singkat untuk meluncurkan sebuah instans.

- Kolom Siklus Aktif menunjukkan status instans Anda. Pada awalnya, instans Anda ada pada status Pending. Setelah instans siap untuk menerima lalu lintas, statusnya adalah InService.
- Kolom status Kesehatan menunjukkan hasil pemeriksaan kesehatan Auto Scaling Amazon EC2 pada instans Anda.

Langkah 4: Menghentikan instans di grup Auto Scaling

Gunakan langkah-langkah ini untuk mempelajari lebih lanjut tentang cara kerja Auto Scaling Amazon EC2, khususnya, cara meluncurkan instans baru bila diperlukan. Ukuran minimum untuk grup Auto Scaling yang Anda buat dalam tutorial ini adalah satu instans. Oleh karena itu, jika Anda mengakhiri instans yang berjalan, Amazon EC2 Auto Scaling harus meluncurkan instans baru untuk menggantikannya.

1. Buka [halaman grup Auto Scaling](#) di konsol Amazon EC2.
2. Pilih kotak centang di samping grup Auto Scaling Anda.
3. Di tab Manajemen instans, di bawah Instans, pilih ID instans.

Ini akan membawa Anda ke halaman Instans konsol Amazon EC2, tempat Anda dapat menghentikan instans.

4. Pilih Tindakan, Status Instans, Akhiri. Ketika diminta konfirmasi, pilih Ya, Akhiri.
5. Pada panel navigasi, di bawah Auto Scaling, pilih Grup Auto Scaling. Pilih grup Auto Scaling Anda dan pilih tab Aktivitas.

Saat Anda menghentikan instance dari halaman Instances, dibutuhkan satu atau dua menit setelah Anda menghentikan instance sebelum instance baru diluncurkan. Dalam riwayat aktivitas, ketika aktivitas penskalaan dimulai, Anda melihat entri untuk pengakhiran instans pertama dan entri untuk peluncuran instans baru. Gunakan tombol refresh sampai Anda melihat entri baru.

6. Di tab Manajemen instans, bagian Instans hanya menunjukkan instans baru.
7. Pada panel navigasi, di bawah Instans, pilih Instans. Halaman ini menampilkan baik instans yang diakhiri maupun instans yang baru berjalan.

Langkah 5: Langkah selanjutnya

Pergi ke langkah berikutnya jika Anda ingin menghapus infrastruktur dasar yang baru saja Anda buat. Jika tidak, Anda dapat menggunakan infrastruktur ini sebagai dasar Anda dan mencoba satu atau beberapa hal berikut:

- Connect ke instans Linux Anda menggunakan Session Manager atau SSH. Untuk informasi selengkapnya, lihat [Connect ke instans Linux menggunakan Session Manager](#) dan [Connect ke instans Linux dari Linux atau macOS menggunakan SSH](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Konfigurasi notifikasi Amazon SNS untuk memberi tahu Anda kapan pun grup Auto Scaling meluncurkan atau menghentikan instans. Untuk informasi selengkapnya, lihat [Opsi notifikasi Amazon SNS](#).
- Skalakan grup Auto Scaling Anda secara manual untuk menguji notifikasi SNS. Untuk informasi selengkapnya, lihat [Ubah kapasitas yang diinginkan dari grup Auto Scaling Anda](#).

Anda juga dapat mulai membiasakan diri dengan konsep penskalaan otomatis dengan membaca tentang [Kebijakan penskalaan pelacakan target](#). Jika beban pada aplikasi Anda berubah, grup Auto Scaling Anda dapat menskalakan (menambahkan instance) dan menskalakan (menjalankan lebih sedikit instance) secara otomatis dengan menyesuaikan kapasitas grup yang diinginkan antara batas kapasitas minimum dan maksimum. Untuk informasi selengkapnya tentang menetapkan batas ini, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).

Langkah 6: Bersihkan

Anda dapat menghapus infrastruktur penskalaan Anda atau menghapus hanya grup Auto Scaling Anda dan menyimpan template peluncuran Anda untuk digunakan nanti.

Jika Anda meluncurkan instans yang tidak berada dalam [Tingkat AWS Gratis](#), Anda harus menghentikan instans Anda untuk menghindari biaya tambahan. Saat Anda mengakhiri instans, data yang terkait dengannya juga akan dihapus.

Untuk menghapus grup Auto Scaling Anda

1. Buka [halaman grup Auto Scaling](#) di konsol Amazon EC2.
2. Pilih kotak centang di samping grup Auto Scaling (`my-first-asg`).
3. Pilih Hapus.

4. Saat diminta konfirmasi, ketik **delete** untuk mengonfirmasi penghapusan grup Auto Scaling yang ditentukan, lalu pilih Hapus.

Ikon memuat di kolom Nama menunjukkan bahwa grup Auto Scaling dihapus. Saat penghapusan telah terjadi, kolom Diinginkan, Min, dan Maks menunjukkan instans 0 untuk grup Auto Scaling. Hanya perlu beberapa menit untuk menghentikan proses dan menghapus grup. Refresh daftar untuk melihat status saat ini.

Lewati prosedur berikut jika Anda ingin menyimpan templat peluncuran Anda.

Untuk menghapus templat peluncuran Anda

1. Buka [halaman Peluncuran templat](#) konsol Amazon EC2.
2. Pilih template peluncuran Anda (my-template-for-auto-scaling).
3. Pilih Tindakan, Hapus templat.
4. Saat diminta konfirmasi, ketik **Delete** untuk mengonfirmasi penghapusan templat peluncuran yang ditentukan, lalu pilih Hapus.

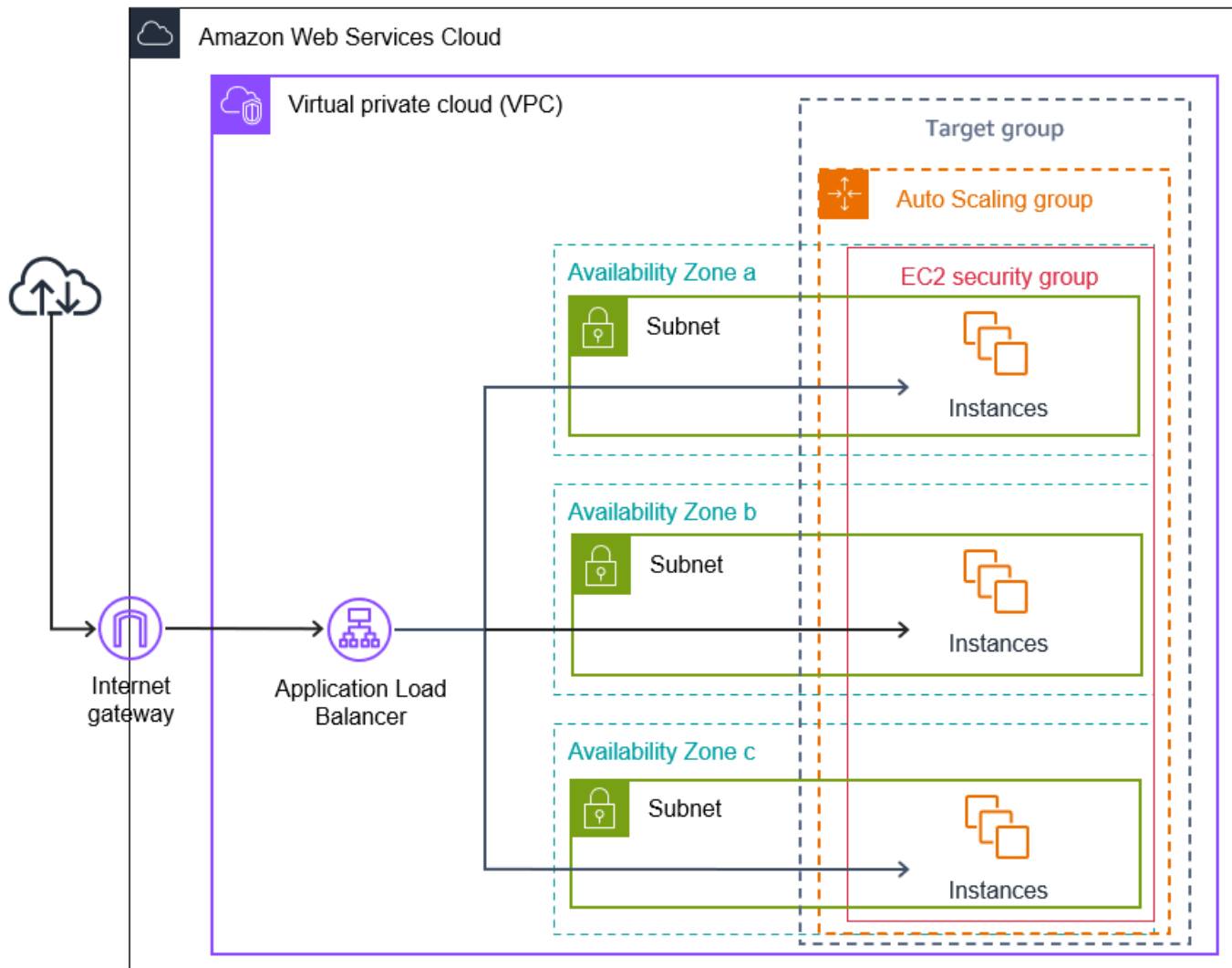
Tutorial: Siapkan aplikasi yang diskalakan dan diseimbangkan beban

Important

Sebelum Anda menjelajahi tutorial ini, kami sarankan Anda terlebih dahulu meninjau tutorial pengantar berikut: [Buat grup Auto Scaling pertama Anda](#).

Mendaftarkan grup Auto Scaling Anda dengan load balancer Elastic Load Balancing membantu Anda menyiapkan aplikasi load-balanced. Elastic Load Balancing bekerja dengan Amazon EC2 Auto Scaling untuk mendistribusikan lalu lintas masuk di seluruh instance Amazon EC2 Anda yang berstatus sehat. Ini meningkatkan skalabilitas dan ketersediaan aplikasi Anda. Anda dapat mengaktifkan Elastic Load Balancing dalam beberapa Zona Ketersediaan untuk meningkatkan toleransi kesalahan aplikasi Anda.

Dalam tutorial ini, kami membahas langkah-langkah dasar untuk pengaturan aplikasi load-balanced ketika grup Auto Scaling dibuat. Ketika selesai, arsitektur Anda akan terlihat mirip dengan diagram berikut:



Elastic Load Balancing mendukung berbagai jenis penyeimbang beban. Kami menyarankan Anda menggunakan Application Load Balancer untuk tutorial ini.

Untuk informasi selengkapnya tentang memperkenalkan penyeimbang beban ke dalam arsitektur Anda, lihat [Gunakan Elastic Load Balancing untuk mendistribusikan lalu lintas di seluruh instans dalam grup Auto Scaling](#).

Tugas

- [Prasyarat](#)
- [Langkah 1: Siapkan templat peluncuran atau konfigurasi peluncuran](#)

- [Langkah 2: Buat grup Auto Scaling](#)
- [Langkah 3: Verifikasi bahwa penyeimbang beban Anda terpasang](#)
- [Langkah 4: Langkah selanjutnya](#)
- [Langkah 5: Bersihkan](#)
- [Sumber daya terkait](#)

Prasyarat

- Penyeimbang beban dan kelompok sasaran. Pastikan untuk memilih Availability Zone yang sama untuk load balancer yang Anda rencanakan untuk digunakan untuk grup Auto Scaling Anda. Untuk informasi lebih lanjut, lihat [Memulai Elastic Load Balancing](#) di Panduan Pengguna Elastic Load Balancing.
- Grup keamanan untuk templat peluncuran atau konfigurasi peluncuran Anda. Grup keamanan harus mengizinkan akses dari penyeimbang beban pada kedua port listener (biasanya port 80 untuk lalu lintas HTTP) dan port yang Anda inginkan Elastic Load Balancing untuk digunakan untuk pemeriksaan kesehatan. Untuk informasi lebih lanjut, lihat dokumentasi yang berlaku:
 - [Grup keamanan target](#) di Panduan Pengguna untuk Application Load Balancer
 - [Grup keamanan target](#) di Panduan Pengguna untuk Network Load Balancer

Secara opsional, jika instans Anda akan memiliki alamat IP publik, Anda dapat mengizinkan lalu lintas SSH untuk menghubungkan ke instance.

- (Opsional) Peran IAM yang memberikan akses aplikasi Anda. AWS
- (Opsional) Gambar Mesin Amazon (AMI) yang didefinisikan sebagai template sumber untuk instans Amazon EC2 Anda. Untuk membuatnya sekarang, luncurkan satu instance. Tentukan peran IAM (jika Anda sudah membuatnya) dan skrip konfigurasi yang diperlukan sebagai data pengguna. Hubungkan ke instance dan sesuaikan. Misalnya, Anda dapat menginstal perangkat lunak dan aplikasi, menyalin data, dan melampirkan volume EBS tambahan. Uji aplikasi Anda pada instans Anda untuk memastikan bahwa itu dikonfigurasi dengan benar. Simpan konfigurasi yang telah diperbarui ini sebagai AMI khusus. Jika Anda tidak memerlukan instance nanti, Anda dapat menghentikannya. Instans yang diluncurkan dari AMI kustom tersebut sudah mencakup kustomisasi yang Anda buat saat membuat AMI.
- Virtual Private Cloud (VPC) Tutorial ini mengacu pada VPC default, tetapi Anda dapat menggunakan VPC Anda sendiri. Jika menggunakan VPC Anda sendiri, pastikan bahwa ia memiliki subnet yang dipetakan ke setiap Availability Zone dari Region tempat Anda bekerja.

Minimal, Anda harus memiliki dua subnet publik yang tersedia untuk membuat penyeimbang beban. Anda juga harus memiliki dua subnet pribadi atau dua subnet publik untuk membuat grup Auto Scaling Anda dan mendaftarkannya ke penyeimbang beban.

Langkah 1: Siapkan templat peluncuran atau konfigurasi peluncuran

Gunakan template peluncuran atau konfigurasi peluncuran untuk tutorial ini.

Topik

- [Pilih atau buat template peluncuran](#)
- [Pilih atau buat konfigurasi peluncuran](#)

Pilih atau buat template peluncuran

Jika Anda sudah memiliki template peluncuran yang ingin Anda gunakan, pilih dengan menggunakan prosedur berikut.

Untuk memilih template peluncuran yang ada

1. Buka [halaman Peluncuran templat](#) konsol Amazon EC2.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah tempat penyeimbang beban dibuat.
3. Pilih template peluncuran.
4. Pilih Tindakan, Buat grup Auto Scaling.

Atau, untuk membuat template peluncuran baru, gunakan prosedur berikut.

Untuk membuat templat peluncuran

1. Buka [halaman Peluncuran templat](#) konsol Amazon EC2.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah tempat penyeimbang beban dibuat.
3. Pilih Buat templat peluncuran.
4. Masukkan nama dan berikan deskripsi untuk versi awal templat peluncuran.
5. Untuk Gambar Aplikasi dan OS (Amazon Machine Image), pilih ID AMI untuk instans Anda. Anda dapat mencari melalui semua AMI yang tersedia, atau memilih AMI dari daftar Terbaru atau Mulai Cepat. Jika Anda tidak melihat AMI yang Anda butuhkan, pilih Jelajahi AMI lainnya untuk menelusuri katalog AMI lengkap.

6. Untuk Jenis instance, pilih konfigurasi perangkat keras untuk instance Anda yang kompatibel dengan AMI yang Anda tentukan.
7. (Opsional) Untuk Pasangan kunci (login), pilih pasangan kunci untuk digunakan saat menghubungkan ke instance Anda.
8. Untuk pengaturan Jaringan, perluas Konfigurasi jaringan lanjutan dan lakukan hal berikut:
 - a. Pilih Tambahkan antarmuka jaringan untuk mengkonfigurasi antarmuka jaringan utama.
 - b. Untuk Auto-assign IP publik, tentukan apakah instans Anda menerima alamat IPv4 publik. Secara default, Amazon EC2 menetapkan alamat IPv4 publik jika instans EC2 diluncurkan ke subnet default atau jika instans diluncurkan ke subnet yang telah dikonfigurasi untuk secara otomatis menetapkan alamat IPv4 publik. Jika Anda tidak perlu terhubung ke instans Anda, Anda dapat memilih Nonaktifkan untuk mencegah instance di grup Anda menerima lalu lintas langsung dari internet. Dalam hal ini, instance-instance tersebut hanya akan menerima lalu lintas dari load balancer.
 - c. Untuk ID grup keamanan, menentukan grup keamanan untuk instance Anda dari VPC yang sama dengan load balancer.
 - d. Untuk Hapus saat penghentian, pilih Ya. Ini menghapus antarmuka jaringan saat grup Auto Scaling masuk, dan mengakhiri instance tempat antarmuka jaringan terpasang.
9. (Opsional) Untuk mendistribusikan kredensial secara aman ke instance Anda, untuk Detail tingkat lanjut, Profil instance IAM, masukkan Amazon Resource Name (ARN) peran IAM Anda.
10. (Opsional) Untuk menentukan data pengguna atau skrip konfigurasi untuk instance Anda, tempelkan ke Detail tingkat lanjut, Data pengguna.
11. Pilih Buat templat peluncuran.
12. Pada halaman konfirmasi, pilih Buat grup Auto Scaling.

Pilih atau buat konfigurasi peluncuran

Note

Kami sangat tidak menyarankan menggunakan konfigurasi peluncuran di aplikasi baru karena ini adalah fitur lama tanpa investasi yang direncanakan. Selain itu, akun baru yang dibuat pada atau setelah 1 Juni 2023 tidak akan memiliki opsi untuk membuat konfigurasi peluncuran baru melalui konsol. Untuk informasi selengkapnya, lihat [Konfigurasi peluncuran](#).

Untuk memilih konfigurasi peluncuran yang ada

1. Buka [halaman konfigurasi Luncurkan](#) konsol Amazon EC2.
2. Di bilah navigasi atas, pilih Wilayah tempat penyeimbang beban dibuat.
3. Pilih konfigurasi peluncuran.
4. Pilih Tindakan, Buat grup Auto Scaling.

Atau, untuk membuat konfigurasi peluncuran baru, gunakan prosedur berikut.

Untuk membuat konfigurasi peluncuran

1. Buka [halaman konfigurasi Luncurkan](#) konsol Amazon EC2. Saat diminta konfirmasi, pilih Lihat konfigurasi peluncuran untuk mengonfirmasi bahwa Anda ingin melihat halaman Konfigurasi peluncuran.
2. Di bilah navigasi atas, pilih Wilayah tempat penyeimbang beban dibuat.
3. Pilih Buat konfigurasi peluncuran, dan masukkan nama untuk konfigurasi peluncuran Anda.
4. Untuk Amazon machine image (AMI), masukkan ID AMI untuk instance Anda sebagai kriteria pencarian.
5. Untuk Jenis instance, pilih konfigurasi perangkat keras untuk instance Anda.
6. Pada Konfigurasi tambahan, perhatikan kolom-kolom berikut:
 - a. (Opsional) Untuk mendistribusikan kredensial secara aman ke instance EC2 Anda, untuk Profil instance IAM, pilih peran IAM Anda. Untuk informasi selengkapnya, lihat [Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2](#).
 - b. (Opsional) Untuk menentukan data pengguna atau skrip konfigurasi untuk instance Anda, tempelkan ke Detail tingkat lanjut, Data pengguna.
 - c. (Opsional) Untuk Detail tingkat lanjut, Jenis alamat IP, pertahankan nilai default. Saat Anda membuat grup Auto Scaling, Anda dapat menetapkan alamat IP publik untuk beberapa instance dalam grup Auto Scaling Anda dengan menggunakan subnet yang sudah mengaktifkan atribut alamat IP publik, seperti subnet default dalam VPC default. Atau, jika Anda tidak perlu terhubung ke instans Anda, Anda dapat memilih Jangan menetapkan alamat IP publik ke instans apa pun untuk mencegah instance di grup Anda menerima lalu lintas langsung dari internet. Dalam hal ini, instance-instance tersebut hanya akan menerima lalu lintas dari load balancer.

7. Untuk Grup keamanan, pilih grup keamanan yang ada dari VPC yang sama dengan load balancer. Jika Anda tetap memilih opsi Buat grup keamanan baru, aturan SSH default dikonfigurasi untuk instans Amazon EC2 yang menjalankan Linux. Aturan RDP default dikonfigurasi untuk instance Amazon EC2 yang menjalankan Windows.
8. Untuk Pasangan kunci (login), pilih opsi pada Opsi pasangan kunci.

Jika Anda sudah mengonfigurasi pasangan kunci instance Amazon EC2, Anda dapat memilihnya di sini.

Jika Anda belum memiliki pasangan kunci instance Amazon EC2, pilih Buat pasangan kunci baru dan berikan nama yang dapat dikenali. Pilih Unduh pasangan kunci untuk mengunduh pasangan kunci ke komputer Anda.

 Important

Jika Anda perlu terhubung ke instance, jangan pilih Lanjutkan tanpa pasangan kunci.

9. Centang kotak konfirmasi, lalu pilih Buat konfigurasi peluncuran.
10. Pilih kotak centang di samping nama konfigurasi peluncuran baru Anda dan pilih Tindakan, Buat grup Auto Scaling.

Langkah 2: Buat grup Auto Scaling

Gunakan prosedur berikut untuk melanjutkan di mana Anda tinggalkan setelah membuat atau memilih template peluncuran atau konfigurasi peluncuran Anda.

Untuk membuat grup Auto Scaling

1. Pada halaman Pilih templat peluncuran atau konfigurasi, untuk Nama grup Auto Scaling, masukkan nama untuk grup Auto Scaling Anda.
2. [Hanya templat peluncuran] Untuk template Launch, pilih apakah grup Auto Scaling menggunakan default, terbaru, atau versi spesifik dari template peluncuran saat melakukan scaling out.
3. Pilih Berikutnya.

Halaman opsi peluncuran instans Pilih muncul, memungkinkan Anda memilih pengaturan jaringan VPC yang ingin digunakan oleh grup Auto Scaling dan memberi Anda opsi untuk meluncurkan Instans Sesuai Permintaan dan Spot (jika Anda memilih templat peluncuran).

4. Di bagian Jaringan, untuk VPC, pilih VPC yang Anda gunakan untuk penyeimbang beban Anda. Jika Anda memilih VPC default, secara otomatis dikonfigurasi untuk menyediakan konektivitas internet untuk instance Anda. VPC ini mencakup subnet publik di setiap Zona Ketersediaan di Wilayah tersebut.
5. Untuk Availability Zones dan subnet, pilih satu atau beberapa subnet dari setiap Availability Zone yang ingin Anda sertakan, berdasarkan Availability Zone tempat penyeimbang beban berada. Untuk informasi selengkapnya, lihat [Pertimbangan saat memilih subnet VPC](#).
6. [Hanya meluncurkan templat] Di bagian Persyaratan tipe instans, gunakan pengaturan default untuk menyederhanakan langkah ini. (Jangan mengganti template peluncuran.) Untuk tutorial ini, Anda hanya akan meluncurkan Instans On-Demand menggunakan jenis instance yang ditentukan dalam template peluncuran Anda.
7. Pilih Berikutnya untuk pergi ke halaman Configure advanced options.
8. Untuk melampirkan grup ke penyeimbang beban yang ada, di bagian Load balancing, pilih Lampirkan ke penyeimbang beban yang ada. Anda dapat memilih Pilih dari grup target penyeimbang beban Anda atau Pilih dari Classic Load Balancers. Anda kemudian dapat memilih nama grup target untuk Application Load Balancer atau Network Load Balancer yang Anda buat, atau pilih nama Classic Load Balancer.
9. (Opsional) Untuk menggunakan pemeriksaan kesehatan Elastic Load Balancing, untuk Pemeriksaan kesehatan, pilih ELB pada Jenis pemeriksaan kesehatan.
10. Setelah selesai mengonfigurasi grup Auto Scaling, pilih Lewati untuk meninjau.
11. Pada halaman Peninjauan, tinjau detail grup Auto Scaling Anda. Anda dapat memilih Edit untuk melakukan perubahan. Setelah selesai, pilih Buat grup Auto Scaling.

Setelah Anda membuat grup Auto Scaling dengan load balancer terlampir, load balancer secara otomatis mendaftarkan instance baru saat online. Anda hanya memiliki satu instance pada titik ini, jadi tidak banyak yang harus didaftarkan. Namun, Anda dapat menambahkan instance tambahan dengan memperbarui kapasitas grup yang diinginkan. Untuk step-by-step instruksi, lihat [Ubah kapasitas yang diinginkan dari grup Auto Scaling Anda](#).

Langkah 3: Verifikasi bahwa penyeimbang beban Anda terpasang

Untuk memverifikasi bahwa penyeimbang beban Anda terpasang

1. Dari [halaman grup Auto Scaling](#) konsol Amazon EC2, pilih kotak centang di sebelah grup Auto Scaling Anda.

2. Pada tab Detail Anda, Load balancing menampilkan grup target load balancer yang ditambahkan atau Classic Load Balancer.
3. Pada tab Aktivitas, dalam riwayat Aktivitas, Anda dapat memverifikasi bahwa instans berhasil diluncurkan. Kolom Status menunjukkan apakah grup Auto Scaling Anda telah berhasil meluncurkan instance. Jika instans Anda gagal diluncurkan, Anda dapat menemukan ide pemecahan masalah untuk masalah peluncuran instans umum di [Memecahkan Masalah Auto Scaling Amazon EC2](#)
4. Pada tab Manajemen instans, di bawah Instans, Anda dapat memverifikasi bahwa instans Anda siap menerima lalu lintas. Pada awalnya, instans Anda akan berada pada status Pending. Setelah instans siap untuk menerima lalu lintas, statusnya adalah InService. Kolom status Kesehatan menunjukkan hasil pemeriksaan kesehatan Auto Scaling Amazon EC2 pada instans Anda. Meskipun suatu instance dapat ditandai sebagai sehat, load balancer hanya akan mengirim lalu lintas untuk instance yang melewati pemeriksaan kesehatan load balancer.
5. Verifikasikan bahwa instance Anda terdaftar dengan load balancer. Buka [halaman Grup target](#) konsol Amazon EC2. Pilih grup target Anda, kemudian pilih tab Target. Jika keadaan kasus `Andainitial`, itu mungkin karena mereka masih dalam proses terdaftar, atau mereka masih menjalani pemeriksaan kesehatan. Saat instans Anda berada dalam tahap `healthy`, berarti instans sudah siap digunakan.

Langkah 4: Langkah selanjutnya

Setelah menyelesaikan tutorial ini, Anda dapat mempelajari lebih lanjut:

- Auto Scaling Amazon EC2 menentukan apakah instans sehat berdasarkan status pemeriksaan kesehatan yang digunakan grup Auto Scaling Anda. Jika Anda mengaktifkan pemeriksaan kesehatan penyeimbang beban dan instans gagal dalam pemeriksaan kesehatan, grup Auto Scaling Anda menganggap instans tersebut tidak sehat dan menggantikannya. Untuk informasi selengkapnya, lihat [Pemeriksaan kondisi](#).
- Anda dapat memperluas aplikasi Anda ke Availability Zone tambahan di Wilayah yang sama untuk meningkatkan toleransi kesalahan jika ada gangguan layanan. Untuk informasi selengkapnya, lihat [Tambahkan Availability Zone](#).
- Anda dapat mengonfigurasi grup Auto Scaling untuk menggunakan kebijakan penskalaan pelacakan target. Ini secara otomatis menambah atau mengurangi jumlah instans saat permintaan pada instans Anda berubah. Ini memungkinkan grup menangani perubahan dalam jumlah lalu lintas yang diterima aplikasi Anda. Untuk informasi selengkapnya, lihat [Kebijakan penskalaan pelacakan target](#).

Langkah 5: Bersihkan

Setelah Anda selesai dengan sumber daya yang Anda buat untuk tutorial ini, Anda harus mempertimbangkan untuk membersihkannya untuk menghindari biaya yang tidak perlu.

Untuk menghapus grup Auto Scaling Anda

1. Buka [halaman grup Auto Scaling](#) di konsol Amazon EC2.
2. Pilih kotak centang di samping grup Auto Scaling Anda.
3. Pilih Hapus.
4. Saat diminta konfirmasi, ketik **delete** untuk mengonfirmasi penghapusan grup Auto Scaling yang ditentukan, lalu pilih Hapus.

Ikon memuat di kolom Nama menunjukkan bahwa grup Auto Scaling dihapus. Saat penghapusan telah terjadi, kolom Diinginkan, Min, dan Maks menunjukkan instans 0 untuk grup Auto Scaling. Hanya perlu beberapa menit untuk menghentikan proses dan menghapus grup. Refresh daftar untuk melihat status saat ini.

Lewati prosedur berikut jika Anda ingin menyimpan templat peluncuran Anda.

Untuk menghapus templat peluncuran Anda

1. Buka [halaman Peluncuran templat](#) konsol Amazon EC2.
2. Pilih templat peluncuran Anda.
3. Pilih Tindakan, Hapus templat.
4. Saat diminta konfirmasi, ketik **Delete** untuk mengonfirmasi penghapusan templat peluncuran yang ditentukan, lalu pilih Hapus.

Lewati prosedur berikut jika Anda ingin menyimpan konfigurasi peluncuran Anda.

Untuk menghapus konfigurasi peluncuran Anda

1. Buka [halaman konfigurasi Luncurkan](#) konsol Amazon EC2.
2. Pilih konfigurasi peluncuran Anda.
3. Pilih Tindakan, Hapus konfigurasi peluncuran.
4. Saat diminta konfirmasi, pilih Hapus.

Lewati prosedur berikut jika Anda ingin menyimpan penyeimbang beban untuk penggunaan di masa mendatang.

Untuk menghapus penyeimbang beban Anda

1. Buka [halaman Load balancers](#) dari konsol Amazon EC2.
2. Pilih load balancer dan pilih Tindakan, Hapus.
3. Ketika diminta konfirmasi, pilih Ya, Hapus.

Untuk menghapus grup target Anda

1. Buka [halaman Grup target](#) konsol Amazon EC2.
2. Pilih grup target dan pilih Tindakan, Hapus.
3. Ketika diminta untuk mengonfirmasi, pilih Ya, Hapus.

Sumber daya terkait

Dengan AWS CloudFormation, Anda dapat membuat dan menyediakan penyebaran AWS infrastruktur yang dapat diprediksi dan berulang kali, dengan menggunakan file template untuk membuat dan menghapus kumpulan sumber daya bersama-sama sebagai satu unit (tumpukan). Untuk informasi selengkapnya, silakan lihat [Panduan Pengguna AWS CloudFormation](#).

Untuk panduan yang menggunakan template tumpukan untuk menyediakan grup Auto Scaling dan Application Load Balancer, [lihat Panduan: Membuat aplikasi yang diskalakan dan diseimbangkan beban di Panduan Pengguna](#).AWS CloudFormation Gunakan templat panduan dan contoh sebagai titik awal untuk membuat templat serupa untuk memenuhi kebutuhan Anda.

Templat peluncuran

Templat peluncuran serupa dengan [konfigurasi peluncuran](#), dalam hal menentukan informasi konfigurasi instans. Ini mencakup ID Amazon Machine Image (AMI), jenis instans, key pair, grup keamanan, dan parameter lain yang digunakan untuk meluncurkan instans EC2. Namun, mendefinisikan template peluncuran alih-alih konfigurasi peluncuran memungkinkan Anda memiliki beberapa versi template peluncuran.

Dengan pembuatan versi template peluncuran, Anda dapat membuat subset dari set parameter lengkap. Kemudian, Anda dapat menggunakannya kembali untuk membuat versi lain dari template peluncuran yang sama. Misalnya, Anda dapat membuat template peluncuran yang mendefinisikan konfigurasi dasar tanpa AMI atau skrip data pengguna. Setelah Anda membuat template peluncuran, Anda dapat membuat versi baru dan menambahkan AMI dan data pengguna yang memiliki versi terbaru dari aplikasi Anda untuk pengujian. Ini menghasilkan dua versi template peluncuran. Menyimpan konfigurasi dasar membantu Anda mempertahankan parameter konfigurasi umum yang diperlukan. Anda dapat membuat versi baru template peluncuran Anda dari konfigurasi dasar kapan pun Anda mau. Anda juga dapat menghapus versi yang digunakan untuk menguji aplikasi Anda ketika Anda tidak lagi membutuhkannya.

Kami menyarankan Anda menggunakan template peluncuran untuk memastikan bahwa Anda mengakses fitur dan peningkatan terbaru. Tidak semua fitur Auto Scaling Amazon EC2 tersedia saat Anda menggunakan konfigurasi peluncuran. Misalnya, Anda tidak dapat membuat grup Auto Scaling yang meluncurkan Instans Spot dan On-Demand atau yang menentukan beberapa jenis instans. Anda harus menggunakan templat peluncuran untuk mengonfigurasi fitur ini. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).

Dengan template peluncuran, Anda juga dapat menggunakan fitur Amazon EC2 yang lebih baru. Ini termasuk parameter Systems Manager (AMI ID), generasi terbaru dari volume IOPS Provisioned EBS (io2), penandaan volume EBS, instans T2 Unlimited, Reservasi Kapasitas, Blok Kapasitas, dan Host Khusus, untuk beberapa nama.

Saat Anda membuat template peluncuran, semua parameter bersifat opsional. Namun, jika template peluncuran tidak menentukan AMI, Anda tidak dapat menambahkan AMI saat membuat grup Auto Scaling. Jika Anda menentukan AMI tetapi tidak ada tipe instans, Anda dapat menambahkan satu atau beberapa jenis instance saat membuat grup Auto Scaling.

Daftar Isi

- [Izin untuk bekerja dengan template peluncuran](#)

- [Operasi API yang didukung oleh template peluncuran](#)
- [Membuat template peluncuran untuk grup Auto Scaling](#)
- [Buat template peluncuran menggunakan pengaturan lanjutan](#)
- [Migrasi grup Auto Scaling Anda untuk meluncurkan template](#)
- [Migrasikan AWS CloudFormation tumpukan untuk meluncurkan templat](#)
- [Contoh untuk membuat dan mengelola template peluncuran dengan AWS Command Line Interface \(AWS CLI\)](#)
- [Gunakan AWS Systems Manager parameter alih-alih ID AMI di templat peluncuran](#)

Izin untuk bekerja dengan template peluncuran

Prosedur di bagian ini mengasumsikan bahwa Anda sudah memiliki izin yang diperlukan untuk membuat templat peluncuran. Untuk informasi tentang cara administrator memberi Anda izin, lihat [Mengontrol akses untuk meluncurkan templat dengan izin IAM di Panduan Pengguna Amazon EC2 untuk Instans Linux](#).

Perhatikan bahwa jika Anda tidak memiliki izin yang cukup untuk menggunakan dan membuat sumber daya yang ditentukan dalam templat peluncuran, Anda menerima kesalahan bahwa Anda tidak diizinkan untuk menggunakan templat peluncuran saat Anda mencoba menentukannya untuk grup Auto Scaling. Untuk informasi selengkapnya, lihat [Memecahkan Masalah Auto Scaling Amazon EC2: Luncurkan templat](#).

Untuk contoh kebijakan IAM yang memungkinkan Anda memanggil `CreateAutoScalingGroup`, `UpdateAutoScalingGroup`, dan operasi `RunInstances` API dengan template peluncuran, lihat [Luncurkan dukungan templat](#).

Operasi API yang didukung oleh template peluncuran

Untuk daftar operasi API yang didukung oleh template peluncuran, lihat [tindakan Amazon EC2](#) di Referensi API [Amazon EC2](#).

Membuat template peluncuran untuk grup Auto Scaling

Sebelum Anda dapat membuat grup Auto Scaling menggunakan template peluncuran, Anda harus membuat template peluncuran yang berisi informasi konfigurasi untuk meluncurkan instance, termasuk ID Amazon Machine Image (AMI).

Untuk membuat templat peluncuran baru, gunakan prosedur berikut.

Daftar Isi

- [Buat template peluncuran Anda \(konsol\)](#)
- [Ubah pengaturan antarmuka jaringan default \(konsol\)](#)
- [Ubah konfigurasi penyimpanan \(konsol\)](#)
- [Buat template peluncuran dari instance yang ada \(konsol\)](#)
- [Sumber daya terkait](#)
- [Batasan](#)

Important

Parameter template peluncuran tidak sepenuhnya divalidasi saat Anda membuat template peluncuran. Jika Anda menentukan nilai yang salah untuk parameter, atau jika Anda tidak menggunakan kombinasi parameter yang didukung, tidak ada instance yang dapat diluncurkan menggunakan template peluncuran ini. Pastikan untuk menentukan nilai yang benar untuk parameter dan gunakan kombinasi parameter yang didukung. Misalnya, untuk meluncurkan instance dengan AMI Graviton AWS atau Graviton2 berbasis ARM, Anda harus menentukan jenis instans yang kompatibel dengan ARM. Untuk informasi selengkapnya, lihat [Meluncurkan pembatasan templat](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Buat template peluncuran Anda (konsol)

Langkah-langkah berikut menjelaskan cara mengkonfigurasi template peluncuran dasar:

- Tentukan Amazon Machine Image (AMI) dari mana untuk meluncurkan instance.
- Pilih jenis instans yang kompatibel dengan AMI yang Anda tentukan.
- Tentukan pasangan kunci yang akan digunakan saat menghubungkan ke instans, misalnya menggunakan SSH.
- Tambahkan satu atau beberapa grup keamanan untuk memungkinkan akses jaringan ke instance.
- Tentukan apakah akan melampirkan volume tambahan ke setiap instance.
- Tambahkan tag khusus (pasangan nilai-kunci) ke instans dan volume.

Untuk membuat templat peluncuran

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pada panel navigasi, di bawah Instans, pilih Luncurkan Template.
3. Pilih Buat templat peluncuran. Masukkan nama dan berikan deskripsi untuk versi awal templat peluncuran.
4. (Opsional) Di bawah panduan Auto Scaling, pilih kotak centang agar Amazon EC2 memberikan panduan untuk membantu membuat template yang akan digunakan dengan Amazon EC2 Auto Scaling.
5. Di bawah Meluncurkan konten template, isi setiap bidang wajib dan kolom opsional sesuai kebutuhan.
 - a. Gambar Aplikasi dan OS (Gambar Mesin Amazon): (Wajib) Pilih ID AMI untuk instans Anda. Anda dapat mencari melalui semua AMI yang tersedia, atau memilih AMI dari daftar Terbaru atau Mulai Cepat. Jika Anda tidak melihat AMI yang Anda butuhkan, pilih Jelajahi AMI lainnya untuk menelusuri katalog AMI lengkap.

Untuk memilih AMI kustom, Anda harus terlebih dahulu membuat AMI Anda dari instance yang disesuaikan. Untuk informasi selengkapnya, lihat [Membuat AMI](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

- b. Untuk tipe Instance, pilih satu jenis instans yang kompatibel dengan AMI yang Anda tentukan.

Atau, untuk menggunakan pemilihan tipe instans berbasis atribut, pilih Advanced, Tentukan atribut tipe instance, lalu tentukan opsi berikut:

- Jumlah vCPU: Masukkan jumlah minimum dan maksimum vCPU. Untuk menunjukkan tidak ada batas, masukkan minimum 0, dan biarkan kosong maksimum.
- Jumlah memori (MiB): Masukkan jumlah memori minimum dan maksimum, di MiB. Untuk menunjukkan tidak ada batas, masukkan minimum 0, dan biarkan kosong maksimum.
- Perluas atribut tipe instans opsional dan pilih Tambahkan atribut untuk membatasi lebih lanjut jenis instance yang dapat digunakan untuk memenuhi kapasitas yang Anda inginkan. Untuk informasi tentang setiap atribut, lihat [InstanceRequirementsRequest](#) di Referensi API Amazon EC2.
- Jenis instans yang dihasilkan: Anda dapat melihat jenis instance yang cocok dengan persyaratan komputasi yang ditentukan, seperti vCPU, memori, dan penyimpanan.

- Untuk mengecualikan jenis instance, pilih Tambahkan atribut. Dari daftar Atribut, pilih Jenis instans yang dikecualikan. Dari daftar Nilai atribut, pilih tipe instans yang akan dikecualikan.
- c. Pasangan kunci (login): Untuk nama Key pair, pilih key pair yang ada, atau pilih Create new key pair untuk membuat yang baru. Untuk informasi selengkapnya, lihat [pasangan kunci Amazon EC2 dan instans Linux](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- d. Pengaturan jaringan: Untuk Firewall (grup keamanan), gunakan satu atau beberapa grup keamanan, atau kosongkan ini dan konfigurasi satu atau beberapa grup keamanan sebagai bagian dari antarmuka jaringan. Untuk informasi selengkapnya, lihat [Grup Keamanan Amazon EC2 untuk instans Linux](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Jika Anda tidak menentukan grup keamanan apa pun di template peluncuran, Amazon EC2 menggunakan grup keamanan default untuk VPC tempat grup Auto Scaling Anda akan meluncurkan instans. Secara default, grup keamanan ini tidak mengizinkan lalu lintas masuk dari jaringan eksternal. Untuk informasi selengkapnya, lihat [Grup keamanan default untuk VPC Anda](#) di Panduan Pengguna Amazon VPC.

- e. Lakukan salah satu hal berikut:
 - Ubah pengaturan antarmuka jaringan default. Misalnya, Anda dapat mengaktifkan atau menonaktifkan fitur pengalamatan IPv4 publik, yang mengesampingkan pengaturan alamat IPv4 publik yang menetapkan otomatis di subnet. Untuk informasi selengkapnya, lihat [Ubah pengaturan antarmuka jaringan default \(konsol\)](#).
 - Lewati langkah ini untuk menjaga pengaturan antarmuka jaringan default.
 - f. Lakukan salah satu hal berikut:
 - Ubah konfigurasi penyimpanan. Untuk informasi selengkapnya, lihat [Ubah konfigurasi penyimpanan \(konsol\)](#).
 - Lewati langkah ini untuk menjaga konfigurasi penyimpanan default.
 - g. Untuk tag Sumber Daya, tentukan tag dengan memberikan kombinasi kunci dan nilai. Jika Anda menentukan tag instance di template peluncuran dan kemudian Anda memilih untuk menyebarkan tag grup Auto Scaling Anda ke instance-nya, semua tag digabungkan. Jika kunci tag yang sama ditentukan untuk tag di template peluncuran dan tag di grup Auto Scaling Anda, maka nilai tag dari grup akan diutamakan.
6. (Opsional) Konfigurasi pengaturan lanjutan. Misalnya, Anda dapat memilih peran IAM yang dapat digunakan aplikasi Anda saat mengakses AWS sumber daya lain atau menentukan

data pengguna instance yang dapat digunakan untuk melakukan tugas konfigurasi otomatis umum setelah instance dimulai. Untuk informasi selengkapnya, lihat [Buat template peluncuran menggunakan pengaturan lanjutan](#).

7. Ketika Anda siap untuk membuat template peluncuran, pilih Buat template peluncuran.
8. Untuk membuat grup Auto Scaling, pilih Buat grup Auto Scaling dari halaman konfirmasi.

Ubah pengaturan antarmuka jaringan default (konsol)

Antarmuka jaringan menyediakan konektivitas ke sumber daya lain di VPC dan internet Anda. Untuk informasi selengkapnya, lihat [Menyediakan konektivitas jaringan untuk instans Auto Scaling menggunakan Amazon VPC](#).

Bagian ini menunjukkan cara mengubah pengaturan antarmuka jaringan default. Misalnya, Anda dapat menentukan apakah Anda ingin menetapkan alamat IPv4 publik untuk setiap instance, bukan default ke pengaturan alamat IPv4 publik yang ditetapkan secara otomatis di subnet.

Pertimbangan dan batasan

Saat mengubah pengaturan antarmuka jaringan default, ingatlah pertimbangan dan batasan berikut:

- Anda harus mengonfigurasi grup keamanan sebagai bagian dari antarmuka jaringan, bukan di bagian Grup keamanan templat. Anda tidak dapat menentukan grup keamanan di kedua tempat.
- Anda tidak dapat menetapkan alamat IP pribadi sekunder, yang dikenal sebagai alamat IP sekunder, ke antarmuka jaringan.
- Jika Anda menentukan ID antarmuka jaringan yang ada, Anda hanya dapat meluncurkan satu instance. Untuk melakukan ini, Anda harus menggunakan AWS CLI atau SDK untuk membuat grup Auto Scaling. Saat membuat grup, Anda harus menentukan Availability Zone, tetapi bukan ID subnet. Selain itu, Anda dapat menentukan antarmuka jaringan yang ada hanya jika memiliki indeks perangkat 0.
- Anda tidak dapat menetapkan alamat IPv4 publik secara otomatis jika Anda menentukan lebih dari satu antarmuka jaringan. Anda juga tidak dapat menentukan indeks perangkat duplikat di seluruh antarmuka jaringan. Kedua antarmuka jaringan primer dan sekunder berada di subnet yang sama.
- Ketika sebuah instance diluncurkan, alamat pribadi secara otomatis dialokasikan ke setiap antarmuka jaringan. Alamat berasal dari kisaran CIDR subnet di mana instance diluncurkan. [Untuk informasi tentang menentukan blok CIDR \(atau rentang alamat IP\) untuk VPC atau subnet Anda, lihat Panduan Pengguna Amazon VPC](#).

Untuk mengubah pengaturan antarmuka jaringan default

1. Di bawah Pengaturan jaringan, perluas Konfigurasi jaringan lanjutan.
2. Pilih Tambahkan antarmuka jaringan untuk mengkonfigurasi antarmuka jaringan utama, memperhatikan bidang-bidang berikut:
 - a. Indeks perangkat: Pertahankan nilai default, 0, untuk menerapkan perubahan Anda ke antarmuka jaringan utama (eth0).
 - b. Antarmuka jaringan: Pertahankan nilai default, Antarmuka baru, agar Amazon EC2 Auto Scaling secara otomatis membuat antarmuka jaringan baru saat instance diluncurkan. Atau, Anda dapat memilih antarmuka jaringan yang ada dan tersedia dengan indeks perangkat 0, tetapi ini membatasi grup Auto Scaling Anda ke satu instance.
 - c. Deskripsi: (Opsional) Masukkan nama deskriptif.
 - d. Subnet: Pertahankan default Jangan sertakan dalam pengaturan template peluncuran.

Jika AMI menentukan subnet untuk antarmuka jaringan, ini menghasilkan kesalahan. Sebaiknya nonaktifkan panduan Auto Scaling sebagai solusinya. Setelah Anda melakukan perubahan ini, Anda tidak akan menerima pesan kesalahan. Namun, terlepas dari di mana subnet ditentukan, pengaturan subnet dari grup Auto Scaling diutamakan dan tidak dapat diganti.

- e. Tetapkan IP publik secara otomatis: Ubah apakah antarmuka jaringan Anda dengan indeks perangkat 0 menerima alamat IPv4 publik. Secara default, instance di subnet default menerima alamat IPv4 publik, sedangkan instance di subnet nondefault tidak. Pilih Aktifkan atau Nonaktifkan untuk mengganti pengaturan default subnet.
- f. Grup keamanan: Pilih satu atau beberapa grup keamanan untuk antarmuka jaringan. Setiap grup keamanan harus dikonfigurasi untuk VPC yang akan digunakan dalam grup Auto Scaling Anda. Untuk informasi selengkapnya, lihat [Grup Keamanan Amazon EC2 untuk instans Linux](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- g. Hapus saat penghentian: Pilih Ya untuk menghapus antarmuka jaringan saat instance dihentikan, atau pilih Tidak untuk menjaga antarmuka jaringan.
- h. Adaptor Kain Elastis: Untuk mendukung komputasi kinerja tinggi dan kasus penggunaan pembelajaran mesin, ubah antarmuka jaringan menjadi antarmuka jaringan Adaptor Kain Elastis. Untuk informasi selengkapnya, lihat [Adaptor Kain Elastis](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- i. Indeks kartu jaringan: Pilih 0 untuk melampirkan antarmuka jaringan utama ke kartu jaringan dengan indeks perangkat 0. Jika opsi ini tidak tersedia, pertahankan nilai default, Jangan

sertakan dalam template peluncuran. Melampirkan antarmuka jaringan ke kartu jaringan tertentu hanya tersedia untuk jenis instans yang didukung. Untuk informasi selengkapnya, lihat [Kartu jaringan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

- j. ENA Express: Misalnya jenis yang mendukung ENA Express, pilih Aktifkan untuk mengaktifkan ENA Express atau Nonaktifkan untuk menonaktifkannya. Untuk informasi selengkapnya, lihat [Meningkatkan performa jaringan dengan ENA Express pada instans Linux](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
 - k. ENA Express UDP: Jika Anda mengaktifkan ENA Express, Anda dapat menggunakannya secara opsional untuk lalu lintas UDP. Pilih Aktifkan untuk mengaktifkan ENA Express UDP atau Nonaktifkan untuk menonaktifkannya.
3. Untuk menambahkan antarmuka jaringan sekunder, pilih Tambahkan antarmuka jaringan.

Ubah konfigurasi penyimpanan (konsol)

Anda dapat mengubah konfigurasi penyimpanan untuk instans yang diluncurkan dari AMI yang didukung Amazon EBS-backed atau AMI yang didukung oleh instance store-backed. Anda juga dapat menentukan volume EBS tambahan untuk dilampirkan ke instance. AMI mencakup satu atau lebih volume penyimpanan, termasuk volume root (Volume 1 (AMI Root)).

Untuk memodifikasi konfigurasi penyimpanan

1. Di Konfigurasi penyimpanan, ubah ukuran atau jenis volume.

Jika nilai yang Anda tentukan untuk ukuran volume berada di luar batas jenis volume, atau lebih kecil dari ukuran snapshot, pesan kesalahan akan ditampilkan. Untuk membantu Anda mengatasi masalah ini, pesan ini memberikan nilai minimum atau maksimum yang dapat diterima bidang.

Hanya volume yang terkait dengan AMI yang didukung Amazon EBS-backed yang muncul. Untuk menampilkan informasi tentang konfigurasi penyimpanan instance yang diluncurkan dari instance instance store-backed AMI, pilih Tampilkan detail dari bagian Volume penyimpanan instans.

Untuk menentukan semua parameter volume EBS, beralih ke tampilan Advanced di sudut kanan atas.

2. Untuk opsi lanjutan, perluas volume yang ingin Anda ubah dan konfigurasi volume sebagai berikut:

- a. Tipe penyimpanan: Tipe volume (EBS atau sementara) yang akan dikaitkan dengan instans Anda. Tipe volume penyimpanan instans (sementara) hanya tersedia jika Anda memilih tipe instans yang mendukungnya. Untuk informasi selengkapnya, lihat [volume Amazon EBS](#) di Panduan Pengguna Amazon EBS dan penyimpanan [instans Amazon EC2](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- b. Nama perangkat: Pilih dari daftar nama perangkat yang tersedia untuk volume.
- c. Snapshot: Pilih snapshot yang akan digunakan untuk membuat volume. Anda dapat mencari snapshot bersama dan publik yang tersedia dengan memasukkan teks ke dalam bidang Snapshot.
- d. Ukuran (GiB): Untuk volume EBS, Anda dapat menentukan ukuran penyimpanan. Jika Anda telah memilih AMI dan instans yang memenuhi syarat untuk tingkat gratis, ingatlah bahwa agar tetap dalam tingkat gratis, Anda harus tetap di bawah 30 GiB dari total penyimpanan. Untuk informasi selengkapnya, [lihat Batasan pada ukuran dan konfigurasi volume EBS di Panduan Pengguna Amazon EBS](#).
- e. Jenis volume: Untuk volume EBS, pilih jenis volume. Untuk informasi selengkapnya, lihat [Jenis volume Amazon EBS](#) di Panduan Pengguna Amazon EBS.
- f. IOPS: Jika Anda telah memilih tipe SSD IOPS yang tersedia (`io1` dan `io2`) atau SSD Tujuan Umum (`gp3`), Anda dapat memasukkan jumlah operasi I/O per detik (IOPS) yang dapat didukung volume. Ini diperlukan untuk volume `io1`, `io2`, dan `gp3`. Ini tidak didukung untuk `gp2`, `st1`, `sc1`, atau volume standar.
- g. Hapus saat penghentian: Untuk volume EBS, pilih Ya untuk menghapus volume saat instance dihentikan, atau pilih Tidak untuk menyimpan volume.
- h. Terenkripsi: Jika tipe instans mendukung enkripsi EBS, Anda dapat memilih Ya untuk mengaktifkan enkripsi untuk volume tersebut. Jika Anda telah mengaktifkan enkripsi secara default di Wilayah ini, enkripsi diaktifkan untuk Anda. Untuk informasi selengkapnya, lihat [enkripsi Amazon EBS](#) dan [Aktifkan enkripsi secara default](#) di Panduan Pengguna Amazon EBS.

Efek default pengaturan parameter ini bervariasi dengan pilihan sumber volume, seperti yang dijelaskan dalam tabel berikut. Dalam semua kasus, Anda harus memiliki izin untuk menggunakan yang ditentukan AWS KMS key.

Hasil enkripsi

Jika parameter Encrypted diatur ke...	Dan jika sumber volume adalah...	Kemudian status enkripsi default adalah...	Catatan
Tidak	Volume (kosong) baru	Tidak terenkripsi*	N/A
	Snapshot tidak dienkripsi yang Anda miliki	Tidak terenkripsi*	
	Snapshot dienkripsi yang Anda miliki	Dienkripsi dengan kunci yang sama	
	Snapshot tidak terenkripsi yang dibagikan dengan Anda	Tidak terenkripsi*	
	Snapshot terenkripsi yang dibagikan dengan Anda	Dienkripsi dengan kunci KMS default	
Ya	Volume baru	Dienkripsi dengan kunci KMS default	Untuk menggunakan kunci KMS non-default, tentukan nilai untuk parameter kunci KMS.
	Snapshot tidak terenkripsi yang Anda miliki	Dienkripsi dengan kunci KMS default	
	Snapshot terenkripsi yang Anda miliki	Dienkripsi dengan kunci yang sama	
	Snapshot yang tidak terenkripsi yang dibagikan dengan Anda	Dienkripsi dengan kunci KMS default	
	Snapshot terenkripsi yang dibagikan dengan Anda	Dienkripsi dengan kunci KMS default	

- * Jika enkripsi secara default diaktifkan, semua volume yang baru dibuat (apakah parameter Terenkripsi diatur ke Ya atau tidak) dienkripsi menggunakan kunci KMS default. Jika Anda mengatur parameter kunci Terenkripsi dan KMS, maka Anda dapat menentukan kunci KMS non-default.
- i. Kunci KMS: Jika Anda memilih Ya untuk Terenkripsi, maka Anda harus memilih kunci yang dikelola pelanggan untuk digunakan untuk mengenkripsi volume. Jika Anda telah mengaktifkan enkripsi secara default di Wilayah ini, kunci yang dikelola pelanggan secara default akan dipilihkan untuk Anda. Anda dapat memilih kunci yang berbeda atau menentukan ARN dari setiap kunci yang dikelola pelanggan yang sebelumnya Anda buat menggunakan AWS Key Management Service
3. Untuk menentukan volume tambahan yang akan dilampirkan ke instance yang diluncurkan oleh template peluncuran ini, pilih Tambahkan volume baru.

Buat template peluncuran dari instance yang ada (konsol)

Untuk membuat templat peluncuran dari instans yang sudah ada

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pada panel navigasi, di bawah Instans, pilih Instans.
3. Pilih instance dan pilih Actions, Image and templates, Create template from instance.
4. Berikan nama dan deskripsi.
5. Di bawah panduan Auto Scaling, pilih kotak centang.
6. Sesuaikan pengaturan apa pun sesuai kebutuhan, dan pilih Buat templat peluncuran.
7. Untuk membuat grup Auto Scaling, pilih Buat grup Auto Scaling dari halaman konfirmasi.

Sumber daya terkait

Kami menyediakan beberapa cuplikan template JSON dan YAMAL yang dapat Anda gunakan untuk memahami cara mendeklarasikan template peluncuran di template tumpukan Anda. AWS CloudFormation Untuk informasi selengkapnya, lihat [AWS::EC2::LaunchTemplate](#) dan [Buat templat peluncuran dengan AWS CloudFormation](#) bagian Panduan AWS CloudFormation Pengguna.

Untuk informasi selengkapnya tentang template peluncuran, lihat [Meluncurkan instance dari template peluncuran](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Batasan

- Meskipun Anda dapat menentukan subnet dalam template peluncuran, hal itu tidak diperlukan jika Anda hanya menggunakan template peluncuran untuk membuat grup Auto Scaling. Anda tidak dapat menentukan subnet untuk grup Auto Scaling dengan menentukan subnet dalam template peluncuran. Subnet untuk grup Auto Scaling diambil dari definisi sumber daya grup Auto Scaling sendiri.
- Untuk batasan lain pada antarmuka jaringan yang ditentukan pengguna, lihat. [Ubah pengaturan antarmuka jaringan default \(konsol\)](#)

Buat template peluncuran menggunakan pengaturan lanjutan

Topik ini menjelaskan cara membuat template peluncuran dengan pengaturan lanjutan dari AWS Management Console.

Untuk membuat template peluncuran menggunakan pengaturan lanjutan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pada panel navigasi, di bawah Instans, pilih Luncurkan Template, lalu pilih Buat template peluncuran.
3. Konfigurasi template peluncuran Anda seperti yang dijelaskan dalam topik berikut:
 - [Pengaturan yang diperlukan](#)
 - [Pengaturan lanjutan](#)
4. Pilih Buat templat peluncuran.

Pengaturan yang diperlukan

Saat Anda membuat template peluncuran, Anda harus menyertakan pengaturan yang diperlukan berikut.

Luncurkan nama template

Masukkan nama unik yang menggambarkan template peluncuran.

Aplikasi dan Gambar OS (Gambar Mesin Amazon)

Pilih Amazon Machine Image (AMI) yang ingin Anda gunakan. Anda dapat mencari atau menelusuri AMI yang ingin Anda gunakan. Untuk efisiensi penskalaan terbaik, pilih AMI kustom yang sepenuhnya dikonfigurasi untuk meluncurkan instance dengan kode aplikasi Anda dan memerlukan sedikit modifikasi saat peluncuran.

Jenis instans

Pilih jenis instans yang kompatibel dengan AMI Anda. Anda dapat melewati penambahan jenis instans ke template peluncuran jika Anda berencana untuk menggunakan beberapa jenis instance yang disematkan dalam definisi sumber daya grup Auto Scaling sendiri. Jenis instans hanya diperlukan jika Anda tidak berencana untuk membuat [grup instance campuran](#).

Pengaturan lanjutan

Pengaturan lanjutan bersifat opsional. Jika Anda tidak mengonfigurasi pengaturan lanjutan apa pun, kemampuan spesifik tidak akan ditambahkan ke instance Anda.

Perluas bagian Detail lanjutan untuk melihat pengaturan lanjutan. Bagian berikut menjelaskan pengaturan lanjutan yang paling berguna untuk difokuskan saat membuat template peluncuran untuk grup Auto Scaling. Untuk informasi selengkapnya, lihat [Detail lanjutan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Profil instans IAM

Profil instance berisi peran IAM yang ingin Anda gunakan. Saat grup Auto Scaling Anda meluncurkan instans EC2, izin yang ditentukan dalam peran IAM terkait akan diberikan kepada aplikasi yang berjalan pada instance. Untuk informasi selengkapnya, lihat [Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2](#).

Perlindungan pengakhiran

Saat diaktifkan, fitur ini mencegah pengguna menghentikan instance menggunakan konsol Amazon EC2, perintah CLI, dan operasi API. Perlindungan penghentian memberikan perlindungan ekstra terhadap penghentian yang tidak disengaja. Ini tidak mencegah Auto Scaling Amazon EC2 menghentikan instance. Untuk mengontrol instans mana yang dapat dihentikan Auto Scaling Amazon EC2, lihat [Gunakan perlindungan skala dalam instance](#)

CloudWatch Pemantauan terperinci

Anda dapat mengaktifkan pemantauan terperinci untuk instans EC2 agar instans EC2 dapat mengirim data metrik ke Amazon dengan CloudWatch interval 1 menit. Secara default, instans EC2 mengirim data metrik ke CloudWatch interval 5 menit. Biaya tambahan berlaku. Untuk informasi selengkapnya, lihat [Konfigurasi pemantauan untuk instans Auto Scaling](#).

Spesifikasi kredit

Amazon EC2 menyediakan instans kinerja burstable, seperti T2, T3, dan T3a, yang memungkinkan aplikasi meledak melampaui kinerja CPU dasar saat diperlukan. Secara default, instance ini dapat meledak untuk waktu yang terbatas sebelum penggunaan CPU mereka dibatasi. Anda dapat mengaktifkan mode tak terbatas secara opsional sehingga instance dapat meledak di luar garis dasar selama diperlukan. Hal ini memungkinkan aplikasi untuk mempertahankan kinerja CPU yang tinggi bila diperlukan. Biaya tambahan mungkin berlaku. Untuk informasi selengkapnya, lihat [Menggunakan grup Auto Scaling untuk meluncurkan instans performa burstable sebagai Unlimited](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Nama grup penempatan

Anda dapat menentukan grup penempatan dan menggunakan cluster atau strategi partisi untuk mempengaruhi bagaimana instance Anda secara fisik berada di pusat AWS data. Untuk grup Auto Scaling kecil, Anda juga dapat menggunakan strategi spread. Untuk informasi selengkapnya, lihat [Grup penempatan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Ada beberapa pertimbangan saat menggunakan grup penempatan dengan grup Auto Scaling:

- Jika grup penempatan ditentukan dalam template peluncuran dan grup Auto Scaling, grup penempatan untuk grup Auto Scaling akan diutamakan. Setelah grup dibuat, grup penempatan yang ditentukan dalam pengaturan grup Auto Scaling tidak dapat diubah.
- Di AWS CloudFormation, berhati-hatilah jika Anda menentukan grup penempatan di template peluncuran. Auto Scaling Amazon EC2 akan meluncurkan instans ke grup penempatan yang ditentukan. Namun, tidak CloudFormation akan menerima sinyal dari instans tersebut jika Anda menggunakan grup Auto Scaling (meskipun ini bisa berubah di masa mendatang).

[UpdatePolicy](#)

Opsi pembelian

Anda dapat memilih Minta Instans Spot untuk meminta Instans Spot pada harga Spot, dibatasi pada harga Sesuai Permintaan, dan pilih Kustomisasi untuk mengubah pengaturan Instans Spot default. Untuk grup Auto Scaling, Anda harus menentukan permintaan sekali waktu tanpa tanggal

akhir (default). Untuk informasi selengkapnya, lihat [Minta Instans Spot untuk aplikasi yang toleran terhadap kesalahan dan fleksibel](#). Pengaturan ini mungkin berguna dalam keadaan khusus, tetapi secara umum yang terbaik adalah membiarkannya tidak ditentukan dan membuat grup instance campuran sebagai gantinya. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).

Jika Anda menentukan permintaan Instans Spot di template peluncuran, Anda tidak dapat membuat grup instance campuran. Jika Anda mencoba menggunakan templat peluncuran yang meminta Instans Spot dengan grup instance campuran, Anda menerima pesan galat berikut: `Incompatible launch template: You cannot use a launch template that is set to request Spot Instances (InstanceMarketOptions) when you configure an Auto Scaling group with a mixed instances policy. Add a different launch template to the group and try again.`

Reservasi Kapasitas

Reservasi Kapasitas memungkinkan Anda untuk memesan kapasitas untuk instans Amazon EC2 Anda di Availability Zone tertentu untuk durasi berapa pun. Lihat informasi yang lebih lengkap di [Reservasi Kapasitas Atas Permintaan](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Anda dapat memilih apakah akan meluncurkan instance ke:

- Reservasi Kapasitas terbuka (Terbuka)
- Reservasi Kapasitas tertentu (Target berdasarkan ID)
- sekelompok Reservasi Kapasitas (Target berdasarkan grup)

Untuk menargetkan Reservasi Kapasitas tertentu, jenis instans dalam template peluncuran Anda harus cocok dengan jenis instance reservasi. Saat Anda membuat grup Auto Scaling, gunakan Availability Zone yang sama dengan Reservasi Kapasitas. Bergantung pada pilihan Wilayah AWS Anda, Anda dapat memilih untuk menargetkan Blok Kapasitas sebagai gantinya. Untuk informasi selengkapnya, lihat [Gunakan Blok Kapasitas untuk beban kerja pembelajaran mesin](#).

Untuk menargetkan sekelompok Reservasi Kapasitas, lihat [Gunakan Reservasi Kapasitas Sesuai Permintaan untuk memesan kapasitas di Availability Zone tertentu](#). Dengan menargetkan grup Reservasi Kapasitas, Anda dapat memiliki kapasitas yang didistribusikan di beberapa Availability Zone untuk meningkatkan ketahanan.

Penghunian

Amazon EC2 menyediakan tiga opsi untuk penyewaan instans EC2 Anda:

- **Shared (Shared)** — Beberapa Akun AWS dapat berbagi perangkat keras fisik yang sama. Ini adalah opsi penyewaan default saat meluncurkan instance.
- **Instans khusus (Khusus)** — Instance Anda berjalan pada perangkat keras penyewa tunggal. Tidak ada AWS pelanggan lain yang berbagi server fisik yang sama. Untuk informasi selengkapnya, lihat [Instans Khusus](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- **Dedicated Host (Dedicated Host)** — Instans berjalan pada server fisik yang didedikasikan untuk Anda gunakan. Menggunakan Host Khusus memudahkan Anda membawa lisensi Anda sendiri (BYOL) yang memiliki persyaratan perangkat keras khusus untuk EC2 dan memenuhi kasus penggunaan kepatuhan. Jika Anda memilih opsi ini, Anda harus menyediakan grup sumber daya host untuk grup sumber daya host Tenancy. Untuk informasi selengkapnya, lihat [Host Khusus](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Support for Dedicated Host hanya tersedia jika Anda menentukan grup sumber daya host. Anda tidak dapat menargetkan ID host tertentu atau menggunakan afinitas penempatan host.

- Jika Anda mencoba menggunakan template peluncuran yang menentukan ID host, Anda menerima pesan galat berikut: `Incompatible launch template: Tenancy host ID is not supported for Auto Scaling.`
- Jika Anda mencoba menggunakan template peluncuran yang menentukan afinitas penempatan host, Anda menerima pesan galat berikut: `Incompatible launch template: Auto Scaling does not support host placement affinity.`

Grup sumber daya host penyewaan

Dengan AWS License Manager, Anda dapat membawa lisensi Anda sendiri AWS dan mengelolanya secara terpusat. Grup sumber daya host adalah grup Host Khusus yang ditautkan ke konfigurasi lisensi License Manager tertentu. Grup sumber daya host memungkinkan Anda meluncurkan instans EC2 dengan mudah ke Host Khusus yang sesuai dengan kebutuhan lisensi perangkat lunak Anda. Anda tidak perlu mengalokasikan Host Khusus secara manual sebelumnya. Mereka secara otomatis dibuat sesuai kebutuhan. Perhatikan bahwa ketika Anda mengaitkan AMI dengan konfigurasi lisensi, AMI hanya dapat dikaitkan dengan satu grup sumber daya host pada satu waktu. Untuk informasi selengkapnya, lihat [Grup sumber daya host AWS License Manager di](#) Panduan Pengguna License Manager.

Konfigurasi lisensi

Dengan pengaturan ini, Anda dapat menentukan konfigurasi lisensi untuk instans Anda tanpa membatasi penyewaan mereka ke Host Khusus. Konfigurasi lisensi melacak lisensi perangkat lunak yang digunakan pada instance sehingga Anda dapat memantau penggunaan dan

kepatuhan lisensi Anda. Untuk informasi selengkapnya, lihat [Membuat lisensi yang dikelola sendiri](#) di Panduan Pengguna License Manager.

Metadata dapat diakses

Anda dapat memilih apakah akan mengaktifkan atau menonaktifkan akses ke titik akhir HTTP dari layanan metadata instance. Secara default, titik akhir HTTP diaktifkan. Jika Anda memilih untuk menonaktifkan titik akhir, akses ke metadata instans Anda dimatikan. Anda dapat menentukan kondisi untuk memerlukan ImDSv2 hanya ketika titik akhir HTTP diaktifkan. Untuk informasi selengkapnya, lihat [Konfigurasi opsi metadata instans](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Versi metadata

Anda dapat memilih untuk mewajibkan penggunaan Layanan Metadata Instans Versi 2 (IMDSv2) saat meminta metadata instans. Jika Anda tidak menentukan nilai, defaultnya adalah mendukung IMDSv1 dan IMDSv2. Untuk informasi selengkapnya, lihat [Konfigurasi opsi metadata instans](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Batas hop respons token metadata

Anda dapat mengatur jumlah hop jaringan yang diizinkan untuk token metadata. Jika Anda tidak menentukan nilai, defaultnya 1. Untuk informasi selengkapnya, lihat [Konfigurasi opsi metadata instans](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Data pengguna

Anda dapat menyesuaikan dan menyelesaikan konfigurasi instance Anda pada waktu peluncuran dengan menentukan skrip shell atau direktif cloud-init sebagai data pengguna. Data pengguna berjalan ketika instance awalnya dimulai, memungkinkan Anda menginstal aplikasi, dependensi, atau penyesuaian secara otomatis pada waktu peluncuran. Untuk informasi selengkapnya, lihat [Menjalankan perintah pada instans Linux saat diluncurkan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Jika Anda memiliki unduhan besar atau skrip yang kompleks, ini menambah waktu yang dibutuhkan instance untuk siap digunakan. Dalam hal ini, Anda mungkin perlu mengonfigurasi hook siklus hidup untuk menunda instance mencapai InService status hingga sepenuhnya disediakan. Untuk informasi selengkapnya tentang menambahkan hook siklus hidup ke grup Auto Scaling, lihat [Hook siklus aktif Amazon EC2 Auto Scaling](#)

Minta Instans Spot untuk aplikasi yang toleran terhadap kesalahan dan fleksibel

Di template peluncuran, Anda dapat meminta Instans Spot secara opsional tanpa tanggal atau durasi berakhir. Amazon EC2 Spot Instances adalah kapasitas cadangan yang tersedia dengan diskon besar dibandingkan dengan harga EC2 Sesuai Permintaan. Instans Spot adalah pilihan hemat biaya jika Anda dapat bersikap fleksibel tentang kapan aplikasi Anda berjalan dan apakah aplikasi Anda dapat diinterupsi. Untuk informasi selengkapnya tentang membuat template peluncuran yang meminta Instans Spot, lihat [Buat template peluncuran menggunakan pengaturan lanjutan](#).

Important

Spot Instances biasanya digunakan untuk melengkapi Instans Sesuai Permintaan. Untuk skenario ini, Anda dapat menentukan pengaturan yang sama yang digunakan untuk meluncurkan Instans Spot sebagai bagian dari pengaturan grup Auto Scaling Anda. Saat Anda menetapkan pengaturan sebagai bagian dari grup Auto Scaling, Anda dapat meminta untuk meluncurkan Spot Instances hanya setelah meluncurkan sejumlah Instans Sesuai Permintaan dan melanjutkan peluncuran beberapa kombinasi Instans Sesuai Permintaan dan Spot seiring grup diskalakan. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).

Topik ini menjelaskan cara meluncurkan hanya Instans Spot di grup Auto Scaling Anda dengan menentukan pengaturan dalam template peluncuran, bukan di grup Auto Scaling itu sendiri. [Informasi dalam topik ini juga berlaku untuk grup Auto Scaling yang meminta Instans Spot dengan konfigurasi peluncuran](#). Perbedaannya adalah bahwa konfigurasi peluncuran membutuhkan harga maksimum, tetapi untuk templat peluncuran, harga maksimum adalah opsional.

Saat Anda membuat template peluncuran untuk hanya meluncurkan Instans Spot, ingatlah pertimbangan berikut:

- Harga spot. Anda hanya membayar harga Spot saat ini untuk Instans Spot yang Anda luncurkan. Harga ini berubah perlahan dari waktu ke waktu berdasarkan tren jangka panjang dalam penawaran dan permintaan. Untuk informasi selengkapnya, lihat [Instans Spot dan Harga serta penghematan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Menetapkan harga maksimum Anda. Anda dapat secara opsional menyertakan harga maksimum per jam untuk Instans Spot di template peluncuran Anda. Jika harga maksimum Anda melebihi

harga Spot saat ini, layanan Amazon EC2 Spot segera memenuhi permintaan Anda jika kapasitas tersedia. Jika harga untuk Instans Spot naik di atas harga maksimum untuk instans yang sedang berjalan di grup Auto Scaling, itu akan mengakhiri instans Anda.

Warning

Aplikasi Anda mungkin tidak berjalan jika Anda tidak menerima Instans Spot, seperti ketika harga maksimum Anda terlalu rendah. Untuk memanfaatkan Instans Spot yang tersedia selama mungkin, tetapkan harga maksimum Anda mendekati harga Sesuai Permintaan.

- Menyeimbangkan di seluruh Availability Zone. Jika Anda menentukan beberapa Availability Zone, Amazon EC2 Auto Scaling mendistribusikan permintaan Spot di zona tertentu. Jika harga maksimum Anda terlalu rendah di satu Availability Zone untuk setiap permintaan yang akan dipenuhi, Amazon EC2 Auto Scaling memeriksa apakah permintaan dipenuhi di zona lain. Jika demikian, Amazon EC2 Auto Scaling membatalkan permintaan yang gagal dan mendistribusikan ulang permintaan tersebut di seluruh Availability Zone yang telah dipenuhi. Jika harga di Availability Zone tanpa permintaan yang terpenuhi turun hingga permintaan mendatang berhasil, Amazon EC2 Auto Scaling diseimbangkan ulang di seluruh Availability Zone.
- Pengakhiran Instans Spot. Instans Spot dapat dihentikan kapan saja. Layanan Amazon EC2 Spot dapat menghentikan Instans Spot di grup Auto Scaling karena ketersediaan, atau harga untuk, Instans Spot berubah. Saat menskalakan atau melakukan pemeriksaan kesehatan, Auto Scaling Amazon EC2 juga dapat menghentikan Instans Spot dengan cara yang sama seperti menghentikan Instans Sesuai Permintaan. Ketika sebuah instance dihentikan, penyimpanan apa pun akan dihapus.
- Mempertahankan kapasitas yang Anda inginkan. Saat Instans Spot dihentikan, Auto Scaling Amazon EC2 mencoba meluncurkan Instans Spot lain untuk mempertahankan kapasitas yang diinginkan untuk grup. Jika harga Spot saat ini kurang dari harga maksimum Anda, maka akan meluncurkan Instans Spot. Jika permintaan untuk Instans Spot tidak berhasil, ia terus mencoba.
- Mengubah harga maksimum Anda. Untuk mengubah harga maksimum Anda, buat template peluncuran baru atau perbarui template peluncuran yang ada dengan harga maksimum baru, lalu kaitkan dengan grup Auto Scaling Anda. Instans Spot yang ada terus berjalan selama harga maksimum yang ditentukan dalam template peluncuran yang digunakan untuk instans tersebut lebih tinggi dari harga Spot saat ini. Jika Anda tidak menetapkan harga maksimum, harga maksimum default adalah harga On-Demand.

Gunakan Blok Kapasitas untuk beban kerja pembelajaran mesin

Capacity Blocks membantu Anda memesan instans GPU yang sangat dicari di masa mendatang untuk mendukung beban kerja machine learning (ML) berdurasi pendek.

Untuk gambaran umum tentang Blok Kapasitas dan cara kerjanya, lihat [Blok Kapasitas untuk ML](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Anda dapat menggunakan Blok Kapasitas dengan jenis instans EC2 berikut dan Wilayah AWS:

Tipe instans	Wilayah
p5.48xlarge	AS Timur (Ohio), AS Timur (Virginia N.)
p4d.24xlarge	AS Timur (Ohio), AS Barat (Oregon)

Untuk mulai menggunakan Blok Kapasitas, Anda membuat reservasi kapasitas di Availability Zone tertentu. Blok Kapasitas dikirimkan sebagai reservasi *targeted* kapasitas dalam satu Availability Zone. Saat Anda membuat template peluncuran, tentukan ID reservasi dan jenis instans Blok Kapasitas. Kemudian, perbarui grup Auto Scaling Anda untuk menggunakan templat peluncuran yang Anda buat dan Zona Ketersediaan Blok Kapasitas. Saat reservasi Blok Kapasitas dimulai, gunakan penskalaan terjadwal untuk meluncurkan jumlah instans yang sama dengan reservasi Blok Kapasitas Anda.

Daftar Isi

- [Pedoman operasional](#)
- [Tentukan Blok Kapasitas di template peluncuran Anda](#)
- [Batasan](#)
- [Sumber daya terkait](#)

Pedoman operasional

Berikut ini adalah pedoman operasional dasar yang harus Anda ikuti saat menggunakan Blok Kapasitas dengan grup Auto Scaling.

- Skala dalam grup Auto Scaling Anda menjadi nol lebih dari 30 menit sebelum waktu akhir reservasi Blok Kapasitas. Amazon EC2 akan menghentikan instans apa pun yang masih berjalan 30 menit sebelum waktu akhir Blok Kapasitas.
- Kami menyarankan Anda menggunakan penskalaan terjadwal untuk memperkecil skala (menambahkan instance) dan menskalakan (menghapus instans) pada waktu reservasi yang sesuai. Untuk informasi selengkapnya, lihat [Skala terjadwal untuk Amazon EC2 Auto Scaling](#).
- Tambahkan kait siklus hidup sesuai kebutuhan untuk melakukan shutdown aplikasi yang anggun di dalam instance saat melakukan penskalaan. Berikan waktu yang cukup untuk menyelesaikan tindakan siklus hidup sebelum Amazon EC2 mulai menghentikan instans Anda secara paksa 30 menit sebelum waktu akhir reservasi Blok Kapasitas. Untuk informasi selengkapnya, lihat [Hook siklus aktif Amazon EC2 Auto Scaling](#).
- Pastikan grup Auto Scaling menunjuk ke versi template peluncuran yang benar selama seluruh durasi reservasi. Kami merekomendasikan untuk menunjuk ke versi tertentu dari template peluncuran alih-alih \$Latest versi \$Default atau.

Note

Jika Anda membiarkan instans Blok Kapasitas berjalan hingga akhir reservasi dan Amazon EC2 merebutnya kembali, aktivitas penskalaan untuk grup Auto Scaling Anda menyatakan bahwa itu adalah "taken out of service in response to an EC2 health check that indicated it had been terminated or stopped", meskipun sengaja direklamasi di akhir Blok Kapasitas. Demikian pula, Amazon EC2 Auto Scaling akan mencoba mengganti instance dengan cara yang sama seperti halnya untuk setiap contoh yang gagal dalam pemeriksaan kesehatan. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

Tentukan Blok Kapasitas di template peluncuran Anda

Untuk membuat template peluncuran yang menargetkan Blok Kapasitas tertentu untuk grup Auto Scaling Anda, gunakan salah satu metode berikut:

Console

Untuk menentukan Blok Kapasitas di template peluncuran (konsol)

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

2. Di bilah navigasi atas, pilih Wilayah AWS tempat Anda membuat Blok Kapasitas.
3. Pada panel navigasi, di bawah Instans, pilih Luncurkan Template.
4. Pilih Buat template peluncuran, dan buat template peluncuran. Sertakan ID Amazon Machine Image (AMI), jenis instans, dan setelan template peluncuran lainnya sesuai kebutuhan.
5. Perluas bagian Detail lanjutan untuk melihat pengaturan lanjutan.
6. Untuk opsi Pembelian, pilih Blok Kapasitas.
7. Untuk reservasi Kapasitas, pilih Target berdasarkan ID, lalu untuk reservasi Kapasitas - Target berdasarkan ID, pilih ID reservasi kapasitas dari Blok Kapasitas yang ada.
8. Setelah selesai, pilih Buat template peluncuran.

AWS CLI

Untuk menentukan Blok Kapasitas di template peluncuran Anda (AWS CLI)

Gunakan [create-launch-template](#) perintah berikut untuk membuat template peluncuran yang menentukan ID reservasi Blok Kapasitas yang ada. Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri.

```
aws ec2 create-launch-template --launch-template-name my-template-for-capacity-block \
  --version-description AutoScalingVersion1 --region us-east-2 \
  --launch-template-data file://config.json
```

Tip

Jika perintah ini menimbulkan kesalahan, pastikan Anda telah memperbarui AWS CLI secara lokal ke versi terbaru.

Isi dari config.json.

```
{
  "ImageId": "ami-04d5cc9b88example",
  "InstanceType": "p4d.24xlarge",
  "SecurityGroupIds": [
    "sg-903004f88example"
  ]
}
```

```

    ],
    "KeyName": "MyKeyPair",
    "InstanceMarketOptions": {
      "MarketType": "capacity-block"
    },
    "CapacityReservationSpecification": {
      "CapacityReservationTarget": {
        "CapacityReservationId": "cr-02168da1478b509e0"
      }
    }
  }
}

```

Berikut ini adalah output contoh.

```

{
  "LaunchTemplate": {
    "LaunchTemplateId": "lt-068f72b724example",
    "LaunchTemplateName": "my-template-for-capacity-block",
    "CreateTime": "2023-10-27T15:12:44.000Z",
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "DefaultVersionNumber": 1,
    "LatestVersionNumber": 1
  }
}

```

Anda dapat menggunakan [describe-launch-template-versions](#) perintah berikut untuk memverifikasi ID reservasi Blok Kapasitas yang terkait dengan templat peluncuran.

```

aws ec2 describe-launch-template-versions --launch-template-names my-template-for-
capacity-block \
  --region us-east-2

```

Berikut ini adalah contoh output untuk template peluncuran yang menentukan reservasi Blok Kapasitas.

```

{
  "LaunchTemplateVersions": [
    {
      "LaunchTemplateId": "lt-068f72b724example",
      "LaunchTemplateName": "my-template-for-capacity-block",
      "VersionNumber": 1,

```

```

    "CreateTime": "2023-10-27T15:12:44.000Z",
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "DefaultVersion": true,
    "LaunchTemplateData": {
      "ImageId": "ami-04d5cc9b88example",
      "InstanceType": "p5.48xlarge",
      "SecurityGroupIds": [
        "sg-903004f88example"
      ],
      "KeyName": "MyKeyPair",
      "InstanceMarketOptions": {
        "MarketType": "capacity-block"
      },
      "CapacityReservationSpecification": {
        "CapacityReservationTarget": {
          "CapacityReservationId": "cr-02168da1478b509e0"
        }
      }
    }
  }
}

```

Batasan

- Support for Capacity Blocks hanya tersedia jika grup Auto Scaling Anda memiliki konfigurasi yang kompatibel. Kelompok contoh campuran dan kolam hangat tidak didukung.
- Anda hanya dapat menargetkan satu Blok Kapasitas pada satu waktu.

Sumber daya terkait

- Untuk prasyarat dan rekomendasi penggunaan Instans P5, lihat [Memulai instans P5 di Panduan Pengguna Amazon EC2 untuk Instans Linux](#).
- Amazon EKS mendukung penggunaan Blok Kapasitas untuk mendukung durasi singkat, beban kerja machine learning (ML) Anda di kluster Amazon EKS. Untuk informasi selengkapnya, lihat [Blok Kapasitas untuk ML](#) di Panduan Pengguna Amazon EKS.
- Anda dapat menggunakan Blok Kapasitas dengan tipe instans dan Wilayah yang didukung. Namun, Reservasi Kapasitas Sesuai Permintaan memberikan fleksibilitas untuk kapasitas cadangan untuk jenis dan Wilayah instans lainnya. Untuk tutorial yang menunjukkan cara

menggunakan opsi Reservasi Kapasitas Sesuai Permintaan, lihat [Gunakan Reservasi Kapasitas Sesuai Permintaan untuk memesan kapasitas di Availability Zone tertentu](#).

Migrasi grup Auto Scaling Anda untuk meluncurkan template

Mulai tahun 2023, Anda tidak dapat menelepon `CreateLaunchConfiguration` dengan jenis instans Amazon EC2 baru yang dirilis setelah 31 Desember 2022. Untuk informasi selengkapnya, lihat [Konfigurasi peluncuran](#).

Untuk memigrasikan grup Auto Scaling Anda dari konfigurasi peluncuran ke templat peluncuran, lihat langkah-langkah berikut.

Important

Sebelum melanjutkan, konfirmasikan bahwa Anda memiliki izin yang diperlukan untuk bekerja dengan templat peluncuran. Untuk informasi selengkapnya, lihat [Izin untuk bekerja dengan template peluncuran](#).

Langkah 1: Temukan grup Auto Scaling yang menggunakan konfigurasi peluncuran

Untuk mengidentifikasi apakah Anda memiliki grup Auto Scaling yang masih menggunakan konfigurasi peluncuran, jalankan [describe-auto-scaling-groups](#) perintah berikut menggunakan AWS CLI Ganti **REGION** dengan Anda Wilayah AWS.

```
aws autoscaling describe-auto-scaling-groups --region REGION \
  --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`]'
```

Berikut ini adalah output contoh.

```
[
  {
    "AutoScalingGroupName": "group-1",
    "AutoScalingGroupARN": "arn",
    "LaunchConfigurationName": "my-launch-config",
    "MinSize": 1,
    "MaxSize": 5,
    "DesiredCapacity": 2,
```

```
"DefaultCooldown": 300,
"AvailabilityZones": [
    "us-west-2a",
    "us-west-2b",
    "us-west-2c"
],
"LoadBalancerNames": [],
"TargetGroupARNs": [],
"HealthCheckType": "EC2",
"HealthCheckGracePeriod": 300,
"Instances": [
    {
        "ProtectedFromScaleIn": false,
        "AvailabilityZone": "us-west-2a",
        "LaunchConfigurationName": "my-launch-config",
        "InstanceId": "i-05b4f7d5be44822a6",
        "InstanceType": "t3.micro",
        "HealthStatus": "Healthy",
        "LifecycleState": "InService"
    },
    {
        "ProtectedFromScaleIn": false,
        "AvailabilityZone": "us-west-2b",
        "LaunchConfigurationName": "my-launch-config",
        "InstanceId": "i-0c20ac468fa3049e8",
        "InstanceType": "t3.micro",
        "HealthStatus": "Healthy",
        "LifecycleState": "InService"
    }
],
"CreatedTime": "2023-03-09T22:15:11.611Z",
"SuspendedProcesses": [],
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
"EnabledMetrics": [],
"Tags": [
    {
        "ResourceId": "group-1",
        "ResourceType": "auto-scaling-group",
        "Key": "environment",
        "Value": "production",
        "PropagateAtLaunch": true
    }
],
"TerminationPolicies": [
```

```

    "Default"
  ],
  "NewInstancesProtectedFromScaleIn": false,
  "ServiceLinkedRoleARN": "arn",
  "TrafficSources": []
},

... additional groups ...

]

```

Atau, untuk menghapus semuanya kecuali nama grup Auto Scaling dengan nama konfigurasi peluncuran dan tag masing-masing di output, jalankan perintah berikut:

```

aws autoscaling describe-auto-scaling-groups --region REGION \
  --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`].{AutoScalingGroupName:
  AutoScalingGroupName, LaunchConfigurationName: LaunchConfigurationName, Tags: Tags}'

```

Berikut ini adalah contoh output.

```

[
  {
    "AutoScalingGroupName": "group-1",
    "LaunchConfigurationName": "my-launch-config",
    "Tags": [
      {
        "ResourceId": "group-1",
        "ResourceType": "auto-scaling-group",
        "Key": "environment",
        "Value": "production",
        "PropagateAtLaunch": true
      }
    ]
  },

  ... additional groups ...

]

```

Untuk informasi selengkapnya tentang pemfilteran, lihat [Memfilter AWS CLI output](#) di AWS Command Line Interface Panduan Pengguna.

Langkah 2: Salin konfigurasi peluncuran ke template peluncuran

Anda dapat menyalin konfigurasi peluncuran ke template peluncuran menggunakan prosedur berikut. Kemudian, Anda dapat menambahkannya ke grup Auto Scaling Anda.

Menyalin beberapa konfigurasi peluncuran menghasilkan templat peluncuran bernama identik. Untuk mengubah nama yang diberikan ke template peluncuran selama proses penyalinan, Anda harus menyalin konfigurasi peluncuran satu per satu.

Note

Fitur penyalinan hanya tersedia dari konsol.

Untuk menyalin konfigurasi peluncuran ke template peluncuran (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Di panel navigasi kiri, di bawah Auto Scaling, pilih Auto Scaling Groups.
3. Pilih Luncurkan konfigurasi di dekat bagian atas halaman. Saat diminta konfirmasi, pilih Lihat konfigurasi peluncuran untuk mengonfirmasi bahwa Anda ingin melihat halaman Konfigurasi peluncuran.
4. Pilih konfigurasi peluncuran yang ingin Anda salin dan pilih Salin untuk meluncurkan template, Salin dipilih. Ini akan menyiapkan templat peluncuran baru dengan nama dan opsi yang sama dengan konfigurasi peluncuran yang Anda pilih.
5. Untuk Nama templat peluncuran baru, Anda dapat menggunakan nama konfigurasi peluncuran (default) atau masukkan nama baru. Nama templat peluncuran harus unik.
6. (Opsional) Pilih Buat grup Auto Scaling menggunakan template baru.

Anda dapat melewati langkah ini untuk menyelesaikan penyalinan konfigurasi peluncuran. Anda tidak perlu membuat grup Auto Scaling baru.

7. Pilih Salin.

Untuk menyalin semua konfigurasi peluncuran untuk meluncurkan templat (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pada panel navigasi, di bawah Auto Scaling, pilih Luncurkan Konfigurasi.

3. Pilih Salin untuk meluncurkan template, Salin semua. Ini menyalin setiap konfigurasi peluncuran di Wilayah saat ini ke templat peluncuran baru dengan nama dan opsi yang sama.
4. Pilih Salin.

Langkah 3: Perbarui grup Auto Scaling untuk menggunakan template peluncuran

Setelah membuat template peluncuran, Anda siap menambahkannya ke grup Auto Scaling Anda.

Untuk memperbarui grup Auto Scaling untuk menggunakan template peluncuran (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel terpisah terbuka di bagian bawah halaman, menunjukkan informasi tentang grup yang dipilih.

3. Di tab Detail pilih, pilih Konfigurasi Peluncuran, Edit.
4. Pilih Beralih ke templat peluncuran.
5. Untuk Templat peluncuran, pilih templat peluncuran Anda.
6. Untuk Versi, pilih versi templat peluncuran, sesuai kebutuhan. Setelah Anda membuat versi templat peluncuran, Anda dapat memilih apakah grup Auto Scaling menggunakan versi default, terbaru, atau spesifik dari templat peluncuran saat memperluas.
7. Pilih Perbarui.

Untuk memperbarui grup Auto Scaling untuk menggunakan template peluncuran (AWS CLI)

[update-auto-scaling-group](#) Perintah berikut memperbarui grup Auto Scaling yang ditentukan untuk menggunakan versi awal template peluncuran yang ditentukan.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
--launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='1'
```

Untuk lebih banyak contoh penggunaan perintah CLI untuk memperbarui grup Auto Scaling untuk menggunakan template peluncuran, lihat. [Memperbarui grup Auto Scaling untuk menggunakan template peluncuran](#)

Langkah 4: Ganti instans Anda

Setelah Anda mengganti konfigurasi peluncuran dengan template peluncuran, setiap instance baru akan menggunakan template peluncuran baru. Instans yang ada tidak terpengaruh.

Untuk memperbarui instans yang ada, Anda dapat memulai penyegaran instans untuk mengganti instans di grup Auto Scaling alih-alih mengganti instans secara manual beberapa sekaligus. Untuk informasi selengkapnya, lihat [Menggunakan penyegaran instans untuk memperbarui instance dalam grup Auto Scaling](#). Jika grupnya besar, penyegaran instance bisa sangat membantu.

Atau, Anda dapat mengizinkan penskalaan otomatis untuk secara bertahap mengganti instans yang ada dengan instans baru berdasarkan [kebijakan penghentian](#) grup, atau Anda dapat menghentikannya. Penghentian manual memaksa grup Auto Scaling Anda meluncurkan instans baru untuk mempertahankan kapasitas yang diinginkan grup. Untuk informasi selengkapnya, lihat [Mengakhiri instance](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Informasi tambahan

Untuk informasi selengkapnya, lihat [Auto Scaling Amazon EC2 tidak akan lagi menambahkan dukungan untuk fitur EC2 baru untuk Meluncurkan](#) Konfigurasi di Blog Komputasi. AWS

Untuk topik yang membawa Anda melalui cara memigrasikan AWS CloudFormation tumpukan dari konfigurasi peluncuran ke templat peluncuran, lihat [Migrasikan AWS CloudFormation tumpukan untuk meluncurkan templat](#)

Migrasikan AWS CloudFormation tumpukan untuk meluncurkan templat

Anda dapat memigrasikan templat AWS CloudFormation tumpukan yang ada dari konfigurasi peluncuran ke templat peluncuran. Untuk melakukan ini, tambahkan template peluncuran langsung ke template tumpukan yang ada dan kemudian kaitkan template peluncuran dengan grup Auto Scaling di template tumpukan. Kemudian, gunakan template yang dimodifikasi untuk memperbarui tumpukan Anda.

Saat bermigrasi ke templat peluncuran, topik ini menghemat waktu Anda dengan memberikan instruksi untuk menulis ulang konfigurasi peluncuran di templat CloudFormation tumpukan Anda sebagai templat peluncuran. Untuk informasi selengkapnya tentang memigrasi konfigurasi

peluncuran untuk meluncurkan templat, lihat. [Migrasi grup Auto Scaling Anda untuk meluncurkan template](#)

Topik

- [Temukan grup Auto Scaling yang menggunakan konfigurasi peluncuran](#)
- [Perbarui tumpukan untuk menggunakan template peluncuran](#)
- [Memahami perilaku pembaruan sumber daya tumpukan](#)
- [Lacak migrasi](#)
- [Luncurkan referensi pemetaan konfigurasi](#)

Temukan grup Auto Scaling yang menggunakan konfigurasi peluncuran

Untuk menemukan grup Auto Scaling yang menggunakan konfigurasi peluncuran

- Gunakan [describe-auto-scaling-groups](#) perintah berikut untuk mencantumkan nama grup Auto Scaling yang menggunakan konfigurasi peluncuran di Wilayah yang ditentukan. Sertakan `--filters` opsi untuk mempersempit hasil ke grup yang terkait dengan CloudFormation tumpukan (dengan memfilter dengan kunci `aws:cloudformation:stack-name` tag).

```
aws autoscaling describe-auto-scaling-groups --region REGION \  
  --filters Name=tag-key,Values=aws:cloudformation:stack-name \  
  --query 'AutoScalingGroups[?LaunchConfigurationName!  
= `null` ].AutoScalingGroupName'
```

Berikut ini adalah contoh output.

```
[  
  "{stack-name}-group-1",  
  "{stack-name}-group-2",  
  "{stack-name}-group-3"  
]
```

Anda dapat menemukan AWS CLI perintah tambahan yang berguna untuk menemukan grup Auto Scaling untuk memigrasi dan memfilter output. [Migrasi grup Auto Scaling Anda untuk meluncurkan template](#)

⚠ Important

Jika sumber daya tumpukan Anda memiliki AWSEB namanya, ini berarti mereka dibuat melalui AWS Elastic Beanstalk. Dalam hal ini, Anda harus memperbarui lingkungan Beanstalk untuk mengarahkan Elastic Beanstalk untuk menghapus konfigurasi peluncuran dan menggantinya dengan template peluncuran.

Perbarui tumpukan untuk menggunakan template peluncuran

Ikuti langkah-langkah di bagian ini untuk melakukan hal berikut:

- Tulis ulang konfigurasi peluncuran sebagai template peluncuran menggunakan properti template peluncuran yang setara.
- Kaitkan template peluncuran baru dengan grup Auto Scaling.
- Terapkan pembaruan ini.

Untuk memodifikasi template tumpukan dan memperbarui tumpukan

1. Ikuti prosedur umum yang sama untuk memodifikasi template tumpukan yang dijelaskan dalam [Memodifikasi template tumpukan](#) di AWS CloudFormation Panduan Pengguna.
2. Tulis ulang konfigurasi peluncuran sebagai template peluncuran. Lihat contoh berikut ini:

Contoh: Konfigurasi peluncuran sederhana

```
---
Resources:
  myLaunchConfig:
    Type: AWS::AutoScaling::LaunchConfiguration
    Properties:
      ImageId: ami-02354e95b3example
      InstanceType: t3.micro
      SecurityGroups:
        - !Ref EC2SecurityGroup
      KeyName: MyKeyPair
      BlockDeviceMappings:
        - DeviceName: /dev/xvda
          Ebs:
            VolumeSize: 150
```

```

        DeleteOnTermination: true
    UserData:
        Fn::Base64: !Sub |
            #!/bin/bash -xe
            yum install -y aws-cfn-bootstrap
            /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource myASG
            --region ${AWS::Region}

```

Contoh: Template peluncuran setara

```

---
Resources:
  myLaunchTemplate:
    Type: AWS::EC2::LaunchTemplate
    Properties:
      LaunchTemplateName: !Sub ${AWS::StackName}-launch-template
      LaunchTemplateData:
        ImageId: ami-02354e95b3example
        InstanceType: t3.micro
        SecurityGroupIds:
          - Ref! EC2SecurityGroup
        KeyName: MyKeyPair
        BlockDeviceMappings:
          - DeviceName: /dev/xvda
            Ebs:
              VolumeSize: 150
              DeleteOnTermination: true
        UserData:
          Fn::Base64: !Sub |
              #!/bin/bash -x
              yum install -y aws-cfn-bootstrap
              /opt/aws/bin/cfn-signal -e $? --stack ${AWS::StackName} --resource
              myASG --region ${AWS::Region}

```

Untuk informasi referensi untuk semua properti yang didukung Amazon EC2, lihat [AWS::EC2::LaunchTemplate](#) di AWS CloudFormation Panduan Pengguna.

Perhatikan bagaimana template peluncuran menyertakan `LaunchTemplateName` properti dengan nilai `!Sub ${AWS::StackName}-launch-template`. Ini diperlukan jika Anda ingin nama template peluncuran menyertakan nama tumpukan.

3. Jika **IamInstanceProfile** properti hadir dalam konfigurasi peluncuran Anda, Anda harus mengubahnya menjadi struktur dan menentukan nama atau ARN dari profil instance. Sebagai contoh, lihat [AWS::EC2::LaunchTemplate](#).
4. Jika **AssociatePublicIpAddress**, **InstanceMonitoring**, atau **PlacementTenancy** properti ada dalam konfigurasi peluncuran Anda, Anda harus mengonversinya menjadi struktur. Sebagai contoh, lihat [AWS::EC2::LaunchTemplate](#).

Pengecualian adalah ketika nilai `MapPublicIpOnLaunch` properti pada subnet yang Anda gunakan untuk grup Auto Scaling cocok dengan nilai properti dalam `AssociatePublicIpAddress` konfigurasi peluncuran Anda. Dalam hal ini, Anda dapat mengabaikan `AssociatePublicIpAddress` properti. `AssociatePublicIpAddress` properti hanya digunakan untuk mengganti `MapPublicIpOnLaunch` properti untuk mengubah apakah instance menerima alamat IPv4 publik saat peluncuran.

5. Anda dapat menyalin grup keamanan dari **SecurityGroups** properti ke salah satu dari dua tempat di template peluncuran Anda. Biasanya, Anda menyalin grup keamanan ke `SecurityGroupIds` properti. Namun, jika Anda membuat `NetworkInterfaces` struktur dalam template peluncuran Anda untuk menentukan `AssociatePublicIpAddress` properti, maka Anda harus menyalin grup keamanan ke `Groups` properti antarmuka jaringan sebagai gantinya.
6. Jika ada `BlockDeviceMapping` struktur yang ada dalam konfigurasi peluncuran Anda dengan **NoDevice** set `true`, maka Anda harus menentukan string kosong untuk `NoDevice` di template peluncuran agar Amazon EC2 menghilangkan perangkat.
7. Jika **SpotPrice** properti hadir dalam konfigurasi peluncuran Anda, kami sarankan Anda menghilangkannya dari template peluncuran Anda. Instans Spot Anda akan diluncurkan pada harga Spot saat ini. Harga ini tidak akan pernah melebihi harga On-Demand.

Untuk meminta Instans Spot, Anda memiliki dua opsi yang saling eksklusif:

- Yang pertama adalah menggunakan `InstanceMarketOptions` struktur dalam template peluncuran Anda (tidak disarankan). Untuk informasi selengkapnya, lihat [AWS::EC2::LaunchTemplate InstanceMarketOptions](#) di Panduan AWS CloudFormation Pengguna.
- Yang lainnya adalah menambahkan `MixedInstancesPolicy` struktur ke grup Auto Scaling Anda. Melakukannya memberi Anda lebih banyak opsi untuk bagaimana Anda membuat permintaan. Permintaan Instans Spot di template peluncuran Anda tidak mendukung lebih dari satu pilihan jenis instans per grup Auto Scaling. Namun, kebijakan

instance campuran mendukung lebih dari satu pemilihan tipe instans per grup Auto Scaling. Permintaan Instans Spot mendapat manfaat dari memiliki lebih dari satu jenis instans untuk dipilih. Untuk informasi selengkapnya, lihat [AWS::AutoScaling::AutoScalingMixedInstancesPolicy](#) di Panduan AWS CloudFormation Pengguna.

8. Hapus **LaunchConfigurationName** properti dari sumber daya [AWS::AutoScaling::AutoScalingGroup](#) . Tambahkan template peluncuran di tempatnya.

Dalam contoh berikut, fungsi intrinsik [Ref](#) mendapatkan ID [AWS::EC2::LaunchTemplate](#) sumber daya dengan ID logis. `myLaunchTemplate` [GetAtt](#) Fungsi mendapatkan nomor versi terbaru (misalnya,1) dari template peluncuran untuk `Version` properti.

Contoh: Tanpa kebijakan instance campuran

```
---
Resources:
  myASG:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      LaunchTemplate:
        LaunchTemplateId: !Ref myLaunchTemplate
        Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    ...
```

Contoh: Dengan kebijakan instance campuran

```
---
Resources:
  myASG:
    Type: AWS::AutoScaling::AutoScalingGroup
    Properties:
      MixedInstancesPolicy:
        LaunchTemplate:
          LaunchTemplateSpecification:
            LaunchTemplateId: !Ref myLaunchTemplate
            Version: !GetAtt myLaunchTemplate.LatestVersionNumber
    ...
```

Untuk informasi referensi untuk semua properti yang didukung Auto Scaling Amazon EC2, [AWS::AutoScaling::AutoScaling](#) lihat [AWS::AutoScaling::AutoScaling](#) di AWS CloudFormation Panduan Pengguna.

9. Saat Anda siap untuk menerapkan pembaruan ini, ikuti CloudFormation prosedur untuk memperbarui tumpukan dengan templat tumpukan yang dimodifikasi. Untuk informasi selengkapnya, lihat [Memodifikasi template tumpukan](#) di Panduan AWS CloudFormation Pengguna.

Memahami perilaku pembaruan sumber daya tumpukan

CloudFormation memperbarui sumber daya tumpukan dengan membandingkan perubahan antara template yang diperbarui yang Anda berikan, dan konfigurasi sumber daya yang Anda jelaskan di versi template tumpukan Anda sebelumnya. Konfigurasi sumber daya yang tidak berubah tetap tidak terpengaruh selama proses pembaruan.

CloudFormation mendukung [UpdatePolicy](#) atribut untuk grup Auto Scaling. Selama pembaruan, jika `UpdatePolicy` diatur ke `AutoScalingRollingUpdate`, CloudFormation menggantikan `InService` instance setelah Anda melakukan langkah-langkah dalam prosedur ini. Jika `UpdatePolicy` disetel ke `AutoScalingReplacingUpdate`, CloudFormation ganti grup Auto Scaling dan kolam hangatnya (jika ada).

Jika Anda tidak menentukan `UpdatePolicy` atribut untuk grup Auto Scaling, template peluncuran akan diperiksa kebenarannya, tetapi CloudFormation tidak menerapkan perubahan apa pun di seluruh instance dalam grup Auto Scaling. Semua instance baru akan menggunakan template peluncuran Anda, tetapi instance yang ada terus berjalan dengan konfigurasi peluncuran yang awalnya diluncurkan (meskipun konfigurasi peluncuran tidak ada). Pengecualiannya adalah ketika Anda mengubah opsi pembelian, misalnya, dengan menambahkan kebijakan instance campuran. Dalam hal ini, grup Auto Scaling Anda secara bertahap mengganti instans yang ada dengan instans baru agar sesuai dengan opsi pembelian baru.

Lacak migrasi

Untuk melacak migrasi

1. Di [Konsol AWS CloudFormation](#), pilih tumpukan yang diperbarui dan kemudian pilih tab Kejadian untuk melihat kejadian tumpukan.

2. Untuk memperbarui daftar acara dengan acara terbaru, pilih tombol segarkan di CloudFormation konsol.
3. Saat tumpukan Anda diperbarui, Anda akan melihat beberapa peristiwa untuk setiap pembaruan sumber daya. Jika Anda melihat pengecualian di kolom Alasan status yang menunjukkan masalah saat mencoba membuat templat peluncuran, lihat [Memecahkan Masalah Auto Scaling Amazon EC2: Luncurkan templat](#) penyebab potensial.
4. (Opsional) Bergantung pada penggunaan UpdatePolicy atribut, Anda dapat memantau kemajuan grup Auto Scaling Anda dari halaman [grup Auto Scaling](#) di konsol Amazon EC2. Pilih grup Auto Scaling. Pada tab Aktivitas, di bawah Riwayat aktivitas, kolom Status menunjukkan apakah grup Auto Scaling Anda telah berhasil meluncurkan atau menghentikan instance, atau apakah aktivitas penskalaan masih berlangsung.
5. Saat pembaruan tumpukan selesai, CloudFormation mengeluarkan acara UPDATE_COMPLETE tumpukan. Untuk informasi selengkapnya, lihat [Memantau kemajuan pembaruan tumpukan](#) di Panduan AWS CloudFormation Pengguna.
6. Setelah pembaruan tumpukan selesai, buka [halaman Launch templates](#) dan [Luncurkan halaman konfigurasi](#) konsol Amazon EC2. Anda akan melihat template peluncuran baru dibuat, dan konfigurasi peluncuran dihapus.

Luncurkan referensi pemetaan konfigurasi

Untuk tujuan referensi, tabel berikut mencantumkan semua properti tingkat atas dalam [AWS::AutoScaling::LaunchConfiguration](#) sumber daya dengan properti yang sesuai di [AWS::EC2::LaunchTemplate](#) sumber daya.

Luncurkan properti sumber konfigurasi	Luncurkan properti target template
AssociatePublicIpAddress	NetworkInterfaces.AssociatePublicIpAddress
BlockDeviceMappings	BlockDeviceMappings
ClassicLinkVPCId	Tidak tersedia ¹
ClassicLinkVPCSecurityGroups	Tidak tersedia ¹
EbsOptimized	EbsOptimized

Luncurkan properti sumber konfigurasi	Luncurkan properti target template
IamInstanceProfile	Entah <code>IamInstanceProfile.Arn</code> atau <code>IamInstanceProfile.Name</code> , tapi tidak keduanya
ImageId	ImageId
InstanceId	InstanceId
InstanceMonitoring	<code>Monitoring.Enabled</code>
InstanceType	InstanceType
KernelId	KernelId
KeyName	KeyName
LaunchConfigurationName	LaunchTemplateName
MetadataOptions	MetadataOptions
PlacementTenancy	<code>Placement.Tenancy</code>
RamDiskId	RamDiskId
SecurityGroups	Entah <code>SecurityGroupIds</code> atau <code>NetworkInterfaces.Groups</code> , tapi tidak keduanya
SpotPrice	<code>InstanceMarketOptions.SpotOptions.MaxPrice</code>
UserData	UserData

¹ `ClassicLinkVPCSecurityGroups` Properti `ClassicLinkVPCId` dan tidak tersedia untuk digunakan dalam template peluncuran karena EC2-Classic tidak lagi tersedia.

Contoh untuk membuat dan mengelola template peluncuran dengan AWS Command Line Interface (AWS CLI)

Anda dapat membuat dan mengelola template peluncuran melalui AWS Management Console, AWS CLI, atau SDK. Bagian ini menunjukkan kepada Anda contoh membuat dan mengelola templat peluncuran untuk Auto Scaling Amazon EC2 dari AWS CLI

Daftar Isi

- [Contoh penggunaan](#)
- [Buat template peluncuran dasar](#)
- [Tentukan tag yang menandai instance saat diluncurkan](#)
- [Tentukan peran IAM untuk diteruskan ke instance](#)
- [Tetapkan alamat IP publik](#)
- [Tentukan skrip data pengguna yang mengonfigurasi instance saat peluncuran](#)
- [Tentukan pemetaan perangkat blok](#)
- [Tentukan Host Khusus untuk membawa lisensi perangkat lunak dari vendor eksternal](#)
- [Tentukan antarmuka jaringan yang ada](#)
- [Buat beberapa antarmuka jaringan](#)
- [Kelola templat peluncuran Anda](#)
- [Memperbarui grup Auto Scaling untuk menggunakan template peluncuran](#)

Contoh penggunaan

```
{
  "LaunchTemplateName": "my-template-for-auto-scaling",
  "VersionDescription": "test description",
  "LaunchTemplateData": {
    "ImageId": "ami-04d5cc9b88example",
    "InstanceType": "t2.micro",
    "SecurityGroupIds": [
      "sg-903004f88example"
    ],
    "KeyName": "MyKeyPair",
    "Monitoring": {
      "Enabled": true
    }
  }
}
```

```
    },
    "Placement": {
      "Tenancy": "dedicated"
    },
    "CreditSpecification": {
      "CpuCredits": "unlimited"
    },
    "MetadataOptions": {
      "HttpTokens": "required",
      "HttpPutResponseHopLimit": 1,
      "HttpEndpoint": "enabled"
    }
  }
}
```

Buat template peluncuran dasar

Untuk membuat template peluncuran dasar, gunakan [create-launch-template](#) perintah sebagai berikut, dengan modifikasi ini:

- Ganti `ami-04d5cc9b88example` dengan ID AMI untuk meluncurkan instance.
- Ganti `t2.micro` dengan tipe instans yang kompatibel dengan AMI yang Anda tentukan.

Contoh ini membuat template peluncuran dengan nama *my-template-for-auto-scaling*. Jika instans yang dibuat oleh templat peluncuran ini diluncurkan dalam VPC default, mereka menerima alamat IP publik secara default. Jika instans diluncurkan dalam VPC nondefault, mereka tidak menerima alamat IP publik secara default.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data
  '{"ImageId": "ami-04d5cc9b88example", "InstanceType": "t2.micro"}'
```

Untuk informasi selengkapnya tentang mengutip parameter berformat JSON, lihat [Menggunakan tanda kutip dengan string](#) di Panduan Pengguna. AWS CLI AWS Command Line Interface

Atau, Anda dapat menentukan parameter berformat JSON dalam file konfigurasi.

Contoh berikut membuat template peluncuran dasar, mereferensikan file konfigurasi untuk nilai-nilai parameter template peluncuran.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data file://config.json
```

Isi dari config.json:

```
{
  "ImageId": "ami-04d5cc9b88example",
  "InstanceType": "t2.micro"
}
```

Tentukan tag yang menandai instance saat diluncurkan

Contoh berikut menambahkan tag (misalnya, `purpose=webserver`) ke instance saat peluncuran.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"TagSpecifications":[{"ResourceType":"instance","Tags":
[{"Key": "purpose", "Value": "webserver"}]}], "ImageId": "ami-04d5cc9b88example", "InstanceType": "t2.
```

Note

Jika Anda menentukan tag instance di template peluncuran dan kemudian Anda memilih untuk menyebarkan tag grup Auto Scaling Anda ke instance-nya, semua tag digabungkan. Jika kunci tag yang sama ditentukan untuk tag di template peluncuran dan tag di grup Auto Scaling Anda, maka nilai tag dari grup akan diutamakan.

Tentukan peran IAM untuk diteruskan ke instance

Contoh berikut menentukan nama profil instance yang terkait dengan peran IAM untuk diteruskan ke instance saat peluncuran. Untuk informasi selengkapnya, lihat [Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2](#).

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"IamInstanceProfile":{"Name": "my-instance-
profile"}, "ImageId": "ami-04d5cc9b88example", "InstanceType": "t2.micro"}'
```

Tetapkan alamat IP publik

[create-launch-template](#) Contoh berikut mengonfigurasi template peluncuran untuk menetapkan alamat publik ke instance yang diluncurkan di VPC nondefault.

Note

Saat Anda menentukan antarmuka jaringan, tentukan nilai Groups yang sesuai dengan grup keamanan untuk VPC tempat grup Auto Scaling Anda akan meluncurkan instans. Tentukan subnet VPC sebagai properti dari grup Auto Scaling.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":
[{"DeviceIndex":0,"AssociatePublicIpAddress":true,"Groups":
["sg-903004f88example"],"DeleteOnTermination":true]}',"ImageId":"ami-04d5cc9b88example","InstanceType":t2.micro}'
```

Tentukan skrip data pengguna yang mengonfigurasi instance saat peluncuran

Contoh berikut menentukan script data pengguna sebagai string base64 yang dikodekan yang mengkonfigurasi instance saat peluncuran. [create-launch-template](#) Perintah ini membutuhkan data pengguna yang dikodekan base64.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data
  '{"UserData":"IyEvYmLuL2Jhc...","ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"}'
```

Tentukan pemetaan perangkat blok

[create-launch-template](#) Contoh berikut membuat template peluncuran dengan pemetaan perangkat blok: volume EBS 22 gigabyte yang dipetakan ke. /dev/xvdcz /dev/xvdczVolume menggunakan tipe volume General Purpose SSD (gp2) dan dihapus saat mengakhiri instance yang dilampirkan.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
```

```
--launch-template-data '{"BlockDeviceMappings":[{"DeviceName":"/dev/xvdcz","Ebs":{"VolumeSize":22,"VolumeType":"gp2","DeleteOnTermination":true}}],"ImageId":"ami-04d5cc9b88example"
```

Tentukan Host Khusus untuk membawa lisensi perangkat lunak dari vendor eksternal

Jika Anda menentukan penyewaan host, Anda dapat menentukan grup sumber daya host dan konfigurasi lisensi License Manager untuk membawa lisensi perangkat lunak yang memenuhi syarat dari vendor eksternal. Kemudian, Anda dapat menggunakan lisensi pada instans EC2 dengan menggunakan perintah berikut. [create-launch-template](#)

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"Placement":
{"Tenancy":"host","HostResourceGroupArn":"arn"},"LicenseSpecifications":
[{"LicenseConfigurationArn":"arn"}],"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"
```

Tentukan antarmuka jaringan yang ada

[create-launch-template](#) Contoh berikut mengkonfigurasi antarmuka jaringan utama untuk menggunakan antarmuka jaringan yang ada.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":
[{"DeviceIndex":0,"NetworkInterfaceId":"eni-
b9a5ac93","DeleteOnTermination":false},"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"
```

Buat beberapa antarmuka jaringan

[create-launch-template](#) Contoh berikut menambahkan antarmuka jaringan sekunder. Antarmuka jaringan utama memiliki indeks perangkat 0, dan antarmuka jaringan sekunder memiliki indeks perangkat 1.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":[{"DeviceIndex":0,"Groups":
["sg-903004f88example"],"DeleteOnTermination":true},{ "DeviceIndex":1,"Groups":
["sg-903004f88example"],"DeleteOnTermination":true},"ImageId":"ami-04d5cc9b88example","InstanceType":"t2.micro"
```

Jika Anda menggunakan jenis instans yang mendukung beberapa kartu jaringan dan Elastic Fabric Adapters (EFA), Anda dapat menambahkan antarmuka sekunder ke kartu jaringan sekunder dan mengaktifkan EFA dengan menggunakan perintah berikut [create-launch-template](#). Untuk informasi selengkapnya, lihat [Menambahkan EFA ke template peluncuran](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling --
version-description version1 \
  --launch-template-data '{"NetworkInterfaces":
[{"NetworkCardIndex":0,"DeviceIndex":0,"Groups":
["sg-7c2270198example"],"InterfaceType":"efa","DeleteOnTermination":true},
{"NetworkCardIndex":1,"DeviceIndex":1,"Groups":
["sg-7c2270198example"],"InterfaceType":"efa","DeleteOnTermination":true}]',"ImageId":"ami-09d95
```

Warning

Jenis instans p4d.24xlarge menimbulkan biaya lebih tinggi daripada contoh lain di bagian ini. Untuk informasi selengkapnya tentang harga instans P4d, lihat harga Instans P4d [Amazon EC2](#).

Note

Melampirkan beberapa antarmuka jaringan dari subnet yang sama ke instance dapat memperkenalkan perutean asimetris, terutama pada instance yang menggunakan varian Linux non-Amazon. Jika Anda memerlukan konfigurasi jenis ini, Anda harus mengkonfigurasi antarmuka jaringan sekunder di dalam OS. Sebagai contoh, [lihat Bagaimana cara membuat antarmuka jaringan sekunder saya berfungsi di instance Ubuntu EC2 saya?](#) di pusat AWS pengetahuan.

Kelola templat peluncuran Anda

AWS CLI Termasuk beberapa perintah lain yang membantu Anda mengelola template peluncuran Anda.

Daftar Isi

- [Buat daftar dan jelaskan templat peluncuran Anda](#)

- [Buat versi templat peluncuran](#)
- [Hapus versi templat peluncuran](#)
- [Hapus templat peluncuran](#)

Buat daftar dan jelaskan templat peluncuran Anda

Anda dapat menggunakan dua AWS CLI perintah untuk mendapatkan informasi tentang template peluncuran Anda: [describe-launch-templates](#) dan [describe-launch-template-versions](#).

[describe-launch-templates](#) Perintah ini memungkinkan Anda untuk mendapatkan daftar dari salah satu template peluncuran yang telah Anda buat. Anda dapat menggunakan opsi untuk memfilter hasil pada nama template peluncuran, waktu buat, kunci tag, atau kombinasi nilai kunci tag. Perintah ini mengembalikan informasi ringkasan tentang templat peluncuran Anda, termasuk pengenalan templat peluncuran, versi terbaru, dan versi default.

Contoh berikut memberikan ringkasan template peluncuran yang ditentukan.

```
aws ec2 describe-launch-templates --launch-template-names my-template-for-auto-scaling
```

Berikut ini adalah contoh respons.

```
{
  "LaunchTemplates": [
    {
      "LaunchTemplateId": "lt-068f72b729example",
      "LaunchTemplateName": "my-template-for-auto-scaling",
      "CreateTime": "2020-02-28T19:52:27.000Z",
      "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
      "DefaultVersionNumber": 1,
      "LatestVersionNumber": 1
    }
  ]
}
```

Jika Anda tidak menggunakan `--launch-template-names` opsi untuk membatasi output ke satu template peluncuran, informasi tentang semua template peluncuran Anda akan dikembalikan.

[describe-launch-template-versions](#) Perintah berikut memberikan informasi yang menjelaskan versi template peluncuran yang ditentukan.


```
aws ec2 describe-launch-template-versions --launch-template-id lt-068f72b729example
```

Berikut ini adalah contoh respons.

```
{
  "LaunchTemplateVersions": [
    {
      "VersionDescription": "version1",
      "LaunchTemplateId": "lt-068f72b729example",
      "LaunchTemplateName": "my-template-for-auto-scaling",
      "VersionNumber": 1,
      "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
      "LaunchTemplateData": {
        "TagSpecifications": [
          {
            "ResourceType": "instance",
            "Tags": [
              {
                "Key": "purpose",
                "Value": "webserver"
              }
            ]
          }
        ],
        "ImageId": "ami-04d5cc9b88example",
        "InstanceType": "t2.micro",
        "NetworkInterfaces": [
          {
            "DeviceIndex": 0,
            "DeleteOnTermination": true,
            "Groups": [
              "sg-903004f88example"
            ],
            "AssociatePublicIpAddress": true
          }
        ],
        "DefaultVersion": true,
        "CreateTime": "2020-02-28T19:52:27.000Z"
      }
    ]
  }
}
```

Buat versi templat peluncuran

[create-launch-template-version](#) Perintah berikut membuat versi template peluncuran baru berdasarkan versi 1 dari template peluncuran dan menentukan ID AMI yang berbeda.

```
aws ec2 create-launch-template-version --launch-template-id lt-068f72b729example --  
version-description version2 \  
--source-version 1 --launch-template-data "ImageId=ami-c998b6b2example"
```

Untuk mengatur versi default template peluncuran, gunakan [modify-launch-template](#) perintah.

Hapus versi templat peluncuran

[delete-launch-template-versions](#) Perintah berikut menghapus versi template peluncuran yang ditentukan.

```
aws ec2 delete-launch-template-versions --launch-template-id lt-068f72b729example --  
versions 1
```

Hapus templat peluncuran

Jika Anda tidak lagi memerlukan template peluncuran, Anda dapat menghapusnya menggunakan [delete-launch-template](#) perintah berikut. Menghapus templat peluncuran akan menghapus semua versinya.

```
aws ec2 delete-launch-template --launch-template-id lt-068f72b729example
```

Memperbarui grup Auto Scaling untuk menggunakan template peluncuran

Anda dapat menggunakan [update-auto-scaling-group](#) perintah untuk menambahkan template peluncuran ke grup Auto Scaling yang ada.

Memperbarui grup Auto Scaling untuk menggunakan versi terbaru dari template peluncuran

[update-auto-scaling-group](#) Perintah berikut memperbarui grup Auto Scaling yang ditentukan untuk menggunakan versi terbaru dari template peluncuran yang ditentukan.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
--launch-template LaunchTemplateId=lt-068f72b729example,Version='$Latest'
```

Memperbarui grup Auto Scaling untuk menggunakan versi tertentu dari template peluncuran

[update-auto-scaling-group](#) Perintah berikut memperbarui grup Auto Scaling yang ditentukan untuk menggunakan versi tertentu dari template peluncuran yang ditentukan.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
--launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

Gunakan AWS Systems Manager parameter alih-alih ID AMI di templat peluncuran

Bagian ini menunjukkan cara membuat template peluncuran yang menentukan AWS Systems Manager parameter yang mereferensikan ID Amazon Machine Image (AMI). Anda dapat menggunakan parameter yang disimpan dalam parameter yang sama Akun AWS, parameter yang dibagikan dari yang lain Akun AWS, atau parameter publik untuk AMI publik yang dikelola oleh AWS.

Dengan parameter Systems Manager, Anda dapat memperbarui grup Auto Scaling untuk menggunakan ID AMI baru tanpa perlu membuat templat peluncuran baru atau versi baru templat peluncuran setiap kali ID AMI berubah. ID ini dapat berubah secara teratur, seperti ketika AMI diperbarui dengan sistem operasi terbaru atau pembaruan perangkat lunak.

Anda dapat membuat, memperbarui, atau menghapus parameter Systems Manager Anda sendiri menggunakan [Parameter Store, kemampuan dari AWS Systems Manager](#). Anda harus membuat parameter Systems Manager sebelum dapat menggunakannya dalam template peluncuran. Untuk memulai, buat parameter dengan tipe `dataaws:ec2:image`, dan untuk nilainya, masukkan ID AMI. ID AMI memiliki formulir `ami-<identifier>`, misalnya, `ami-123example456`. ID AMI yang benar bergantung pada jenis instans dan Wilayah AWS tempat Anda meluncurkan grup Auto Scaling.

Untuk informasi selengkapnya tentang membuat parameter yang valid untuk ID AMI, lihat [Membuat parameter Systems Manager](#).

Buat template peluncuran yang menentukan parameter untuk AMI

Untuk membuat template peluncuran yang menentukan parameter untuk AMI, gunakan salah satu metode berikut:

Console

Untuk membuat template peluncuran menggunakan AWS Systems Manager parameter

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Di panel navigasi, pilih Templat Peluncuran, lalu pilih Buat templat peluncuran.
3. Untuk Nama templat peluncuran, masukkan nama deskriptif untuk templat peluncuran.
4. Di bawah Gambar Aplikasi dan OS (Amazon Machine Image), pilih Telusuri AMI lainnya.
5. Pilih tombol panah di sebelah kanan bilah pencarian, lalu pilih Tentukan berdasarkan nilai kustom/parameter Systems Manager.
6. Di kotak dialog Tentukan nilai kustom atau parameter Systems Manager, lakukan hal berikut:
 - a. Untuk String parameter ID AMI atau Systems Manager, masukkan nama parameter Systems Manager menggunakan salah satu format berikut:

Untuk mereferensikan parameter publik:

- **resolve:ssm:*public-parameter***

Untuk mereferensikan parameter yang disimpan di akun yang sama:

- **resolve:ssm:*parameter-name***
- **resolve:ssm:*parameter-name:version-number***
- **resolve:ssm:*parameter-name:label***

Untuk mereferensikan parameter yang dibagikan dari yang lain Akun AWS:

- **resolve:ssm:*parameter-ARN***
- **resolve:ssm:*parameter-ARN:version-number***
- **resolve:ssm:*parameter-ARN:label***

- b. Pilih Simpan.

7. Konfigurasi pengaturan template peluncuran lainnya sesuai kebutuhan, lalu pilih Buat templat peluncuran. Untuk informasi selengkapnya, lihat [Membuat template peluncuran untuk grup Auto Scaling](#).

AWS CLI

Untuk membuat template peluncuran yang menentukan parameter Systems Manager, Anda dapat menggunakan salah satu perintah contoh berikut. Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri.

Contoh: Buat template peluncuran yang menentukan parameter publik yang AWS dimiliki

Gunakan sintaks berikut: `resolve:ssm:public-parameter`, di mana `resolve:ssm` awalan standar dan `public-parameter` merupakan jalur dan nama parameter publik.

Dalam contoh ini, template peluncuran menggunakan parameter publik AWS yang disediakan untuk meluncurkan instance menggunakan Amazon Linux 2 AMI terbaru di Wilayah AWS yang dikonfigurasi untuk profil Anda.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling
--version-description version1 \
--launch-template-data file://config.json
```

Isi dari `config.json`:

```
{
  "ImageId": "resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-
x86_64-gp2",
  "InstanceType": "t2.micro"
}
```

Berikut ini adalah contoh respons.

```
{
  "LaunchTemplate": {
    "LaunchTemplateId": "lt-089c023a30example",
    "LaunchTemplateName": "my-template-for-auto-scaling",
    "CreateTime": "2022-12-28T19:52:27.000Z",
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "DefaultVersionNumber": 1,
    "LatestVersionNumber": 1
  }
}
```

Contoh: Buat template peluncuran yang menentukan parameter yang disimpan di akun yang sama

Gunakan sintaks berikut: `resolve:ssm:parameter-name`, di mana `resolve:ssm` awalan standar dan *parameter-name* merupakan nama parameter Systems Manager.

Contoh berikut membuat template peluncuran yang mendapatkan ID AMI dari parameter Systems Manager yang ada bernama *golden-ami*.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling \
  --launch-template-data file://config.json
```

Isi dari `config.json`:

```
{
  "ImageId": "resolve:ssm:golden-ami",
  "InstanceType": "t2.micro"
}
```

Versi default parameter, jika tidak ditentukan, adalah versi terbaru.

Contoh berikut mereferensikan versi *golden-ami* parameter tertentu. Contoh menggunakan *3* versi *golden-ami* parameter, tetapi Anda dapat menggunakan nomor versi yang valid.

```
{
  "ImageId": "resolve:ssm:golden-ami:3",
  "InstanceType": "t2.micro"
}
```

Contoh serupa berikut mereferensikan label *prod* parameter yang memetakan ke versi *golden-ami* parameter tertentu.

```
{
  "ImageId": "resolve:ssm:golden-ami:prod",
  "InstanceType": "t2.micro"
}
```

Berikut ini adalah output contoh.

```
{
  "LaunchTemplate": {
    "LaunchTemplateId": "lt-068f72b724example",

```

```

    "LaunchTemplateName": "my-template-for-auto-scaling",
    "CreateTime": "2022-12-27T17:11:21.000Z",
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "DefaultVersionNumber": 1,
    "LatestVersionNumber": 1
  }
}

```

Contoh: Buat template peluncuran yang menentukan parameter yang dibagikan dari yang lain Akun AWS

Gunakan sintaks berikut: `resolve:ssm:parameter-ARN`, di `resolve:ssm` mana awalan standar dan *parameter-ARN* ARN dari parameter Systems Manager.

Contoh berikut membuat template peluncuran yang mendapatkan ID AMI dari parameter Systems Manager yang ada dengan ARN. *arn:aws:ssm:us-east-2:123456789012:parameter/MyParameter*

```

aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling
--version-description version1 \
--launch-template-data file://config.json

```

Isi dari `config.json`:

```

{
  "ImageId": "resolve:ssm:arn:aws:ssm:us-east-2:123456789012:parameter/MyParameter",
  "InstanceType": "t2.micro"
}

```

Versi default parameter, jika tidak ditentukan, adalah versi terbaru.

Contoh berikut mereferensikan versi *MyParameter* parameter tertentu. Contoh menggunakan *3* versi *MyParameter* parameter, tetapi Anda dapat menggunakan nomor versi yang valid.

```

{
  "ImageId": "resolve:ssm:arn:aws:ssm:us-east-2:123456789012:parameter/MyParameter:3",
  "InstanceType": "t2.micro"
}

```

Contoh serupa berikut mereferensikan label *prod* parameter yang memetakan ke versi *MyParameter:prod* parameter tertentu.

```
{
  "ImageId": "resolve:ssm:arn:aws:ssm:us-east-2:123456789012:parameter/
  MyParameter:prod",
  "InstanceType": "t2.micro"
}
```

Berikut ini adalah contoh respons.

```
{
  "LaunchTemplate": {
    "LaunchTemplateId": "lt-00f93d4588example",
    "LaunchTemplateName": "my-template-for-auto-scaling",
    "CreateTime": "2024-01-08T12:43:21.000Z",
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "DefaultVersionNumber": 1,
    "LatestVersionNumber": 1
  }
}
```

Untuk menentukan parameter dari Parameter Store dalam template peluncuran, Anda harus memiliki `ssm:GetParameters` izin untuk parameter yang ditentukan. Siapa pun yang menggunakan template peluncuran juga memerlukan `ssm:GetParameters` izin agar nilai parameter divalidasi. Untuk informasi selengkapnya, lihat [Membatasi akses ke parameter Systems Manager menggunakan kebijakan IAM](#) di AWS Systems Manager Panduan Pengguna.

Verifikasi template peluncuran mendapatkan ID AMI yang benar

Gunakan [describe-launch-template-versions](#) perintah dan sertakan `--resolve-alias` opsi untuk menyelesaikan parameter ke ID AMI yang sebenarnya.

```
aws ec2 describe-launch-template-versions --launch-template-name my-template-for-auto-
scaling \
  --versions $Default --resolve-alias
```

Contoh mengembalikan ID AMI untuk `ImageId`. Saat instans diluncurkan menggunakan templat peluncuran ini, ID AMI akan diselesaikan ke `ami-0ac394d6a3example`.


```
{
  "LaunchTemplateVersions": [
    {
      "LaunchTemplateId": "lt-089c023a30example",
      "LaunchTemplateName": "my-template-for-auto-scaling",
      "VersionNumber": 1,
      "CreateTime": "2022-12-28T19:52:27.000Z",
      "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
      "DefaultVersion": true,
      "LaunchTemplateData": {
        "ImageId": "ami-0ac394d6a3example",
        "InstanceType": "t2.micro",
      }
    }
  ]
}
```

Sumber daya terkait

Untuk detail selengkapnya tentang menentukan parameter Systems Manager di template peluncuran, lihat [Menggunakan parameter Systems Manager, bukan ID AMI](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk informasi selengkapnya tentang bekerja dengan parameter Systems Manager, lihat materi referensi berikut dalam dokumentasi Systems Manager.

- Untuk membuat versi dan label parameter, lihat [Bekerja dengan versi parameter](#) dan [Bekerja dengan label parameter](#).
- Untuk informasi tentang cara mencari parameter publik AMI yang didukung oleh Amazon EC2, lihat Memanggil parameter [publik AMI](#).
- Untuk informasi tentang berbagi parameter dengan AWS akun lain atau melalui AWS Organizations, lihat [Bekerja dengan parameter bersama](#).
- Untuk informasi tentang pemantauan apakah parameter berhasil dibuat, lihat [Dukungan parameter asli untuk ID Gambar Mesin Amazon](#).

Batasan

Saat bekerja dengan parameter Systems Manager, perhatikan batasan berikut:

- Auto Scaling Amazon EC2 hanya mendukung penetapan ID AMI sebagai parameter.
- Membuat atau memperbarui [grup instance campuran](#) menggunakan template peluncuran yang menentukan parameter Systems Manager saat ini tidak didukung.
- Jika grup Auto Scaling menggunakan template peluncuran yang menentukan parameter Systems Manager, Anda tidak akan dapat memulai penyegaran instans dengan konfigurasi yang diinginkan atau menggunakan pencocokan lewati.
- Pada setiap panggilan untuk membuat atau memperbarui grup Auto Scaling, Auto Scaling Amazon EC2 akan menyelesaikan parameter Systems Manager di template peluncuran. Jika Anda menggunakan parameter lanjutan atau batas throughput yang lebih tinggi, panggilan yang sering ke Parameter Store (yaitu, `GetParameters` operasi) dapat meningkatkan biaya untuk Systems Manager karena biaya dikeluarkan per interaksi Parameter Store API. Untuk informasi lebih lanjut, lihat [Harga AWS Systems Manager](#).

Konfigurasi peluncuran

Important

Anda tidak dapat menelepon `CreateLaunchConfiguration` dengan jenis instans Amazon EC2 baru yang dirilis setelah 31 Desember 2022. Selain itu, akun baru apa pun yang dibuat pada atau setelah 1 Juni 2023 tidak akan memiliki opsi untuk membuat konfigurasi peluncuran baru melalui konsol. Di masa depan, akun baru tidak akan dapat membuat konfigurasi peluncuran baru dengan menggunakan konsol, API, CLI, dan CloudFormation Migrasi untuk meluncurkan template untuk memastikan bahwa Anda tidak perlu membuat konfigurasi peluncuran baru sekarang atau di masa mendatang. Untuk informasi tentang memigrasi grup Auto Scaling untuk meluncurkan templat, lihat [Migrasi grup Auto Scaling Anda untuk meluncurkan template](#)

Konfigurasi peluncuran adalah templat konfigurasi instans yang digunakan oleh grup Auto Scaling untuk meluncurkan instans EC2. Saat Anda membuat konfigurasi peluncuran, Anda menentukan informasi untuk instans. Sertakan ID Amazon Machine Image (AMI), jenis instans, pasangan kunci, satu grup keamanan atau lebih, dan pemetaan perangkat blok. Jika Anda telah meluncurkan satu instans EC2 sebelumnya, Anda menentukan informasi yang sama untuk meluncurkan instans.

Anda dapat menentukan konfigurasi peluncuran Anda dengan beberapa grup Auto Scaling. Namun, Anda hanya dapat menentukan satu konfigurasi peluncuran untuk grup Auto Scaling dalam satu waktu, dan Anda tidak dapat memodifikasi konfigurasi peluncuran setelah Anda membuatnya. Untuk mengubah konfigurasi peluncuran untuk grup Auto Scaling, Anda harus membuat konfigurasi peluncuran lalu memperbarui grup Auto Scaling Anda dengannya.

Daftar Isi

- [Buat konfigurasi peluncuran](#)
- [Mengubah konfigurasi peluncuran untuk grup Auto Scaling](#)

Buat konfigurasi peluncuran

Important

Anda tidak dapat menelepon `CreateLaunchConfiguration` dengan jenis instans Amazon EC2 baru yang dirilis setelah 31 Desember 2022. Selain itu, akun baru apa pun yang dibuat pada atau setelah 1 Juni 2023 tidak akan memiliki opsi untuk membuat konfigurasi peluncuran baru melalui konsol. Di masa depan, akun baru tidak akan dapat membuat konfigurasi peluncuran baru dengan menggunakan konsol, API, CLI, dan CloudFormation Migrasi untuk meluncurkan template untuk memastikan bahwa Anda tidak perlu membuat konfigurasi peluncuran baru sekarang atau di masa mendatang. Untuk informasi tentang memigrasi grup Auto Scaling untuk meluncurkan templat, lihat [Migrasi grup Auto Scaling Anda untuk meluncurkan template](#)

Topik ini menjelaskan cara membuat konfigurasi peluncuran.

Setelah Anda membuat konfigurasi peluncuran, Anda tidak dapat memodifikasinya. Sebagai gantinya, Anda harus membuat konfigurasi peluncuran baru.

Untuk mengaitkan konfigurasi peluncuran baru dengan grup Auto Scaling yang ada, lihat [Mengubah konfigurasi peluncuran untuk grup Auto Scaling](#) Untuk membuat grup Auto Scaling baru, lihat [Buat grup Auto Scaling menggunakan konfigurasi peluncuran](#)

Daftar Isi

- [Buat konfigurasi peluncuran](#)
- [Mengonfigurasi opsi metadata instans](#)
- [Buat konfigurasi peluncuran menggunakan instans EC2](#)

Buat konfigurasi peluncuran


Untuk membuat konfigurasi peluncuran (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Di bilah navigasi atas, pilih AWS Wilayah Anda.
3. Di panel navigasi kiri, di bawah Auto Scaling, pilih Auto Scaling Groups.

4. Pilih Luncurkan konfigurasi di dekat bagian atas halaman. Saat diminta konfirmasi, pilih Lihat konfigurasi peluncuran untuk mengonfirmasi bahwa Anda ingin melihat halaman Konfigurasi peluncuran.
5. Pilih Buat konfigurasi peluncuran, dan masukkan nama untuk konfigurasi peluncuran Anda.
6. Untuk Amazon machine image (AMI), pilih AMI. Untuk menemukan AMI spesifik, Anda dapat [temukan AMI yang cocok](#), mencatat ID-nya, dan memasukkan ID sebagai kriteria pencarian.

Untuk mendapatkan ID Amazon Linux 2 AMI:

- a. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
 - b. Di panel navigasi kiri, di bawah Instans, pilih Instans, lalu pilih Luncurkan instance.
 - c. Di tab Mulai Cepat dari Pilih Amazon Machine Image , catat ID AMI di samping Amazon Linux 2 AMI (HVM).
7. Untuk Jenis instans, pilih konfigurasi perangkat keras untuk instans Anda.
 8. Pada Konfigurasi tambahan, perhatikan kolom-kolom berikut:
 - a. (Opsional) Untuk Opsi pembelian, Anda dapat memilih Meminta Spot Instance untuk meminta Spot Instances dengan harga Spot, dengan harga Sesuai Permintaan. Atau, Anda dapat menentukan harga maksimum per jam instans untuk Spot Instance Anda.

 Note


Spot Instances adalah pilihan hemat biaya dibandingkan dengan Instans Sesuai Permintaan, jika Anda dapat bersikap fleksibel saat aplikasi berjalan dan jika aplikasi Anda dapat terganggu. Untuk informasi selengkapnya, lihat [Minta Instans Spot untuk aplikasi yang toleran terhadap kesalahan dan fleksibel](#).

- b. (Opsional) Untuk Profil instans IAM, pilih peran untuk dikaitkan dengan instans. Untuk informasi selengkapnya, lihat [Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2](#).
- c. (Opsional) Untuk Pemantauan, pilih apakah akan mengaktifkan instans untuk mempublikasikan data metrik pada interval 1 menit ke Amazon CloudWatch dengan mengaktifkan pemantauan terperinci. Biaya tambahan berlaku. Untuk informasi selengkapnya, lihat [Konfigurasi pemantauan untuk instans Auto Scaling](#).

- d. (Opsional) Untuk Detail lanjutan, Data pengguna, Anda dapat menentukan data pengguna untuk mengonfigurasi instans selama peluncuran, atau untuk menjalankan skrip konfigurasi setelah instans dimulai.
 - e. (Opsional) Untuk Detail lanjutan, Jenis alamat IP, pilih apakah akan menugaskan [alamat IP publik](#) ke instans grup. Jika Anda tidak menetapkan nilai, defaultnya adalah menggunakan pengaturan IP publik yang ditetapkan otomatis di subnet tempat instans diluncurkan.
9. (Opsional) Untuk Penyimpanan (volume), jika Anda tidak memerlukan penyimpanan tambahan, Anda dapat melewati bagian ini. Atau, untuk menentukan volume untuk dilampirkan ke instans selain volume yang ditentukan oleh AMI, pilih Tambahkan volume baru. Kemudian pilih opsi yang diinginkan dan nilai terkait untuk Perangkat, Snapshot, Ukuran, Jenis volume, IOPS, Throughput, Hapus saat pengakhiran, dan Dienkripsi.
 10. Untuk Grup keamanan, buat atau pilih grup keamanan untuk dikaitkan dengan instans kelompok. Jika Anda membiarkan Buat grup keamanan baru tetap dipilih, aturan SSH default dikonfigurasi untuk instans Amazon EC2 yang menjalankan Linux. Aturan RDP default dikonfigurasi untuk instance Amazon EC2 yang menjalankan Windows.
 11. Untuk Pasangan kunci (login), pilih opsi pada Opsi pasangan kunci.

Jika Anda sudah mengonfigurasi pasangan kunci instance Amazon EC2, Anda dapat memilihnya di sini.

Jika Anda belum memiliki pasangan kunci instance Amazon EC2, pilih Buat pasangan kunci baru dan berikan nama yang dapat dikenali. Pilih Unduh pasangan kunci untuk mengunduh pasangan kunci ke komputer Anda.

 Important

Jika Anda perlu terhubung ke instance, jangan pilih Lanjutkan tanpa pasangan kunci.

12. Centang kotak konfirmasi, lalu pilih Buat konfigurasi peluncuran.

Untuk membuat konfigurasi peluncuran dari konfigurasi peluncuran yang ada (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Di bilah navigasi atas, pilih AWS Wilayah Anda.
3. Di panel navigasi kiri, di bawah Auto Scaling, pilih Auto Scaling Groups.

4. Pilih Luncurkan konfigurasi di dekat bagian atas halaman. Saat diminta konfirmasi, pilih Lihat konfigurasi peluncuran untuk mengonfirmasi bahwa Anda ingin melihat halaman Konfigurasi peluncuran.
5. Pilih konfigurasi peluncuran dan pilih Tindakan, Salin konfigurasi peluncuran. Ini akan mengatur konfigurasi peluncuran baru dengan opsi yang sama seperti aslinya, tetapi dengan “Salin” ditambahkan ke namanya.
6. Di halaman Salin Konfigurasi Peluncuran, edit opsi konfigurasi sesuai kebutuhan dan pilih Buat konfigurasi peluncuran.

Untuk membuat konfigurasi peluncuran menggunakan baris perintah

Anda dapat menggunakan salah satu perintah berikut:

- [create-launch-configuration](#) (AWS CLI)
- [Baru-AS \(LaunchConfiguration\)](#) AWS Tools for Windows PowerShell

Mengonfigurasi opsi metadata instans

Amazon EC2 Auto Scaling mendukung konfigurasi Instance Metadata Service (IMDS) dalam konfigurasi peluncuran. Ini memberi Anda pilihan untuk menggunakan konfigurasi peluncuran untuk mengonfigurasi instans Amazon EC2 dalam grup Auto Scaling Anda untuk mewajibkan Instance Metadata Service Versi 2 (IMDSv2), yang merupakan metode berorientasi sesi untuk meminta metadata instans. Untuk detail tentang keunggulan IMDSv2, lihat artikel ini di AWS Blog tentang [penyempurnaan untuk menambahkan pertahanan secara mendalam ke layanan metadata instans EC2](#).

Anda dapat mengonfigurasi IMDS untuk mendukung IMDSv2 dan IMDSv1 (default), atau memerlukan penggunaan IMDSv2. Jika Anda menggunakan AWS CLI atau salah satu SDK untuk mengonfigurasi IMDS, Anda harus menggunakan versi terbaru AWS CLI atau SDK untuk meminta penggunaan IMDSv2.

Anda dapat mengonfigurasi konfigurasi peluncuran untuk hal berikut:

- Mewajibkan penggunaan IMDSv2 saat meminta metadata instans
- Tentukan batas hop tanggapan PUT
- Nonaktifkan akses untuk metadata instans

Anda dapat menemukan detail selengkapnya tentang mengonfigurasi Layanan Metadata Instans dalam topik berikut: [Mengonfigurasi layanan metadata instans di Panduan Pengguna Amazon EC2 untuk Instans Linux](#).

Gunakan prosedur berikut untuk mengonfigurasi opsi IMDS dalam konfigurasi peluncuran. Setelah Anda membuat konfigurasi peluncuran, Anda dapat mengaitkannya dengan grup Auto Scaling Anda. Jika Anda menghubungkan konfigurasi peluncuran dengan grup Auto Scaling yang ada, konfigurasi peluncuran yang ada tidak dikaitkan dengan grup Auto Scaling, dan instans yang ada akan memerlukan penggantian untuk menggunakan opsi IMDS yang Anda tentukan dalam konfigurasi peluncuran baru. Untuk informasi selengkapnya, lihat [Mengubah konfigurasi peluncuran untuk grup Auto Scaling](#).

Untuk mengkonfigurasi IMDS dalam konfigurasi peluncuran (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Di bilah navigasi atas, pilih AWS Wilayah Anda.
3. Di panel navigasi kiri, di bawah Auto Scaling, pilih Auto Scaling Groups.
4. Pilih Luncurkan konfigurasi di dekat bagian atas halaman. Saat diminta konfirmasi, pilih Lihat konfigurasi peluncuran untuk mengonfirmasi bahwa Anda ingin melihat halaman Konfigurasi peluncuran.
5. Pilih Buat konfigurasi peluncuran, dan buat konfigurasi peluncuran dengan cara yang biasa. Sertakan ID Amazon Machine Image (AMI), jenis instans, dan secara opsional, pasangan kunci, satu grup keamanan atau lebih, dan volume EBS tambahan atau volume penyimpanan instans untuk instans Anda.
6. Untuk mengonfigurasi opsi metadata instans untuk semua instans yang terkait dengan konfigurasi peluncuran ini, di Konfigurasi tambahan, di bawah Detail lanjutan, lakukan hal berikut:
 - a. Untuk Metadata dapat diakses, pilih apakah akan mengaktifkan atau menonaktifkan akses ke titik akhir HTTP dari layanan metadata instans. Secara default, titik akhir HTTP diaktifkan. Jika Anda memilih untuk menonaktifkan titik akhir, akses ke metadata instans Anda dimatikan. Anda dapat menentukan kondisi untuk memerlukan ImDSv2 hanya ketika titik akhir HTTP diaktifkan.
 - b. Untuk Versi metadata, Anda dapat memilih untuk mewajibkan penggunaan Instance Metadata Service Version 2 (IMDSv2) saat meminta metadata instans. Jika Anda tidak menentukan nilai, defaultnya adalah mendukung IMDSv1 dan IMDSv2.
 - c. Untuk Batas hop respons token metadata, Anda dapat mengatur jumlah hop jaringan yang diizinkan untuk token metadata. Jika Anda tidak menentukan nilai, defaultnya 1.

7. Setelah selesai, pilih Buat konfigurasi peluncuran.

Untuk meminta penggunaan IMDSv2 dalam konfigurasi peluncuran menggunakan AWS CLI

Gunakan [create-launch-configuration](#) perintah berikut dengan `--metadata-options` set ke `HttpTokens=required`. Saat Anda menentukan nilai untuk `HttpTokens`, Anda juga harus mengatur `HttpEndpoint` ke diaktifkan. Karena header token aman disetel ke `required` untuk permintaan pengambilan metadata, ini memilih instance untuk menggunakan `ImDSv2` saat meminta metadata instance.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc-with-imdsv2 \  
  --image-id ami-01e24be29428c15b2 \  
  --instance-type t2.micro \  
  ...  
  --metadata-options "HttpEndpoint=enabled,HttpTokens=required"
```

Untuk menonaktifkan akses untuk metadata instans

Gunakan [create-launch-configuration](#) perintah berikut untuk mematikan akses ke metadata instance. Anda dapat mengaktifkan kembali akses nanti dengan menggunakan [modify-instance-metadata-options](#) perintah.

```
aws autoscaling create-launch-configuration \  
  --launch-configuration-name my-lc-with-ims-disabled \  
  --image-id ami-01e24be29428c15b2 \  
  --instance-type t2.micro \  
  ...  
  --metadata-options "HttpEndpoint=disabled"
```

Buat konfigurasi peluncuran menggunakan instans EC2

Anda juga memiliki opsi untuk membuat konfigurasi peluncuran menggunakan atribut dari instans EC2 yang sedang berjalan.

Terdapat perbedaan antara membuat konfigurasi peluncuran dari awal dan membuat konfigurasi peluncuran dari instans EC2 yang sudah ada. Saat Anda membuat konfigurasi peluncuran dari awal, Anda menentukan ID gambar, jenis instans, sumber daya opsional (misalnya perangkat penyimpanan), dan pengaturan opsional (misalnya pemantauan). Saat Anda membuat konfigurasi

peluncuran dari instans yang berjalan, Amazon EC2 Auto Scaling memperoleh atribut untuk konfigurasi peluncuran dari instans yang ditentukan. Atribut juga berasal dari pemetaan perangkat blok untuk AMI dari tempat instans diluncurkan, mengabaikan perangkat blok tambahan yang ditambahkan setelah peluncuran.

Saat membuat konfigurasi peluncuran menggunakan instance yang sedang berjalan, Anda dapat mengganti atribut berikut dengan menentukannya sebagai bagian dari permintaan yang sama: AMI, block devices, key pair, instance profile, instance type, kernel, instance monitoring, placement tenancy, ramdisk, security groups, Spot (max) price, user data, apakah instans memiliki alamat IP publik, dan apakah instance tersebut dioptimalkan EBS.

Note

Jika instans yang ditentukan memiliki properti yang saat ini tidak didukung oleh konfigurasi peluncuran, instans yang diluncurkan oleh grup Auto Scaling mungkin tidak identik dengan instans EC2 asli.

Important

AMI yang digunakan untuk meluncurkan instans yang ditentukan tetap harus ada.

Topik

- [Buat konfigurasi peluncuran dari instans EC2 \(AWS CLI\)](#)
- [Buat konfigurasi peluncuran dari sebuah instance dan ganti perangkat blok \(AWS CLI\)](#)
- [Buat konfigurasi peluncuran dan ganti jenis instance \(AWS CLI\)](#)

Buat konfigurasi peluncuran dari instans EC2 (AWS CLI)

Gunakan [create-launch-configuration](#) perintah berikut untuk membuat konfigurasi peluncuran dari sebuah instance menggunakan atribut yang sama dengan instance. Perangkat blok yang ditambahkan setelah peluncuran akan diabaikan.

```
aws autoscaling create-launch-configuration --launch-configuration-name my-lc-from-instance --instance-id i-a8e09d9c
```

Anda dapat menggunakan [describe-launch-configurations](#) perintah berikut untuk menjelaskan konfigurasi peluncuran dan memverifikasi bahwa atributnya cocok dengan atribut instance.

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-lc-from-instance
```

Berikut ini adalah contoh respons.

```
{
  "LaunchConfigurations": [
    {
      "UserData": null,
      "EbsOptimized": false,
      "LaunchConfigurationARN": "arn",
      "InstanceMonitoring": {
        "Enabled": false
      },
      "ImageId": "ami-05355a6c",
      "CreatedTime": "2014-12-29T16:14:50.382Z",
      "BlockDeviceMappings": [],
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        "sg-8422d1eb"
      ],
      "LaunchConfigurationName": "my-lc-from-instance",
      "KernelId": "null",
      "RamdiskId": null,
      "InstanceType": "t1.micro",
      "AssociatePublicIpAddress": true
    }
  ]
}
```

Buat konfigurasi peluncuran dari sebuah instance dan ganti perangkat blok ()AWS CLI

Secara default, Amazon EC2 Auto Scaling menggunakan atribut dari instans EC2 yang Anda tetapkan untuk membuat konfigurasi peluncuran. Namun, perangkat blok berasal dari AMI yang digunakan untuk meluncurkan instans, bukan instans itu sendiri. Untuk menambahkan perangkat blok ke konfigurasi peluncuran, timpa pemetaan perangkat blok untuk konfigurasi peluncuran.

Gunakan [create-launch-configuration](#) perintah berikut untuk membuat konfigurasi peluncuran menggunakan instans EC2 tetapi dengan pemetaan perangkat blok khusus.

```
aws autoscaling create-launch-configuration --launch-configuration-name my-lc-from-instance-bdm --instance-id i-a8e09d9c \
  --block-device-mappings "[{\"DeviceName\":\"/dev/sda1\",\"Ebs\":{\"SnapshotId\":\"snap-3decf207\"}}, {\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"SnapshotId\":\"snap-eed6ac86\"}}]"
```

Gunakan [describe-launch-configurations](#) perintah berikut untuk menjelaskan konfigurasi peluncuran dan verifikasi bahwa itu menggunakan pemetaan perangkat blok kustom Anda.

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-lc-from-instance-bdm
```

Contoh respons berikut menjelaskan konfigurasi peluncuran.

```
{
  "LaunchConfigurations": [
    {
      "UserData": null,
      "EbsOptimized": false,
      "LaunchConfigurationARN": "arn",
      "InstanceMonitoring": {
        "Enabled": false
      },
      "ImageId": "ami-c49c0dac",
      "CreatedTime": "2015-01-07T14:51:26.065Z",
      "BlockDeviceMappings": [
        {
          "DeviceName": "/dev/sda1",
          "Ebs": {
            "SnapshotId": "snap-3decf207"
          }
        },
        {
          "DeviceName": "/dev/sdf",
          "Ebs": {
            "SnapshotId": "snap-eed6ac86"
          }
        }
      ],
      "KeyName": "my-key-pair",
      "SecurityGroups": [
        "sg-8637d3e3"
      ]
    }
  ]
}
```

```

    ],
    "LaunchConfigurationName": "my-lc-from-instance-bdm",
    "KernelId": null,
    "RamdiskId": null,
    "InstanceType": "t1.micro",
    "AssociatePublicIpAddress": true
  }
]
}

```

Buat konfigurasi peluncuran dan ganti jenis instance ()AWS CLI

Secara default, Amazon EC2 Auto Scaling menggunakan atribut dari instans EC2 yang Anda tetapkan untuk membuat konfigurasi peluncuran. Bergantung pada persyaratan Anda, Anda mungkin ingin menimpa atribut dari instans tersebut dan menggunakan nilai yang Anda butuhkan. Misalnya, Anda dapat menimpa jenis instans.

Gunakan [create-launch-configuration](#) perintah berikut untuk membuat konfigurasi peluncuran menggunakan instans EC2 tetapi dengan jenis instans yang berbeda (misalnya `t2.medium`) dari instance (misalnya `t2.micro`).

```

aws autoscaling create-launch-configuration --launch-configuration-name my-lc-from-
instance-changetype \
  --instance-id i-a8e09d9c --instance-type t2.medium

```

Gunakan [describe-launch-configurations](#) perintah berikut untuk menjelaskan konfigurasi peluncuran dan verifikasi bahwa jenis instance telah diganti.

```

aws autoscaling describe-launch-configurations --launch-configuration-names my-lc-from-
instance-changetype

```

Contoh respons berikut menjelaskan konfigurasi peluncuran.

```

{
  "LaunchConfigurations": [
    {
      "UserData": null,
      "EbsOptimized": false,
      "LaunchConfigurationARN": "arn",
      "InstanceMonitoring": {

```

```
        "Enabled": false
    },
    "ImageId": "ami-05355a6c",
    "CreatedTime": "2014-12-29T16:14:50.382Z",
    "BlockDeviceMappings": [],
    "KeyName": "my-key-pair",
    "SecurityGroups": [
        "sg-8422d1eb"
    ],
    "LaunchConfigurationName": "my-lc-from-instance-changetype",
    "KernelId": "null",
    "RamdiskId": null,
    "InstanceType": "t2.medium",
    "AssociatePublicIpAddress": true
    }
]
}
```

Mengubah konfigurasi peluncuran untuk grup Auto Scaling

Important

Kami memberikan informasi tentang konfigurasi peluncuran untuk pelanggan yang belum bermigrasi dari konfigurasi peluncuran ke templat peluncuran. Untuk informasi tentang memigrasi grup Auto Scaling untuk meluncurkan templat, lihat [Migrasi grup Auto Scaling Anda untuk meluncurkan template](#)

Topik ini menjelaskan cara mengaitkan konfigurasi peluncuran yang berbeda dengan grup Auto Scaling Anda.

Setelah Anda mengubah konfigurasi peluncuran, setiap instance baru diluncurkan menggunakan opsi konfigurasi baru, tetapi instance yang ada tidak terpengaruh. Untuk informasi selengkapnya, lihat [Perbarui instans Auto Scaling](#).

Untuk mengubah konfigurasi peluncuran untuk grup Auto Scaling (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Di panel navigasi kiri, di bawah Auto Scaling, pilih Auto Scaling Groups.
3. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

4. Di tab Detail pilih, pilih Konfigurasi Peluncuran, Edit.
5. Untuk konfigurasi Launch, pilih konfigurasi peluncuran.
6. Setelah selesai, pilih Perbarui.

Untuk mengubah konfigurasi peluncuran untuk grup Auto Scaling menggunakan baris perintah

Anda dapat menggunakan salah satu perintah berikut:

- [update-auto-scaling-group](#) (AWS CLI)
- [Perbaruan-AS \(\) AutoScalingGroup](#) AWS Tools for Windows PowerShell

Grup Auto Scaling

Note

Jika Anda baru mengenal grup Auto Scaling, kerjakan langkah-langkah dalam tutorial [Create your first Auto Scaling](#) group untuk memulai dan lihat bagaimana grup Auto Scaling merespons saat instance dalam grup berakhir.

Grup Auto Scaling berisi kumpulan instans EC2 yang diperlakukan sebagai pengelompokan logis untuk tujuan penskalaan dan manajemen otomatis. Grup Auto Scaling juga memungkinkan Anda menggunakan fitur Auto Scaling Amazon EC2 seperti penggantian pemeriksaan kesehatan dan kebijakan penskalaan. Kedua hal tersebut mempertahankan jumlah instans dalam grup Auto Scaling dan penskalaan otomatis adalah fungsi inti dari layanan Amazon EC2 Auto Scaling .

Ukuran grup Auto Scaling bergantung pada jumlah instans yang Anda tetapkan sebagai kapasitas yang diinginkan. Anda dapat menyesuaikan ukurannya untuk memenuhi permintaan, baik secara manual atau dengan menggunakan Auto Scaling.

Sebuah grup Auto Scaling dimulai dengan meluncurkan cukup instans untuk memenuhi kapasitas yang diinginkan. Hal ini mempertahankan jumlah kasus dengan melakukan pemeriksaan kesehatan berkala terhadap kasus-kasus dalam grup. Grup Auto Scaling terus mempertahankan jumlah instans tetap bahkan jika suatu instans menjadi tidak sehat. Jika suatu instans menjadi tidak sehat, grup tersebut mengakhiri instans yang tidak sehat dan meluncurkan instans lain untuk menggantikannya. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

Anda dapat menggunakan kebijakan penskalaan untuk meningkatkan atau mengurangi jumlah instans dalam grup Anda secara dinamis untuk memenuhi kondisi yang berubah. Ketika kebijakan penskalaan berlaku, grup Auto Scaling menyesuaikan kapasitas yang diinginkan dalam kelompok, antara nilai kapasitas minimum dan maksimum yang Anda tentukan, dan meluncurkan atau menghentikan instans sesuai kebutuhan. Anda juga dapat menskalakan pada jadwal. Untuk informasi selengkapnya, lihat [Pilih metode penskalaan Anda](#).

Saat membuat grup Auto Scaling, Anda dapat memilih apakah akan meluncurkan Instans Sesuai Permintaan, Instans Spot, atau keduanya. Anda dapat menentukan beberapa opsi pembelian untuk grup Auto Scaling hanya jika Anda menggunakan template peluncuran. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).

Instans Spot memberi Anda akses ke kapasitas EC2 yang tidak terpakai dengan diskon besar dibandingkan dengan harga Sesuai Permintaan. Untuk informasi lebih lanjut, lihat [Amazon EC2 Spot Instance](#). Ada perbedaan utama antara Spot Instance dan Instans Sesuai Permintaan:

- Harga untuk Spot Instance berbeda-beda berdasarkan permintaan
- Amazon EC2 dapat mengakhiri masing-masing Spot Instance sebagai ketersediaan, atau harga untuk, perubahan Spot Instance

Saat Spot Instance dihentikan, grup Auto Scaling mencoba meluncurkan instans pengganti untuk mempertahankan kapasitas yang diinginkan bagi grup.

Ketika instans diluncurkan, jika Anda menentukan beberapa Availability Zone, kapasitas yang diinginkan didistribusikan di seluruh Availability Zone. Jika tindakan penskalaan terjadi, Amazon EC2 Auto Scaling secara otomatis mempertahankan keseimbangan di semua Availability Zone yang Anda tentukan.

Daftar Isi

- [Buat grup Auto Scaling menggunakan template peluncuran](#)
- [Buat grup Auto Scaling menggunakan konfigurasi peluncuran](#)
- [Memperbarui grup Auto Scaling](#)
- [Menandai grup dan instance Auto Scaling](#)
- [Kebijakan pemeliharaan instans](#)
- [Hook siklus aktif Amazon EC2 Auto Scaling](#)
- [Kolam hangat untuk Auto Scaling Amazon EC2](#)
- [Lepaskan atau lampirkan instance](#)
- [Hapus instans sementara dari grup Auto Scaling](#)
- [Hapus infrastruktur Auto Scaling Anda](#)
- [Contoh untuk membuat dan mengelola grup Auto Scaling dengan SDK AWS](#)

Buat grup Auto Scaling menggunakan template peluncuran

Jika Anda telah membuat template peluncuran, Anda dapat membuat grup Auto Scaling yang menggunakan template peluncuran sebagai templat konfigurasi untuk instans EC2-nya. Template peluncuran menentukan informasi seperti ID AMI, tipe instans, key pair, grup keamanan, dan

pemetaan perangkat blok untuk instans Anda. Untuk informasi tentang membuat template peluncuran, lihat [Membuat template peluncuran untuk grup Auto Scaling](#) .

Anda harus memiliki izin yang cukup untuk membuat grup Auto Scaling. Anda juga harus memiliki izin yang cukup untuk membuat peran terkait layanan yang digunakan Auto Scaling Amazon EC2 untuk melakukan tindakan atas nama Anda jika belum ada. Untuk contoh kebijakan IAM yang dapat digunakan administrator sebagai referensi untuk memberikan izin kepada Anda, lihat dan. [Contoh kebijakan berbasis identitas](#) [Luncurkan dukungan templat](#)

Daftar Isi

- [Buat grup Auto Scaling menggunakan templat peluncuran](#)
- [Membuat grup Auto Scaling menggunakan wizard peluncuran Amazon EC2](#)
- [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#)

Buat grup Auto Scaling menggunakan templat peluncuran

Saat membuat grup Auto Scaling, Anda harus menentukan informasi yang diperlukan untuk mengonfigurasi instans Amazon EC2, Availability Zone, dan subnet VPC untuk instans, kapasitas yang diinginkan, dan batas kapasitas minimum dan maksimum.

Untuk mengonfigurasi instans Amazon EC2 yang diluncurkan oleh grup Auto Scaling, Anda dapat menentukan templat peluncuran atau konfigurasi peluncuran. Prosedur berikut menunjukkan cara membuat grup Auto Scaling menggunakan templat peluncuran.

Prasyarat

- Anda harus telah membuat template peluncuran. Untuk informasi selengkapnya, lihat [Membuat template peluncuran untuk grup Auto Scaling](#).

Untuk membuat grup Auto Scaling menggunakan templat peluncuran (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pada bilah navigasi di bagian atas layar, pilih yang sama dengan Wilayah AWS yang Anda gunakan saat membuat templat peluncuran.
3. Pilih Buat grup Auto Scaling.
4. Di halaman Pilih templat peluncuran atau konfigurasi, lakukan hal berikut:

- a. Untuk Nama grup Auto Scaling, masukkan nama untuk grup Auto Scaling Anda.
 - b. Untuk Templat peluncuran, pilih templat peluncuran yang ada.
 - c. Untuk Versi templat peluncuran, pilih apakah grup Auto Scaling menggunakan default, terbaru, atau versi spesifik templat peluncuran saat menskalakan.
 - d. Verifikasi bahwa templat peluncuran Anda mendukung semua opsi yang Anda rencanakan untuk digunakan, lalu pilih Selanjutnya.
5. Pada halaman Pilih opsi peluncuran instance, jika Anda tidak menggunakan beberapa jenis instans, Anda dapat melewati bagian Persyaratan tipe instans untuk menggunakan jenis instans EC2 yang ditentukan dalam template peluncuran.

Untuk menggunakan beberapa jenis instance, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).

6. Di bawah Jaringan, untuk VPC, pilih VPC. Grup Auto Scaling harus dibuat dalam VPC yang sama dengan grup keamanan yang Anda tentukan dalam template peluncuran Anda.
7. Untuk Zona Ketersediaan dan subnet, pilih satu atau lebih subnet di VPC yang ditentukan. Gunakan subnet di beberapa Availability Zone untuk ketersediaan tinggi. Untuk informasi selengkapnya, lihat [Pertimbangan saat memilih subnet VPC](#).
8. Jika Anda membuat template peluncuran dengan jenis instance yang ditentukan, Anda dapat melanjutkan ke langkah berikutnya untuk membuat grup Auto Scaling yang menggunakan jenis instance di template peluncuran.

Atau, Anda dapat memilih opsi Override launch template jika tidak ada jenis instans yang ditentukan dalam template peluncuran Anda atau jika Anda ingin menggunakan beberapa jenis instance untuk penskalaan otomatis. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).

9. Pilih Berikutnya untuk melanjutkan ke langkah berikutnya.

Atau, Anda dapat menerima pengaturan default lainnya, dan memilih Lewati untuk meninjau.

10. (Opsional) Pada Konfigurasi opsi lanjutan, konfigurasi opsi berikut, lalu pilih Selanjutnya:
 - a. Untuk mendaftarkan instans Amazon EC2 Anda dengan penyeimbang beban, pilih penyeimbang beban yang ada atau buat yang baru. Untuk informasi selengkapnya, lihat [Gunakan Elastic Load Balancing untuk mendistribusikan lalu lintas di seluruh instans dalam grup Auto Scaling](#). Untuk membuat penyeimbang beban baru, ikuti prosedur

[diKonfigurasi Application Load Balancer atau Network Load Balancer dari konsol Auto Scaling Amazon EC2.](#)

- b. (Opsional) Untuk pemeriksaan Kesehatan, Jenis pemeriksaan kesehatan tambahan, pilih Aktifkan pemeriksaan kesehatan Elastic Load Balancing.
 - c. (Opsional) Untuk Health check masa tenggang, masukkan jumlah waktu, dalam hitungan detik. Ini adalah berapa lama Amazon EC2 Auto Scaling perlu menunggu sebelum memeriksa status kesehatan suatu instans setelah memasuki status. InService Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).
 - d. Di bawah Pengaturan tambahan, Pemantauan, pilih apakah akan mengaktifkan pengumpulan metrik CloudWatch grup. Metrik ini memberikan pengukuran yang dapat menjadi indikator potensi masalah, seperti jumlah instance penghentian atau jumlah instance yang tertunda. Untuk informasi selengkapnya, lihat [Pantau CloudWatch metrik untuk grup dan instans Auto Scaling](#).
 - e. Untuk Aktifkan pemanasan instans default, pilih opsi ini dan pilih waktu pemanasan untuk aplikasi Anda. Jika Anda membuat grup Auto Scaling yang memiliki kebijakan penskalaan, fitur pemanasan instans default akan meningkatkan CloudWatch metrik Amazon yang digunakan untuk penskalaan dinamis. Untuk informasi selengkapnya, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#).
11. (Opsional) Di halaman Konfigurasi ukuran grup dan kebijakan penskalaan, konfigurasi opsi berikut, lalu pilih Selanjutnya:
- a. Di bawah Ukuran grup, untuk kapasitas yang diinginkan, masukkan jumlah instans awal yang akan diluncurkan.
 - b. Di bagian Penskalaan, di bawah batas Penskalaan, jika nilai baru Anda untuk kapasitas yang diinginkan lebih besar dari kapasitas yang diinginkan Min dan kapasitas yang diinginkan Maks, kapasitas Maks yang diinginkan secara otomatis ditingkatkan ke nilai kapasitas baru yang diinginkan. Anda dapat mengubah batasan ini sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).
 - c. Untuk Penskalaan otomatis, pilih apakah Anda ingin membuat kebijakan penskalaan pelacakan target. Anda juga dapat membuat kebijakan ini setelah membuat grup Auto Scaling.

Jika Anda memilih kebijakan penskalaan pelacakan target, ikuti petunjuk [Buat kebijakan penskalaan pelacakan target](#) untuk membuat kebijakan.

- d. Untuk kebijakan pemeliharaan Instance, pilih apakah Anda ingin membuat kebijakan pemeliharaan instans. Anda juga dapat membuat kebijakan ini setelah membuat grup Auto Scaling. Ikuti petunjuk [Tetapkan kebijakan pemeliharaan instans](#) untuk membuat kebijakan.
 - e. Di bawah Perlindungan penskalaan instans, pilih apakah akan mengaktifkan perlindungan skala instans. Untuk informasi selengkapnya, lihat [Gunakan perlindungan skala dalam instance](#).
12. (Opsional) Untuk menerima notifikasi, untuk Tambahkan notifikasi, konfigurasi notifikasi, lalu pilih Selanjutnya. Untuk informasi selengkapnya, lihat [Opsi pemberitahuan Amazon SNS untuk Auto Scaling Amazon EC2](#).
 13. (Opsional) Untuk menambahkan tag, pilih Tambahkan tag, berikan kunci tag dan nilai tag untuk setiap tag, lalu pilih Selanjutnya. Untuk informasi selengkapnya, lihat [Menandai grup dan instance Auto Scaling](#).
 14. Di halaman Peninjauan, pilih Buat grup Auto Scaling.

Untuk membuat grup Auto Scaling menggunakan jalur perintah

Anda dapat menggunakan salah satu perintah berikut:

- [create-auto-scaling-group](#) (AWS CLI)
- [Baru-AS \(AutoScalingGroup\)](#) AWS Tools for Windows PowerShell

Membuat grup Auto Scaling menggunakan wizard peluncuran Amazon EC2

Prosedur berikut menunjukkan cara membuat grup Auto Scaling menggunakan panduan Launch instance di konsol Amazon EC2. Opsi ini secara otomatis mengisi template peluncuran dengan detail konfigurasi tertentu dari wizard Launch instance.

Note

Wizard tidak mengisi grup Auto Scaling dengan jumlah instance yang Anda tentukan; itu hanya mengisi template peluncuran dengan ID Amazon Machine Image (AMI) dan jenis instans. Gunakan wizard grup Create Auto Scaling untuk menentukan jumlah instance yang akan diluncurkan.

AMI menyediakan informasi yang diperlukan untuk mengkonfigurasi instance. Anda dapat meluncurkan beberapa instans dari AMI tunggal ketika Anda memerlukan beberapa instans dengan konfigurasi yang sama. Sebaiknya gunakan AMI kustom yang sudah menginstal

aplikasi Anda di dalamnya untuk menghindari instance Anda dihentikan jika Anda me-reboot instance milik grup Auto Scaling. Untuk menggunakan AMI kustom dengan Auto Scaling Amazon EC2, Anda harus terlebih dahulu membuat AMI dari instans yang disesuaikan, lalu gunakan AMI untuk membuat template peluncuran untuk grup Auto Scaling Anda.

Prasyarat

- Anda harus telah membuat AMI kustom di tempat yang sama Wilayah AWS di mana Anda berencana untuk membuat grup Auto Scaling. Untuk informasi selengkapnya, lihat [Membuat AMI](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Gunakan AMI kustom sebagai templat

Di bagian ini, Anda menggunakan wizard peluncuran Amazon EC2 untuk secara otomatis mengisi template peluncuran dengan AMI kustom Anda. Atau, untuk mengatur template peluncuran dari awal atau untuk deskripsi lebih lanjut tentang parameter yang dapat Anda konfigurasi untuk template peluncuran Anda, lihat [Buat template peluncuran Anda \(konsol\)](#).

Untuk menggunakan AMI kustom sebagai template

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pada bilah navigasi di bagian atas layar, arus Wilayah AWS ditampilkan. Pilih Wilayah untuk meluncurkan grup Auto Scaling Anda.
3. Di panel navigasi, pilih Instans.
4. Pilih Launch instance, lalu lakukan hal berikut:
 - a. Di bawah Nama dan tag, biarkan Nama kosong. Nama ini bukan bagian dari data yang digunakan untuk membuat template peluncuran.
 - b. Di bawah Gambar Aplikasi dan OS (Gambar Mesin Amazon), pilih Jelajahi AMI lainnya untuk menelusuri katalog AMI lengkap.
 - c. Pilih AMI Saya, temukan AMI yang Anda buat, lalu pilih Pilih.
 - d. Di bawah jenis Instance, pilih jenis instance.

Note

Pilih jenis instans yang sama yang Anda gunakan saat membuat AMI atau yang lebih kuat.

- e. Di sisi kanan layar, di bawah Ringkasan, untuk Jumlah contoh, masukkan nomor apa pun. Jumlah yang Anda masukkan di sini tidak penting. Anda akan menentukan jumlah instance yang ingin Anda luncurkan saat membuat grup Auto Scaling.

Di bawah bidang Jumlah instans, pesan ditampilkan yang bertuliskan Saat meluncurkan lebih dari 1 instans, pertimbangkan EC2 Auto Scaling.

- f. Pilih teks hyperlink EC2 Auto Scaling yang dipertimbangkan.
- g. Pada dialog konfirmasi Launch into Auto Scaling Group, pilih Lanjutkan untuk membuka halaman Buat template peluncuran dengan AMI dan jenis instans yang Anda pilih di wizard instance peluncuran yang sudah diisi.

Setelah Anda memilih Lanjutkan, halaman Buat template peluncuran terbuka. Ikuti prosedur ini untuk menyelesaikan pembuatan template peluncuran.

Untuk membuat templat peluncuran

1. Di bawah Launch nama dan deskripsi template, masukkan nama dan deskripsi untuk template peluncuran baru.
2. (Opsional) Di bawah Pasangan kunci (login), untuk nama pasangan Kunci, pilih nama key pair yang dibuat sebelumnya untuk digunakan saat menghubungkan ke instance, misalnya, menggunakan SSH.
3. (Opsional) Di bawah Pengaturan jaringan, untuk grup Keamanan, pilih satu atau beberapa [grup keamanan](#) yang dibuat sebelumnya.
4. (Opsional) Di bawah Konfigurasi penyimpanan, perbarui konfigurasi penyimpanan. Konfigurasi penyimpanan default ditentukan oleh AMI dan jenis instans.
5. Setelah selesai mengonfigurasi template peluncuran, pilih Buat templat peluncuran.
6. Pada halaman konfirmasi, pilih Buat grup Auto Scaling.

Membuat grup Auto Scaling

Note

Sisa topik ini menjelaskan prosedur dasar untuk membuat grup Auto Scaling. Untuk deskripsi selengkapnya tentang parameter yang dapat Anda konfigurasi untuk grup Auto Scaling, lihat [Buat grup Auto Scaling menggunakan templat peluncuran](#)

Setelah Anda memilih grup Create Auto Scaling, wizard grup Create Auto Scaling akan terbuka. Ikuti prosedur ini untuk membuat grup Auto Scaling.

Untuk membuat grup Auto Scaling

1. Pada halaman Choose launch template or configuration, masukkan nama untuk grup Auto Scaling.
2. Template peluncuran yang Anda buat sudah dipilih untuk Anda.

Untuk Versi templat peluncuran, pilih apakah grup Auto Scaling menggunakan default, terbaru, atau versi spesifik templat peluncuran saat menskalakan.

3. Pilih Berikutnya untuk melanjutkan ke langkah berikutnya.
4. Pada halaman Pilih opsi peluncuran instance, jika Anda tidak menggunakan beberapa jenis instans, Anda dapat melewati bagian Persyaratan tipe instans untuk menggunakan jenis instans EC2 yang ditentukan dalam template peluncuran.

Untuk menggunakan beberapa jenis instance, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).

5. Di bawah Jaringan, untuk VPC, pilih VPC. Grup Auto Scaling harus dibuat dalam VPC yang sama dengan grup keamanan yang Anda tentukan dalam template peluncuran Anda.

Tip

Jika Anda tidak menentukan grup keamanan dalam template peluncuran, instance Anda akan diluncurkan dengan grup keamanan default dari VPC yang Anda tentukan. Secara default, grup keamanan ini tidak mengizinkan lalu lintas masuk dari jaringan eksternal.

6. Untuk Zona Ketersediaan dan subnet, pilih satu atau lebih subnet di VPC yang ditentukan.

7. Pilih Berikutnya dua kali untuk membuka halaman Konfigurasi ukuran grup dan kebijakan penskalaan.
8. Di bawah Ukuran grup, tentukan kapasitas yang diinginkan (jumlah awal instans yang akan diluncurkan segera setelah grup Auto Scaling dibuat).
9. Di bagian Penskalaan, di bawah batas Penskalaan, jika nilai baru Anda untuk kapasitas yang diinginkan lebih besar dari kapasitas yang diinginkan Min dan kapasitas yang diinginkan Maks, kapasitas Maks yang diinginkan secara otomatis ditingkatkan ke nilai kapasitas baru yang diinginkan. Anda dapat mengubah batasan ini sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).
10. Pilih Lewati untuk meninjau.
11. Pada halaman Peninjauan, pilih Buat grup Auto Scaling.

Langkah selanjutnya

Anda dapat memeriksa apakah grup Auto Scaling telah dibuat dengan benar dengan melihat riwayat aktivitas. Pada tab Aktivitas, di bawah Riwayat aktivitas, kolom Status menunjukkan apakah grup Auto Scaling Anda telah berhasil meluncurkan instance. Jika instans gagal diluncurkan atau diluncurkan tetapi kemudian segera dihentikan, lihat topik berikut untuk kemungkinan penyebab dan resolusi:

- [Memecahkan Masalah Auto Scaling Amazon EC2: Kegagalan peluncuran instans EC2](#)
- [Memecahkan Masalah Auto Scaling Amazon EC2: Masalah AMI](#)
- [Memecahkan Masalah Auto Scaling Amazon EC2: Pemeriksaan Kesehatan](#)

Anda sekarang dapat memasang penyeimbang beban di Wilayah yang sama dengan grup Auto Scaling Anda, jika diinginkan. Untuk informasi selengkapnya, lihat [Gunakan Elastic Load Balancing untuk mendistribusikan lalu lintas di seluruh instans dalam grup Auto Scaling](#).

Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian

Anda dapat meluncurkan dan secara otomatis menyesuaikan armada Instans Sesuai Permintaan dan Spot Instance dalam satu grup Auto Scaling. Selain menerima diskon untuk menggunakan Instans Spot, Anda dapat menggunakan Instans Cadangan atau Savings Plan untuk menerima diskon pada harga Instans Sesuai Permintaan reguler. Faktor-faktor ini membantu Anda mengoptimalkan penghematan biaya untuk instans EC2 dan mendapatkan skala dan kinerja yang diinginkan untuk aplikasi Anda.

Instans Spot adalah kapasitas cadangan yang tersedia dengan diskon besar dibandingkan dengan harga EC2 On-Demand. Instans Spot adalah pilihan hemat biaya jika Anda dapat bersikap fleksibel tentang kapan aplikasi Anda berjalan dan apakah aplikasi Anda dapat diinterupsi. Mereka dapat digunakan untuk berbagai aplikasi yang toleran terhadap kesalahan dan fleksibel. Contohnya termasuk server web stateless, titik akhir API, aplikasi data dan analitik besar, beban kerja kontainer, pipeline CI/CD, komputasi kinerja tinggi dan throughput tinggi (HPC/HTC), beban kerja rendering, dan beban kerja fleksibel lainnya.

Untuk informasi selengkapnya, lihat [Opsi pembelian instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Topik

- [Gambaran umum pengaturan](#)
- [Strategi alokasi](#)
- [Buat grup instance campuran menggunakan pemilihan tipe instans berbasis atribut](#)
- [Buat grup instance campuran dengan memilih jenis instance secara manual](#)
- [Konfigurasi grup Auto Scaling untuk menggunakan bobot instance](#)
- [Gunakan template peluncuran yang berbeda untuk jenis instans](#)

Gambaran umum pengaturan

Topik ini memberikan ikhtisar dan praktik terbaik untuk membuat grup instance campuran.

Daftar Isi

- [Gambaran Umum](#)
- [Fleksibilitas tipe instans](#)
- [Fleksibilitas Zona Ketersediaan](#)
- [Harga spot max](#)
- [Penyeimbangan kembali kapasitas proaktif](#)
- [Perilaku penskalaan](#)
- [Ketersediaan regional jenis instans](#)
- [Sumber daya terkait](#)
- [Batasan](#)

Gambaran Umum

Untuk membuat grup instance campuran, Anda memiliki dua opsi:

- [Pemilihan tipe instans berbasis atribut — Tentukan](#) persyaratan komputasi Anda untuk memilih jenis instans Anda secara otomatis berdasarkan atribut instans spesifiknya.
- [Pemilihan tipe instans manual](#) — Pilih jenis instans yang sesuai dengan beban kerja Anda secara manual.

Manual selection

Langkah-langkah berikut menjelaskan cara membuat grup instance campuran dengan memilih jenis instance secara manual:

1. Pilih template peluncuran yang memiliki parameter untuk meluncurkan instans EC2. Parameter dalam template peluncuran bersifat opsional, tetapi Auto Scaling Amazon EC2 tidak dapat meluncurkan instance jika ID Amazon Machine Image (AMI) hilang dari template peluncuran.
2. Pilih opsi untuk mengganti template peluncuran.
3. Pilih jenis instance yang sesuai dengan beban kerja Anda secara manual.
4. Tentukan persentase Instans Sesuai Permintaan dan Instans Spot yang akan diluncurkan.
5. Pilih strategi alokasi yang menentukan cara Auto Scaling Amazon EC2 memenuhi kapasitas Sesuai Permintaan dan Spot Anda dari jenis instans yang memungkinkan.
6. Pilih Availability Zones dan VPC subnet untuk meluncurkan instans Anda.
7. Tentukan ukuran awal grup (kapasitas yang diinginkan) dan ukuran minimum dan maksimum grup.

Penggantian diperlukan untuk mengganti jenis instance yang dideklarasikan dalam template peluncuran dan menggunakan beberapa jenis instance yang disematkan dalam definisi sumber daya grup Auto Scaling sendiri. Untuk informasi selengkapnya tentang jenis instans yang tersedia, lihat [Jenis instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Anda juga dapat mengonfigurasi parameter opsional berikut untuk setiap jenis instans:

- `LaunchTemplateSpecification`— Anda dapat menetapkan template peluncuran yang berbeda untuk jenis instance sesuai kebutuhan. Opsi ini saat ini tidak tersedia dari konsol. Untuk informasi selengkapnya, lihat [Gunakan template peluncuran yang berbeda untuk jenis instans](#).

- **WeightedCapacity**— Anda memutuskan berapa banyak instans diperhitungkan terhadap kapasitas yang diinginkan relatif terhadap sisa instance dalam grup Anda. Jika Anda menentukan **WeightedCapacity** nilai untuk satu jenis instance, Anda harus menentukan **WeightedCapacity** nilai untuk semuanya. Secara default, setiap instance dihitung sebagai satu terhadap kapasitas yang Anda inginkan. Untuk informasi selengkapnya, lihat [Konfigurasi grup Auto Scaling untuk menggunakan bobot instance](#).

Attribute-based selection

Agar Auto Scaling Amazon EC2 dapat memilih jenis instans Anda secara otomatis berdasarkan atribut instans spesifiknya, gunakan langkah-langkah berikut untuk membuat grup instans campuran dengan menentukan persyaratan komputasi Anda:

1. Pilih template peluncuran yang memiliki parameter untuk meluncurkan instans EC2. Parameter dalam template peluncuran bersifat opsional, tetapi Auto Scaling Amazon EC2 tidak dapat meluncurkan instance jika ID Amazon Machine Image (AMI) hilang dari template peluncuran.
2. Pilih opsi untuk mengganti template peluncuran.
3. Tentukan atribut instance yang sesuai dengan persyaratan komputasi Anda, seperti vCPU dan persyaratan memori.
4. Tentukan persentase Instans Sesuai Permintaan dan Instans Spot yang akan diluncurkan.
5. Pilih strategi alokasi yang menentukan cara Auto Scaling Amazon EC2 memenuhi kapasitas Sesuai Permintaan dan Spot Anda dari jenis instans yang memungkinkan.
6. Pilih Availability Zones dan VPC subnet untuk meluncurkan instans Anda.
7. Tentukan ukuran awal grup (kapasitas yang diinginkan) dan ukuran minimum dan maksimum grup.

Penggantian diperlukan untuk mengganti jenis instance yang dideklarasikan dalam template peluncuran dan menggunakan sekumpulan atribut instance yang menjelaskan persyaratan komputasi Anda. Untuk atribut yang didukung, lihat [InstanceRequirements](#) di Referensi API Auto Scaling Amazon EC2. Atau, Anda dapat menggunakan template peluncuran yang sudah memiliki definisi atribut instance Anda.

Anda juga dapat mengonfigurasi `LaunchTemplateSpecification` parameter dalam struktur penggantian untuk menetapkan template peluncuran yang berbeda ke serangkaian persyaratan instance sesuai kebutuhan. Opsi ini saat ini tidak tersedia dari konsol. Untuk informasi lebih lanjut, lihat [LaunchTemplateOverrides](#) dalam Referensi API Amazon EC2 Auto Scaling.

Secara default, Anda menetapkan jumlah instans sebagai kapasitas yang diinginkan dari grup Auto Scaling Anda.

Atau, Anda dapat mengatur nilai untuk kapasitas yang diinginkan ke jumlah vCPU atau jumlah memori. Untuk melakukannya, gunakan `DesiredCapacityType` properti dalam operasi `CreateAutoScalingGroup` API atau bidang tarik-turun tipe kapasitas yang diinginkan di AWS Management Console. Ini adalah alternatif yang berguna untuk [bobot contoh](#).

Fleksibilitas tipe instans

Untuk meningkatkan ketersediaan, terapkan aplikasi Anda di beberapa jenis instans. Ini adalah praktik terbaik untuk menggunakan beberapa jenis instans untuk memenuhi persyaratan kapasitas. Dengan cara ini, Auto Scaling Amazon EC2 dapat meluncurkan jenis instans lain jika kapasitas instans tidak mencukupi di Availability Zone yang Anda pilih.

Jika kapasitas instans dengan Instans Spot tidak mencukupi, Auto Scaling Amazon EC2 terus mencoba diluncurkan dari kumpulan Instans Spot lainnya. (Kumpulan yang digunakannya ditentukan oleh pilihan jenis instans dan strategi alokasi Anda.) Auto Scaling Amazon EC2 membantu Anda memanfaatkan penghematan biaya Instans Spot dengan meluncurkannya alih-alih Instans Sesuai Permintaan.

Sebaiknya fleksibel di setidaknya 10 jenis instans untuk setiap beban kerja. Saat memilih jenis instans Anda, jangan membatasi diri Anda pada jenis instans baru yang paling populer. Memilih jenis instans generasi sebelumnya cenderung menghasilkan lebih sedikit gangguan Spot karena kurang diminati oleh pelanggan On-Demand.

Fleksibilitas Zona Ketersediaan

Kami sangat menyarankan agar Anda menjangkau grup Auto Scaling Anda di beberapa Availability Zone. Dengan beberapa Availability Zone, Anda dapat merancang aplikasi yang secara otomatis gagal di antara zona untuk ketahanan yang lebih besar.

Sebagai manfaat tambahan, Anda dapat mengakses kumpulan kapasitas Amazon EC2 yang lebih dalam jika dibandingkan dengan grup dalam satu Availability Zone. Karena kapasitas berfluktuasi secara independen untuk setiap jenis instans di setiap Availability Zone, Anda seringkali bisa mendapatkan lebih banyak kapasitas komputasi dengan fleksibilitas untuk tipe instans dan Availability Zone.

Untuk informasi selengkapnya tentang menggunakan beberapa Availability Zone, lihat [Contoh: Mendistribusikan instance di seluruh Availability Zone](#).

Harga spot max

Saat membuat grup Auto Scaling menggunakan AWS CLI atau SDK, Anda dapat menentukan parameternya. `SpotMaxPrice` `SpotMaxPriceParameter` menentukan harga maksimum yang bersedia Anda bayar untuk jam Instans Spot.

Saat Anda menentukan `WeightedCapacity` parameter dalam penggantian Anda (atau `"DesiredCapacityType": "vcpu"` atau `"DesiredCapacityType": "memory-mib"` pada tingkat grup), harga maksimum mewakili harga satuan maksimum, bukan harga maksimum untuk keseluruhan instance.

Kami sangat menyarankan agar Anda tidak menentukan harga maksimum. Aplikasi Anda mungkin tidak berjalan jika Anda tidak menerima Instans Spot, seperti ketika harga maksimum Anda terlalu rendah. Jika Anda tidak menentukan harga maksimum, harga maksimum default adalah harga Sesuai Permintaan. Anda hanya membayar harga Spot untuk Spot Instance yang Anda luncurkan. Anda masih menerima diskon besar yang disediakan oleh Instans Spot. Diskon ini dimungkinkan karena harga Spot stabil yang tersedia dengan [model harga Spot](#). Untuk informasi lebih lanjut, lihat [Harga dan penghematan](#) di Panduan Pengguna Amazon EC2 untuk Instance Linux.

Penyeimbangan kembali kapasitas proaktif

Jika kasus penggunaan Anda memungkinkan, kami merekomendasikan Penyeimbangan Kembali Kapasitas. Penyeimbangan Kembali Kapasitas membantu Anda mempertahankan ketersediaan beban kerja dengan secara proaktif menambah armada Anda dengan Instans Spot baru sebelum Instans Spot yang sedang berjalan menerima pemberitahuan interupsi Instans Spot dua menit.

Saat Penyeimbangan Kembali Kapasitas diaktifkan, Auto Scaling Amazon EC2 mencoba mengganti Instans Spot secara proaktif yang telah menerima rekomendasi penyeimbangan ulang. Ini memberikan kesempatan untuk menyeimbangkan kembali beban kerja Anda ke Instans Spot baru yang tidak berisiko tinggi mengalami gangguan.

Untuk informasi selengkapnya, lihat [Gunakan Rebalancing Kapasitas untuk menangani interupsi Spot Amazon EC2](#).

Perilaku penskalaan

Saat Anda membuat grup instance campuran, grup ini menggunakan Instans Sesuai Permintaan secara default. Untuk menggunakan Instans Spot, Anda harus mengubah persentase grup yang akan diluncurkan sebagai Instans Sesuai Permintaan. Anda dapat menentukan angka berapa pun dari 0 hingga 100 untuk persentase Sesuai Permintaan.

Secara opsional, Anda juga dapat menetapkan nomor dasar Instans Sesuai Permintaan untuk memulai. Jika Anda melakukannya, Auto Scaling Amazon EC2 menunggu untuk meluncurkan Instans Spot hingga setelah meluncurkan kapasitas dasar Instans Sesuai Permintaan saat grup keluar. Apa pun di luar kapasitas dasar menggunakan persentase Sesuai Permintaan untuk menentukan berapa banyak Instans Sesuai Permintaan dan Spot Instance untuk diluncurkan.

Auto Scaling Amazon EC2 mengonversi persentase ke jumlah instans yang setara. Jika hasilnya membuat bilangan pecahan, itu membulatkan ke bilangan bulat berikutnya yang mendukung Instans On-Demand.

Tabel berikut menunjukkan perilaku grup Auto Scaling saat bertambah dan berkurang ukurannya.

Contoh: Perilaku penskalaan

Opsi pembelian Ukuran grup dan jumlah instans yang berjalan di seluruh opsi pembelian

10 20 30 40

Contoh 1: basis
10, 50/ 50% On-
Demand/Spot

On-Demand Instances (base amount)	10	10	10	10
---	----	----	----	----

On-Demand Instances	0	5	10	15
------------------------	---	---	----	----

Spot Instances	0	5	10	15
----------------	---	---	----	----

Contoh 2: basis
0, 0/ 100% On-
Demand/Spot

On-Demand Instances (base amount)	0	0	0	0
---	---	---	---	---

On-Demand Instances	0	0	0	0
------------------------	---	---	---	---

Opsi pembelian	Ukuran grup dan jumlah instans yang berjalan di seluruh opsi pembelian			
Spot Instances	10	20	30	40
Contoh 3: basis 0, 60/ 40% On-Demand/Spot				
On-Demand Instances (base amount)	0	0	0	0
On-Demand Instances	6	12	18	24
Spot Instances	4	8	12	16
Contoh 4: basis 0, 100/ 0% On-Demand/Spot				
On-Demand Instances (base amount)	0	0	0	0
On-Demand Instances	10	20	30	40
Spot Instances	0	0	0	0
Contoh 5: basis 12, 0/ 100% On-Demand/Spot				
On-Demand Instances (base amount)	10	12	12	12
On-Demand Instances	0	0	0	0

Opsi pembelian	Ukuran grup dan jumlah instans yang berjalan di seluruh opsi pembelian			
Spot Instances	0	8	18	28

Saat ukuran grup bertambah, Auto Scaling Amazon EC2 mencoba menyeimbangkan kapasitas Anda secara merata di seluruh Availability Zone yang Anda tentukan. Kemudian, ia meluncurkan jenis instance sesuai dengan strategi alokasi yang ditentukan.

Ketika ukuran grup berkurang, Auto Scaling Amazon EC2 terlebih dahulu mengidentifikasi mana dari dua jenis (Spot atau Sesuai Permintaan) yang harus dihentikan. Kemudian, ia mencoba untuk menghentikan instance secara seimbang di seluruh Availability Zone yang Anda tentukan. Ini juga mendukung penghentian instance dengan cara yang sejalan lebih dekat dengan strategi alokasi Anda. Untuk informasi tentang kebijakan penghentian, lihat [Mengonfigurasi kebijakan penghentian untuk Auto Scaling Amazon EC2](#).

Ketersediaan regional jenis instans

Ketersediaan jenis instans EC2 bervariasi tergantung pada Anda Wilayah AWS. Misalnya, jenis instance generasi terbaru mungkin belum tersedia di Wilayah tertentu. Karena varians ketersediaan instans di seluruh Wilayah, Anda mungkin mengalami masalah saat membuat permintaan terprogram jika beberapa jenis instans dalam penggantian Anda tidak tersedia di Wilayah Anda. Menggunakan beberapa jenis instans yang tidak tersedia di Wilayah Anda dapat menyebabkan permintaan gagal sepenuhnya. Untuk mengatasi masalah ini, coba lagi permintaan dengan tipe instans yang berbeda, pastikan setiap jenis instance tersedia di Wilayah. Untuk mencari jenis contoh yang ditawarkan berdasarkan lokasi, gunakan [describe-instance-type-offerings](#) perintah. Untuk informasi selengkapnya, lihat [Menemukan jenis instans Amazon EC2](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Sumber daya terkait

Untuk praktik terbaik lainnya untuk Instans Spot, lihat [Praktik terbaik untuk EC2 Spot](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Batasan

Setelah menambahkan penggantian ke grup Auto Scaling menggunakan kebijakan [instance campuran](#), Anda dapat memperbarui penggantian dengan UpdateAutoScalingGroup panggilan API tetapi tidak menghapusnya. Untuk menghapus penggantian sepenuhnya, Anda harus terlebih dahulu mengganti grup Auto Scaling untuk menggunakan templat peluncuran atau konfigurasi

peluncuran alih-alih kebijakan instance campuran. Kemudian, Anda dapat menambahkan kebijakan instance campuran lagi tanpa penggantian apa pun.

Strategi alokasi

Saat menggunakan beberapa jenis instans, Anda mengelola cara Auto Scaling Amazon EC2 memenuhi On-Demand dan Spot sesuai permintaan dari jenis instans yang memungkinkan. Untuk melakukan ini, Anda menentukan strategi alokasi strategi .

Untuk meninjau praktik terbaik untuk grup instance campuran, lihat [Gambaran umum pengaturan](#).

Daftar Isi

- [Instans Spot](#)
- [Instans Sesuai Permintaan](#)
- [Bagaimana strategi alokasi bekerja dengan bobot](#)

Instans Spot

Auto Scaling Amazon EC2 menyediakan strategi alokasi berikut untuk Instans Spot:

price-capacity-optimized (direkomendasikan)

Strategi alokasi harga dan kapasitas yang dioptimalkan melihat harga dan kapasitas untuk memilih kumpulan Instans Spot yang paling kecil kemungkinannya untuk terganggu dan memiliki harga serendah mungkin.

Kami merekomendasikan strategi ini ketika Anda memulai. Untuk informasi selengkapnya, lihat [Memperkenalkan strategi price-capacity-optimized alokasi untuk Instans Spot EC2 di blog](#). AWS

capacity-optimized

Auto Scaling Amazon EC2 meminta Instans Spot Anda dari kumpulan dengan kapasitas optimal untuk jumlah instans yang diluncurkan.

Dengan Instans Spot, harga berubah secara perlahan dari waktu ke waktu berdasarkan tren penawaran dan permintaan jangka panjang. Namun, kapasitas berfluktuasi secara real time. Strategi capacity-optimized secara otomatis meluncurkan Instans Spot ke dalam kolam yang paling tersedia dengan melihat data kapasitas waktu nyata dan memprediksi kolam mana yang paling tersedia. Ini membantu meminimalkan kemungkinan gangguan untuk beban kerja yang mungkin memiliki biaya interupsi yang lebih tinggi terkait dengan memulai kembali pekerjaan dan

pos pemeriksaan. Untuk memberikan jenis instans tertentu kesempatan yang lebih tinggi untuk meluncurkan terlebih dahulu, gunakan `capacity-optimized-prioritized`.

`capacity-optimized-prioritized`

Anda mengatur urutan jenis instance untuk penggantian template peluncuran dari prioritas tertinggi ke prioritas terendah (dari pertama hingga terakhir dalam daftar). Amazon EC2 Auto Scaling patuh pada prioritas jenis instans berdasarkan upaya terbaik tetapi mengoptimalkan kapasitas terlebih dahulu. Ini adalah pilihan yang baik untuk beban kerja di mana kemungkinan gangguan harus diminimalkan, tetapi preferensi untuk jenis instance tertentu juga penting. Jika strategi alokasi On-Demand diatur `prioritized`, prioritas yang sama diterapkan ketika memenuhi kapasitas On-Demand.

`lowest-price`

Auto Scaling Amazon EC2 meminta Instans Spot Anda menggunakan kumpulan harga terendah dalam Availability Zone, di seluruh N jumlah kumpulan Spot yang Anda tentukan untuk pengaturan pool dengan harga terendah. Misalnya, jika Anda menentukan empat jenis instans dan empat Availability Zone, grup Auto Scaling Anda dapat mengakses hingga 16 kumpulan Spot. (Empat di setiap Availability Zone.) Jika Anda menentukan dua kumpulan Spot (N=2) untuk strategi alokasi, grup Auto Scaling Anda dapat menggunakan dua pool dengan harga terendah per Availability Zone untuk memenuhi kapasitas Spot Anda.

Karena strategi ini hanya mempertimbangkan harga instans dan bukan ketersediaan kapasitas, hal ini dapat menyebabkan tingkat interupsi yang tinggi.

Auto Scaling Amazon EC2 berupaya menarik Instans Spot dari jumlah N kumpulan yang Anda tentukan. Namun, jika kolam kehabisan kapasitas Spot sebelum memenuhi kapasitas yang Anda inginkan, Auto Scaling Amazon EC2 terus memenuhi permintaan Anda dengan menarik dari kumpulan harga terendah berikutnya. Untuk memenuhi kapasitas yang Anda inginkan, Anda mungkin menerima Instans Spot dari lebih banyak kumpulan daripada nomor N yang Anda tentukan. Demikian juga, jika sebagian besar kolam tidak memiliki kapasitas Spot, Anda mungkin menerima kapasitas penuh yang diinginkan dari kumpulan yang lebih sedikit daripada nomor N yang Anda tentukan.

Note

Jika Anda mengonfigurasi Instans Spot untuk diluncurkan dengan [AMD SEV-SNP](#) diaktifkan, Anda akan dikenakan biaya penggunaan tambahan per jam yang setara dengan 10%

dari tarif [per jam Sesuai Permintaan dari jenis instans yang dipilih](#). Jika strategi alokasi menggunakan harga sebagai input, Auto Scaling Amazon EC2 tidak termasuk biaya tambahan ini; hanya harga Spot yang digunakan.

Instans Sesuai Permintaan

Auto Scaling Amazon EC2 menyediakan strategi alokasi berikut yang dapat digunakan untuk Instans Sesuai Permintaan:

lowest-price

Auto Scaling Amazon EC2 secara otomatis menerapkan jenis instans dengan harga terendah di setiap Availability Zone berdasarkan harga Sesuai Permintaan saat ini.

Untuk memenuhi kapasitas yang Anda inginkan, Anda mungkin menerima Instans Sesuai Permintaan lebih dari satu jenis instans di setiap Availability Zone. Ini tergantung pada seberapa banyak kapasitas yang Anda minta.

prioritized

Saat memenuhi kapasitas Sesuai Permintaan, Auto Scaling Amazon EC2 menentukan jenis instans mana yang akan digunakan terlebih dahulu berdasarkan urutan jenis instans dalam daftar penggantian template peluncuran. Misalnya, katakanlah Anda menentukan tiga penggantian template peluncuran dalam urutan berikut: `c5.large`, `c4.large`, dan `c3.large`. Saat Instans Sesuai Permintaan Anda diluncurkan, grup Auto Scaling memenuhi kapasitas Sesuai Permintaan dalam urutan `c5.large` berikut:., dan kemudian. `c4.large` `c3.large`

Pertimbangkan hal berikut saat mengelola urutan prioritas Instans Sesuai Permintaan Anda:

- Anda dapat membayar penggunaan di muka untuk mendapatkan diskon signifikan untuk Instans Sesuai Permintaan dengan menggunakan Savings Plans atau Instans Cadangan. Untuk informasi selengkapnya, lihat halaman [harga Amazon EC2](#).
- Dengan Instans Pesanan, tarif diskon Anda dari harga Instans Sesuai Permintaan berlaku jika Amazon EC2 Auto Scaling meluncurkan jenis instans yang cocok. Oleh karena itu, jika Anda memiliki Instans Cadangan yang tidak digunakan `c4.large`, Anda dapat mengatur prioritas tipe instans untuk memberikan prioritas tertinggi untuk Instans Cadangan Anda ke tipe instans. `c4.large`. Saat `c4.large` instans diluncurkan, Anda menerima harga Instans Dipesan.
- Dengan Savings Plans, tarif diskon dari harga Instans Sesuai Permintaan reguler berlaku saat menggunakan Amazon EC2 Instans Savings Plans atau Compute Savings Plans. Dengan

Savings Plans, Anda memiliki lebih banyak fleksibilitas saat memprioritaskan jenis instans Anda. Selama Anda menggunakan jenis instans yang dicakup oleh Savings Plan, Anda dapat mengaturnya dalam urutan prioritas apa pun. Anda juga sesekali dapat mengubah seluruh urutan jenis instans Anda, sambil tetap menerima tarif diskon Savings Plan. Untuk mempelajari Savings Plans selengkapnya, lihat [Panduan Pengguna Savings Plans](#).

Bagaimana strategi alokasi bekerja dengan bobot

Saat Anda menentukan `WeightedCapacity` parameter dalam penggantian Anda (atau `"DesiredCapacityType": "vcpu"` atau `"DesiredCapacityType": "memory-mib"` di tingkat grup), strategi alokasi bekerja persis seperti yang mereka lakukan untuk grup Auto Scaling lainnya.

Satu-satunya perbedaan adalah ketika Anda memilih `price-capacity-optimized` strategi `lowest-price` atau, instans Anda berasal dari kumpulan instans dengan harga per unit terendah di setiap Availability Zone. Untuk informasi selengkapnya, lihat [Konfigurasi grup Auto Scaling untuk menggunakan bobot instance](#).

Misalnya, bayangkan Anda memiliki grup Auto Scaling dengan beberapa tipe instance yang memiliki jumlah vCPU yang bervariasi. Anda gunakan `lowest-price` untuk strategi alokasi Spot dan On-Demand Anda. Jika Anda memilih untuk menetapkan bobot berdasarkan jumlah vCPU dari setiap jenis instans, Auto Scaling Amazon EC2 meluncurkan jenis instans mana pun yang memiliki harga terendah per nilai bobot yang ditetapkan (misalnya, per vCPU) pada saat pemenuhan. Jika itu adalah Instance Spot, maka ini berarti harga Spot terendah per vCPU. Jika ini adalah Instans On-Demand, maka ini berarti harga On-Demand terendah per vCPU.

Buat grup instance campuran menggunakan pemilihan tipe instans berbasis atribut

Alih-alih memilih tipe instans secara manual untuk grup instance campuran, Anda dapat menentukan satu set atribut instance yang menjelaskan persyaratan komputasi Anda. Saat Auto Scaling Amazon EC2 meluncurkan instans, semua jenis instans yang digunakan oleh grup Auto Scaling harus cocok dengan atribut instans yang Anda perlukan. Hal ini dikenal sebagai pemilihan tipe instans berbasis atribut.

Pendekatan ini sangat ideal untuk beban kerja dan kerangka kerja yang dapat fleksibel tentang jenis instance yang mereka gunakan, seperti wadah, data besar, dan CI/CD.

Berikut ini adalah manfaat pemilihan jenis instans berbasis atribut:

- Fleksibilitas optimal untuk Instans Spot — Auto Scaling Amazon EC2 dapat memilih dari berbagai jenis instans untuk meluncurkan Instans Spot. Ini memenuhi praktik terbaik Spot untuk menjadi fleksibel tentang jenis instans, yang memberi layanan Amazon EC2 Spot peluang lebih baik untuk menemukan dan mengalokasikan jumlah kapasitas komputasi yang Anda butuhkan.
- Mudah menggunakan jenis instans yang tepat — Dengan begitu banyak jenis instans yang tersedia, menemukan jenis instans yang tepat untuk beban kerja Anda dapat memakan waktu. Saat Anda menentukan atribut instans, tipe instans akan secara otomatis memiliki atribut yang diperlukan untuk beban kerja Anda.
- Penggunaan otomatis jenis instans baru — Grup Auto Scaling Anda dapat menggunakan tipe instance generasi yang lebih baru saat dirilis. Jenis instans generasi yang lebih baru secara otomatis digunakan jika sesuai dengan kebutuhan Anda dan selaras dengan strategi alokasi yang Anda pilih untuk grup Auto Scaling Anda.

Topik

- [Cara kerja pemilihan tipe instans berbasis atribut](#)
- [Perlindungan harga](#)
- [Prasyarat](#)
- [Buat grup instance campuran dengan pemilihan tipe instans berbasis atribut \(konsol\)](#)
- [Buat grup instance campuran dengan pemilihan tipe instance berbasis atribut \(AWS CLI\)](#)
- [Contoh konfigurasi](#)
- [Pratinjau jenis instans Anda](#)
- [Sumber daya terkait](#)

Cara kerja pemilihan tipe instans berbasis atribut

Dengan pemilihan tipe instans berbasis atribut, alih-alih menyediakan daftar tipe instans tertentu, Anda menyediakan daftar atribut instance yang dibutuhkan instance Anda, seperti:

- Jumlah vCPU — Jumlah minimum dan maksimum vCPU per instance.
- Memori — Minimum dan GiBs maksimum memori per instance.
- Penyimpanan lokal — Apakah akan menggunakan EBS atau volume penyimpanan instans untuk penyimpanan lokal.
- Kinerja burstable — Apakah akan menggunakan keluarga instans T, termasuk tipe T4G, T3a, T3, dan T2.

Ada banyak opsi yang tersedia untuk menentukan persyaratan instans Anda. Untuk deskripsi setiap opsi dan nilai default, lihat [InstanceRequirements](#) di Referensi API Auto Scaling Amazon EC2.

Saat grup Auto Scaling Anda perlu meluncurkan instance, grup tersebut akan mencari jenis instance yang cocok dengan atribut yang Anda tentukan dan tersedia di Availability Zone tersebut. Strategi alokasi kemudian menentukan jenis instance mana yang cocok untuk diluncurkan. Secara default, pemilihan jenis instans berbasis atribut memiliki fitur perlindungan harga yang diaktifkan untuk mencegah grup Auto Scaling meluncurkan jenis instans yang melebihi ambang batas anggaran Anda.

Secara default, Anda menggunakan jumlah instance sebagai unit pengukuran saat menyetel kapasitas grup Auto Scaling yang diinginkan, yang berarti setiap instans dihitung sebagai satu unit.

Atau, Anda dapat mengatur nilai untuk kapasitas yang diinginkan ke jumlah vCPU atau jumlah memori. Untuk melakukannya, gunakan bidang tarik-turun tipe kapasitas yang diinginkan di AWS Management Console atau `DesiredCapacityType` properti dalam operasi `CreateAutoScalingGroup` atau `UpdateAutoScalingGroup` API. Auto Scaling Amazon EC2 kemudian meluncurkan jumlah instans yang diperlukan untuk memenuhi kapasitas vCPU atau memori yang diinginkan. Misalnya, jika Anda menggunakan vCPU sebagai tipe kapasitas yang diinginkan dan menggunakan instance dengan masing-masing 2 vCPU, kapasitas 10 vCPU yang diinginkan akan meluncurkan 5 instance. Ini adalah alternatif yang berguna untuk [bobot contoh](#).

Perlindungan harga

Dengan perlindungan harga, Anda dapat menentukan harga maksimum yang bersedia Anda bayar untuk instans EC2 yang diluncurkan oleh grup Auto Scaling Anda. Perlindungan harga adalah fitur yang mencegah grup Auto Scaling Anda menggunakan tipe instans yang Anda anggap terlalu mahal meskipun sesuai dengan atribut yang Anda tentukan.

Perlindungan harga diaktifkan secara default dan memiliki ambang harga terpisah untuk Instans Sesuai Permintaan dan Instans Spot. Jika Auto Scaling Amazon EC2 perlu meluncurkan instans baru, semua jenis instans dengan harga di atas ambang batas yang relevan tidak akan diluncurkan.

Topik

- [Perlindungan harga On-Demand](#)
- [Perlindungan harga spot](#)
- [Sesuaikan perlindungan harga](#)

Perlindungan harga On-Demand

Untuk Instans Sesuai Permintaan, Anda menentukan harga On-Demand maksimum yang bersedia Anda bayarkan sebagai persentase lebih tinggi daripada harga On-Demand yang teridentifikasi. Harga Sesuai Permintaan yang teridentifikasi adalah harga tipe instans C, M, atau R generasi terbaru dengan harga terendah dengan atribut yang Anda tentukan.

Jika nilai perlindungan harga On-Demand tidak didefinisikan secara eksplisit, harga On-Demand maksimum default 20 persen lebih tinggi dari harga On-Demand yang diidentifikasi akan digunakan.

Perlindungan harga spot

Secara default, Auto Scaling Amazon EC2 akan secara otomatis menerapkan perlindungan harga Instans Spot yang optimal untuk secara konsisten memilih dari berbagai jenis instans. Anda juga dapat mengatur sendiri perlindungan harga secara manual. Namun, membiarkan Amazon EC2 Auto Scaling melakukannya untuk Anda dapat meningkatkan kemungkinan kapasitas Spot Anda terpenuhi.

Anda dapat menentukan perlindungan harga secara manual menggunakan salah satu opsi berikut. Jika Anda secara manual mengatur perlindungan harga, kami sarankan menggunakan opsi pertama.

- Persentase harga Sesuai Permintaan yang teridentifikasi — Harga Sesuai Permintaan yang teridentifikasi adalah harga dari jenis instans C, M, atau R generasi terbaru dengan harga terendah dengan atribut yang Anda tentukan.
- Persentase lebih tinggi dari harga Spot yang teridentifikasi — Harga Spot yang teridentifikasi adalah harga dari jenis instans C, M, atau R generasi terbaru dengan harga terendah dengan atribut yang Anda tentukan. Kami tidak menyarankan menggunakan opsi ini karena harga Spot dapat berfluktuasi, dan oleh karena itu ambang batas perlindungan harga Anda mungkin juga berfluktuasi.

Sesuaikan perlindungan harga

Anda dapat menyesuaikan ambang batas perlindungan harga di konsol Auto Scaling Amazon EC2 atau menggunakan atau SDK. AWS CLI

- Di konsol, gunakan perlindungan harga Sesuai Permintaan dan pengaturan perlindungan harga Spot di atribut instance tambahan.
- Dalam [InstanceRequirements](#) struktur, untuk menentukan ambang perlindungan harga Instance On-Demand, gunakan `OnDemandMaxPricePercentageOverLowestPrice`

properti. Untuk menentukan ambang perlindungan harga Instans Spot, gunakan properti `MaxSpotPriceAsPercentageOfOptimalOnDemandPrice` atau `SpotMaxPricePercentageOverLowestPrice` properti.

Jika Anda menyetel tipe kapasitas yang diinginkan (`DesiredCapacityType`) ke vCPU atau Memory GiB, perlindungan harga berlaku berdasarkan harga per vCPU atau per memori, bukan harga per instans.

Anda juga dapat mematikan perlindungan harga. Untuk menunjukkan tidak ada ambang perlindungan harga, tentukan nilai persentase tinggi, seperti 999999.

Note

Jika tidak ada tipe instance C, M, atau R generasi saat ini yang cocok dengan atribut yang Anda tentukan, perlindungan harga masih berlaku. Jika tidak ada kecocokan yang ditemukan, harga yang diidentifikasi berasal dari jenis instans generasi saat ini dengan harga terendah, atau jika gagal, jenis instance generasi sebelumnya dengan harga terendah, yang cocok dengan atribut Anda.

Prasyarat

- Buat template peluncuran. Untuk informasi selengkapnya, lihat [Membuat template peluncuran untuk grup Auto Scaling](#).
- Verifikasi bahwa template peluncuran belum meminta Instans Spot.

Buat grup instance campuran dengan pemilihan tipe instans berbasis atribut (konsol)

Gunakan prosedur berikut untuk membuat grup instance campuran dengan menggunakan pemilihan tipe instans berbasis atribut. Untuk membantu Anda melewati langkah-langkah secara efisien, beberapa bagian opsional dilewati.

Untuk sebagian besar beban kerja tujuan umum, cukup untuk menentukan jumlah vCPU dan memori yang Anda butuhkan. Untuk kasus penggunaan lanjutan, Anda dapat menentukan atribut seperti tipe penyimpanan, antarmuka jaringan, produsen CPU, dan tipe akselerator.

Untuk meninjau praktik terbaik untuk grup instance campuran, lihat [Gambaran umum pengaturan](#).

Untuk membuat grup instance campuran

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pada bilah navigasi di bagian atas layar, pilih yang sama dengan Wilayah AWS yang Anda gunakan saat membuat templat peluncuran.
3. Pilih Buat grup Auto Scaling.
4. Pada halaman Pilih templat peluncuran atau konfigurasi, untuk Nama grup Auto Scaling, masukkan nama untuk grup Auto Scaling Anda.
5. Untuk memilih template peluncuran Anda, lakukan hal berikut:
 - a. Untuk Templat peluncuran, pilih templat peluncuran yang ada.
 - b. Untuk Versi templat peluncuran, pilih apakah grup Auto Scaling menggunakan default, terbaru, atau versi spesifik templat peluncuran saat menskalakan.
 - c. Verifikasi bahwa templat peluncuran Anda mendukung semua opsi yang Anda rencanakan untuk digunakan, lalu pilih Selanjutnya.
6. Pada halaman Opsi peluncuran contoh Pilih, lakukan hal berikut:
 - a. Untuk persyaratan tipe Instance, pilih Override launch template.

Note

Jika Anda memilih template peluncuran yang sudah berisi sekumpulan atribut instance, seperti vCPU dan memori, maka atribut instance akan ditampilkan. Atribut ini ditambahkan ke properti grup Auto Scaling, tempat Anda dapat memperbaruinya dari konsol Auto Scaling Amazon EC2 kapan saja.

- b. Di bawah Tentukan atribut instance, mulailah dengan memasukkan vCPU dan persyaratan memori Anda.
 - Untuk vCPU, masukkan jumlah minimum dan maksimum vCPU yang diinginkan. Untuk menentukan tanpa batasan, pilih Tanpa minimum, Tanpa maksimum, atau keduanya.
 - Untuk Memori (GiB), masukkan jumlah memori minimum dan maksimum yang diinginkan. Untuk menentukan tanpa batasan, pilih Tanpa minimum, Tanpa maksimum, atau keduanya.

- c. (Opsional) Untuk atribut instans Tambahan, Anda dapat secara opsional menentukan satu atau lebih atribut untuk mengekspresikan kebutuhan komputasi Anda secara lebih mendetail. Setiap atribut tambahan menambahkan batasan lebih lanjut untuk permintaan Anda.
 - d. Perluas jenis instans yang cocok dengan Pratinjau untuk melihat jenis instans yang memiliki atribut yang Anda tentukan.
 - e. Di bawah Opsi pembelian Instans, untuk distribusi Instans, tentukan persentase grup yang akan diluncurkan sebagai Instans Sesuai Permintaan dan sebagai Instans Spot. Jika aplikasi Anda stateless, toleran terhadap kesalahan, dan dapat menangani instance yang sedang terputus, Anda dapat menentukan persentase Instans Spot yang lebih tinggi.
 - f. (Opsional) Bila Anda menentukan persentase untuk Instans Spot, pilih Sertakan kapasitas dasar Sesuai Permintaan, lalu tentukan jumlah minimum kapasitas awal grup Auto Scaling yang harus dipenuhi oleh Instans Sesuai Permintaan. Apa pun di luar kapasitas dasar menggunakan setelan distribusi Instans untuk menentukan berapa banyak Instans Sesuai Permintaan dan Instans Spot yang akan diluncurkan.
 - g. Di bawah strategi Alokasi, harga terendah secara otomatis dipilih untuk strategi alokasi On-Demand dan tidak dapat diubah.
 - h. Untuk strategi alokasi Spot, pilih strategi alokasi. Kapasitas harga yang dioptimalkan dipilih secara default. Harga terendah disembunyikan secara default dan hanya muncul ketika Anda memilih Tampilkan semua strategi. Jika Anda memilih harga terendah, masukkan jumlah kolam dengan harga terendah untuk melakukan diversifikasi di seluruh kolam dengan harga terendah.
 - i. Untuk Penyeimbangan Kembali Kapasitas, pilih apakah akan mengaktifkan atau menonaktifkan Penyeimbangan Kembali Kapasitas. Gunakan Penyeimbangan Kembali Kapasitas untuk merespons secara otomatis saat Instans Spot Anda mendekati penghentian dari gangguan Spot. Untuk informasi selengkapnya, lihat [Gunakan Rebalancing Kapasitas untuk menangani interupsi Spot Amazon EC2](#).
 - j. Di bawah Jaringan, untuk VPC, pilih VPC. Grup Auto Scaling harus dibuat dalam VPC yang sama dengan grup keamanan yang Anda tentukan dalam template peluncuran Anda.
 - k. Untuk Availability Zones dan subnet, pilih satu atau lebih subnet di VPC yang ditentukan. Gunakan subnet di beberapa Availability Zone untuk ketersediaan tinggi. Untuk informasi selengkapnya, lihat [Pertimbangan saat memilih subnet VPC](#).
 - l. Pilih Berikutnya, Berikutnya.
7. Untuk langkah Mengonfigurasi ukuran grup dan kebijakan penskalaan, lakukan hal berikut:

- a. Untuk mengukur kapasitas yang Anda inginkan dalam unit selain instance, pilih opsi yang sesuai untuk Ukuran grup, Jenis kapasitas yang diinginkan. Unit, vCPU, dan Memory GiB didukung. Secara default, Auto Scaling Amazon EC2 menentukan Unit, yang diterjemahkan ke dalam jumlah instans.
 - b. Untuk kapasitas yang diinginkan, ukuran awal grup Auto Scaling Anda.
 - c. Di bagian Penskalaan, di bawah batas Penskalaan, jika nilai baru Anda untuk kapasitas yang diinginkan lebih besar dari kapasitas yang diinginkan Min dan kapasitas yang diinginkan Maks, kapasitas Maks yang diinginkan secara otomatis ditingkatkan ke nilai kapasitas baru yang diinginkan. Anda dapat mengubah batasan ini sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).
8. Pilih Lewati untuk meninjau.
 9. Pada halaman Peninjauan, pilih Buat grup Auto Scaling.

Buat grup instance campuran dengan pemilihan tipe instance berbasis atribut (AWS CLI)

Untuk membuat grup instance campuran menggunakan baris perintah

Gunakan salah satu perintah berikut:

- [create-auto-scaling-group](#) (AWS CLI)
- [Baru-AS \(AutoScalingGroup\)](#) AWS Tools for Windows PowerShell

Contoh konfigurasi

Untuk membuat grup Auto Scaling dengan pemilihan tipe instans berbasis atribut menggunakan AWS CLI, gunakan perintah berikut. [create-auto-scaling-group](#)

Atribut contoh berikut ditentukan:

- VCpuCountJenis instans harus memiliki minimal empat vCPU dan maksimal delapan vCPU.
- MemoryMiBJenis instans harus memiliki memori minimal 16.384 MiB.
- CpuManufacturersJenis instans harus memiliki CPU yang diproduksi Intel.

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

Berikut ini adalah contoh `config.json` file.

```
{
  "AutoScalingGroupName": "my-asg",
  "DesiredCapacityType": "units",
  "MixedInstancesPolicy": {
    "LaunchTemplate": {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "$Default"
      },
      "Overrides": [{
        "InstanceRequirements": {
          "VCpuCount": {"Min": 4, "Max": 8},
          "MemoryMiB": {"Min": 16384},
          "CpuManufacturers": ["intel"]
        }
      }]
    },
    "InstancesDistribution": {
      "OnDemandPercentageAboveBaseCapacity": 50,
      "SpotAllocationStrategy": "price-capacity-optimized"
    }
  },
  "MinSize": 0,
  "MaxSize": 100,
  "DesiredCapacity": 4,
  "DesiredCapacityType": "units",
  "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}
```

Untuk mengatur nilai kapasitas yang diinginkan sebagai jumlah vCPU atau jumlah memori, `"DesiredCapacityType": "vcpu"` tentukan `"DesiredCapacityType": "memory-mib"` atau dalam file. Jenis kapasitas default yang diinginkan adalah `units`, yang menetapkan nilai untuk kapasitas yang diinginkan sebagai jumlah instance.

YAML

Atau, Anda dapat menggunakan [create-auto-scaling-group](#) perintah berikut untuk membuat grup Auto Scaling. Ini mereferensikan file YAMB sebagai satu-satunya parameter untuk grup Auto Scaling Anda.

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

Berikut ini adalah contoh `config.yaml` file.

```
---
AutoScalingGroupName: my-asg
DesiredCapacityType: units
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
      - InstanceRequirements:
          VCpuCount:
            Min: 2
            Max: 4
          MemoryMiB:
            Min: 2048
          CpuManufacturers:
            - intel
      InstancesDistribution:
        OnDemandPercentageAboveBaseCapacity: 50
        SpotAllocationStrategy: price-capacity-optimized
  MinSize: 0
  MaxSize: 100
  DesiredCapacity: 4
DesiredCapacityType: units
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

Untuk mengatur nilai kapasitas yang diinginkan sebagai jumlah vCPU atau jumlah memori, `DesiredCapacityType: vcpu` tentukan `DesiredCapacityType: memory-mib` atau dalam file. Jenis kapasitas default yang diinginkan adalah `units`, yang menetapkan nilai untuk kapasitas yang diinginkan sebagai jumlah instance.

Pratinjau jenis instans Anda

Anda dapat melihat pratinjau jenis instans yang sesuai dengan persyaratan komputasi tanpa meluncurkannya dan menyesuaikan persyaratan Anda jika perlu. Saat membuat grup Auto Scaling di konsol Auto Scaling Amazon EC2, pratinjau jenis instans akan muncul di bagian Pratinjau jenis instans yang cocok di halaman Pilih opsi peluncuran instans.

Atau, Anda dapat melihat pratinjau jenis instans dengan membuat panggilan [GetInstanceTypesFromInstanceRequirements](#) API Amazon EC2 menggunakan AWS CLI atau SDK. Lulus InstanceRequirements parameter dalam permintaan dalam format yang tepat yang akan Anda gunakan untuk membuat atau memperbarui grup Auto Scaling. Untuk informasi selengkapnya, lihat [Pratinjau jenis instans dengan atribut tertentu](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Sumber daya terkait

Untuk mempelajari selengkapnya tentang pemilihan jenis instans berbasis atribut, lihat Pemilihan Jenis Instans [Berbasis Atribut untuk EC2 Auto Scaling dan Armada EC2 di Blog](#). AWS

Anda dapat mendeklarasikan pemilihan jenis instans berbasis atribut saat membuat grup Auto Scaling menggunakan AWS CloudFormation. Untuk informasi selengkapnya, lihat contoh cuplikan di bagian [Cuplikan templat penskalaan otomatis pada Panduan Pengguna AWS CloudFormation](#).

Buat grup instance campuran dengan memilih jenis instance secara manual

Topik ini menunjukkan cara meluncurkan beberapa jenis instans dalam satu grup Auto Scaling dengan memilih tipe instans secara manual.

Jika Anda lebih suka menggunakan atribut instance sebagai kriteria untuk memilih jenis instance, lihat [Buat grup instance campuran menggunakan pemilihan tipe instans berbasis atribut](#).

Daftar Isi

- [Prasyarat](#)
- [Buat grup instance campuran \(konsol\)](#)
- [Buat grup instance campuran \(AWS CLI\)](#)
- [Contoh konfigurasi](#)

Prasyarat

- Buat template peluncuran. Untuk informasi selengkapnya, lihat [Membuat template peluncuran untuk grup Auto Scaling](#).
- Verifikasi bahwa template peluncuran belum meminta Instans Spot.

Buat grup instance campuran (konsol)

Gunakan prosedur berikut untuk membuat grup instance campuran dengan secara manual memilih jenis instans mana yang dapat diluncurkan grup Anda. Untuk membantu Anda melewati langkah-langkah secara efisien, beberapa bagian opsional dilewati.

Untuk meninjau praktik terbaik untuk grup instance campuran, lihat [Gambaran umum pengaturan](#).

Untuk membuat grup instance campuran

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pada bilah navigasi di bagian atas layar, pilih yang sama dengan Wilayah AWS yang Anda gunakan saat membuat templat peluncuran.
3. Pilih Buat grup Auto Scaling.
4. Pada halaman Pilih templat peluncuran atau konfigurasi, untuk Nama grup Auto Scaling, masukkan nama untuk grup Auto Scaling Anda.
5. Untuk memilih template peluncuran Anda, lakukan hal berikut:
 - a. Untuk Templat peluncuran, pilih templat peluncuran yang ada.
 - b. Untuk Versi templat peluncuran, pilih apakah grup Auto Scaling menggunakan default, terbaru, atau versi spesifik templat peluncuran saat menskalakan.
 - c. Verifikasi bahwa template peluncuran Anda mendukung semua opsi yang Anda rencanakan untuk digunakan, lalu pilih Berikutnya.
6. Pada halaman Opsi peluncuran contoh Pilih, lakukan hal berikut:
 - a. Untuk persyaratan tipe Instance, pilih Override launch template, lalu pilih Tambahkan tipe instance secara manual.
 - b. Pilih jenis instans Anda. Anda dapat menggunakan rekomendasi kami sebagai titik awal. Fleksibel keluarga dan generasi dipilih secara default.
 - Untuk mengubah urutan jenis instance, gunakan panah. Jika Anda memilih strategi alokasi yang mendukung prioritas, urutan tipe instance menetapkan prioritas peluncurannya.
 - Untuk menghapus jenis instance, pilih X.
 - (Opsional) Untuk kotak di kolom Berat, tetapkan setiap jenis contoh bobot relatif. Untuk melakukannya, masukkan jumlah unit yang dihitung oleh instance dari jenis itu terhadap

kapasitas grup yang diinginkan. Melakukannya mungkin berguna jika jenis instance menawarkan kemampuan vCPU, memori, penyimpanan, atau bandwidth jaringan yang berbeda. Untuk informasi selengkapnya, lihat [Konfigurasi grup Auto Scaling untuk menggunakan bobot instance](#).

Perhatikan bahwa jika Anda memilih untuk menggunakan rekomendasi fleksibel Ukuran, maka semua jenis instans yang merupakan bagian dari bagian ini secara otomatis memiliki nilai bobot. Jika Anda tidak ingin menentukan bobot apa pun, kosongkan kotak di kolom Berat untuk semua jenis instance.

- c. Di bawah Opsi pembelian Instans, untuk distribusi Instans, tentukan persentase grup yang akan diluncurkan sebagai Instans Sesuai Permintaan dan Instans Spot masing-masing. Jika aplikasi Anda stateless, toleran terhadap kesalahan, dan dapat menangani instance yang sedang terputus, Anda dapat menentukan persentase Instans Spot yang lebih tinggi.
- d. (Opsional) Bila Anda menentukan persentase untuk Instans Spot, pilih Sertakan kapasitas dasar Sesuai Permintaan, lalu tentukan jumlah minimum kapasitas awal grup Auto Scaling yang harus dipenuhi oleh Instans Sesuai Permintaan. Apa pun di luar kapasitas dasar menggunakan setelan distribusi Instans untuk menentukan berapa banyak Instans Sesuai Permintaan dan Instans Spot yang akan diluncurkan.
- e. Di bawah strategi alokasi, untuk strategi alokasi On-Demand, pilih strategi alokasi. Saat Anda memilih jenis instans secara manual, Prioritisasi dipilih secara default.
- f. Untuk strategi alokasi Spot, pilih strategi alokasi. Kapasitas harga yang dioptimalkan dipilih secara default. Harga terendah disembunyikan secara default dan hanya muncul ketika Anda memilih Tampilkan semua strategi.
 - Jika Anda memilih harga terendah, masukkan jumlah kolom dengan harga terendah untuk melakukan diversifikasi di seluruh kolam dengan harga terendah.
 - Jika memilih Kapasitas yang dioptimalkan, Anda dapat secara opsional mencentang kotak Prioritaskan jenis instans agar Auto Scaling Amazon EC2 memilih jenis instans mana yang akan diluncurkan terlebih dahulu berdasarkan urutan jenis instans yang dicantumkan.
- g. Untuk Penyeimbangan Kembali Kapasitas, pilih apakah akan mengaktifkan atau menonaktifkan Penyeimbangan Kembali Kapasitas. Gunakan Penyeimbangan Kembali Kapasitas untuk merespons secara otomatis saat Instans Spot Anda mendekati penghentian dari gangguan Spot. Untuk informasi selengkapnya, lihat [Gunakan Rebalancing Kapasitas untuk menangani interupsi Spot Amazon EC2](#).

- h. Di bawah Jaringan, untuk VPC, pilih VPC. Grup Auto Scaling harus dibuat dalam VPC yang sama dengan grup keamanan yang Anda tentukan dalam template peluncuran Anda.
 - i. Untuk Availability Zones dan subnet, pilih satu atau lebih subnet di VPC yang ditentukan. Gunakan subnet di beberapa Availability Zone untuk ketersediaan tinggi. Untuk informasi selengkapnya, lihat [Pertimbangan saat memilih subnet VPC](#).
 - j. Pilih Berikutnya, Berikutnya.
7. Untuk langkah Mengonfigurasi ukuran grup dan kebijakan penskalaan, lakukan hal berikut:
- a. Di bawah Ukuran grup, untuk kapasitas yang diinginkan, masukkan jumlah instans awal yang akan diluncurkan.

Secara default, kapasitas yang diinginkan dinyatakan sebagai jumlah instance. Jika Anda menetapkan bobot ke jenis instans, Anda harus mengonversi nilai ini ke satuan pengukuran yang sama dengan yang Anda gunakan untuk menetapkan bobot, seperti jumlah vCPU.
 - b. Di bagian Penskalaan, di bawah batas Penskalaan, jika nilai baru Anda untuk kapasitas yang diinginkan lebih besar dari kapasitas yang diinginkan Min dan kapasitas yang diinginkan Maks, kapasitas Maks yang diinginkan secara otomatis ditingkatkan ke nilai kapasitas baru yang diinginkan. Anda dapat mengubah batasan ini sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).
8. Pilih Lewati untuk meninjau.
9. Pada halaman Peninjauan, pilih Buat grup Auto Scaling.

Buat grup instance campuran (AWS CLI)

Untuk membuat grup instance campuran menggunakan baris perintah

Gunakan salah satu perintah berikut:

- [create-auto-scaling-group](#) (AWS CLI)
- [Baru-AS \(AutoScalingGroup\)](#) AWS Tools for Windows PowerShell

Contoh konfigurasi

Contoh konfigurasi berikut menunjukkan cara membuat grup instance campuran menggunakan strategi alokasi Spot yang berbeda.

Note

Contoh-contoh ini menunjukkan cara menggunakan file konfigurasi yang diformat dalam JSON atau YANG. Jika Anda menggunakan AWS CLI versi 1, Anda harus menentukan file konfigurasi berformat JSON. Jika Anda menggunakan AWS CLI versi 2, Anda dapat menentukan file konfigurasi yang diformat dalam YANG atau JSON.

Contoh-contoh

- [Contoh 1: Luncurkan Instans Spot menggunakan strategi capacity-optimized alokasi](#)
- [Contoh 2: Luncurkan Instans Spot menggunakan strategi capacity-optimized-prioritized alokasi](#)
- [Contoh 3: Luncurkan Instans Spot menggunakan strategi lowest-price alokasi yang terdiversifikasi pada dua kumpulan](#)
- [Contoh 4: Meluncurkan Instans Spot menggunakan strategi alokasi price-capacity-optimized](#)

Contoh 1: Luncurkan Instans Spot menggunakan strategi **capacity-optimized** alokasi

Perintah [create-auto-scaling-group](#) berikut membuat grup Auto Scaling yang menentukan hal berikut:

- Persentase grup yang akan diluncurkan sebagai Instans Sesuai Permintaan (0) dan nomor dasar Instans Sesuai Permintaan untuk memulai dengan (1).
- Jenis instance untuk diluncurkan dalam urutan prioritas (c5.large,c5a.large,m5.large,m5a.large,c4.large,m4.large,c3.large,m3.large).
- Subnet untuk meluncurkan instance (subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782). Masing-masing sesuai dengan Availability Zone yang berbeda.
- Template peluncuran (my-launch-template) dan versi template peluncuran (\$Default).

Ketika Amazon EC2 Auto Scaling berupaya memenuhi kapasitas Sesuai Permintaan Anda, maka akan meluncurkan instans c5.large terlebih dahulu. Spot Instance berasal dari kumpulan Spot optimal di setiap Availability Zone berdasarkan kapasitas Spot Instance.

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

config.jsonFile berisi konten berikut.

```
{
  "AutoScalingGroupName": "my-asg",
  "MixedInstancesPolicy": {
    "LaunchTemplate": {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "$Default"
      },
      "Overrides": [
        {
          "InstanceType": "c5.large"
        },
        {
          "InstanceType": "c5a.large"
        },
        {
          "InstanceType": "m5.large"
        },
        {
          "InstanceType": "m5a.large"
        },
        {
          "InstanceType": "c4.large"
        },
        {
          "InstanceType": "m4.large"
        },
        {
          "InstanceType": "c3.large"
        },
        {
          "InstanceType": "m3.large"
        }
      ]
    },
    "InstancesDistribution": {
      "OnDemandBaseCapacity": 1,
      "OnDemandPercentageAboveBaseCapacity": 0,
      "SpotAllocationStrategy": "capacity-optimized"
    }
  },
  "MinSize": 1,
  "MaxSize": 5,
}
```

```
"DesiredCapacity": 3,  
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"  
}
```

YAML

Atau, Anda dapat menggunakan [create-auto-scaling-group](#) perintah berikut untuk membuat grup Auto Scaling. Ini mereferensikan file YAMG sebagai satu-satunya parameter untuk grup Auto Scaling Anda.

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

config.yamlFile berisi konten berikut.

```
---  
AutoScalingGroupName: my-asg  
MixedInstancesPolicy:  
  LaunchTemplate:  
    LaunchTemplateSpecification:  
      LaunchTemplateName: my-launch-template  
      Version: $Default  
    Overrides:  
      - InstanceType: c5.large  
      - InstanceType: c5a.large  
      - InstanceType: m5.large  
      - InstanceType: m5a.large  
      - InstanceType: c4.large  
      - InstanceType: m4.large  
      - InstanceType: c3.large  
      - InstanceType: m3.large  
    InstancesDistribution:  
      OnDemandBaseCapacity: 1  
      OnDemandPercentageAboveBaseCapacity: 0  
      SpotAllocationStrategy: capacity-optimized  
  MinSize: 1  
  MaxSize: 5  
  DesiredCapacity: 3  
  VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

Contoh 2: Luncurkan Instans Spot menggunakan strategi **capacity-optimized-prioritized** alokasi

Perintah [create-auto-scaling-group](#) berikut membuat grup Auto Scaling yang menentukan hal berikut:

- Persentase grup yang akan diluncurkan sebagai Instans Sesuai Permintaan (0) dan nomor dasar Instans Sesuai Permintaan untuk memulai dengan (). 1
- Jenis instance untuk diluncurkan dalam urutan prioritas (c5.large,c5a.large,m5.large,m5a.large,c4.large,m4.large,c3.large,m3.large).
- Subnet untuk meluncurkan instance (subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782). Masing-masing sesuai dengan Availability Zone yang berbeda.
- Template peluncuran (my-launch-template) dan versi template peluncuran (\$Latest).

Ketika Amazon EC2 Auto Scaling berupaya memenuhi kapasitas Sesuai Permintaan Anda, maka akan meluncurkan instans c5.large terlebih dahulu. Ketika Auto Scaling Amazon EC2 mencoba memenuhi kapasitas Spot Anda, itu menghormati prioritas jenis instans dengan upaya terbaik. Namun, ini mengoptimalkan kapasitas terlebih dahulu.

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

config.jsonFile berisi konten berikut.

```
{
  "AutoScalingGroupName": "my-asg",
  "MixedInstancesPolicy": {
    "LaunchTemplate": {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "$Latest"
      },
      "Overrides": [
        {
          "InstanceType": "c5.large"
        },
        {
          "InstanceType": "c5a.large"
        }
      ]
    }
  }
}
```

```

        "InstanceType": "m5.large"
    },
    {
        "InstanceType": "m5a.large"
    },
    {
        "InstanceType": "c4.large"
    },
    {
        "InstanceType": "m4.large"
    },
    {
        "InstanceType": "c3.large"
    },
    {
        "InstanceType": "m3.large"
    }
]
},
"InstancesDistribution": {
    "OnDemandBaseCapacity": 1,
    "OnDemandPercentageAboveBaseCapacity": 0,
    "SpotAllocationStrategy": "capacity-optimized-prioritized"
}
},
"MinSize": 1,
"MaxSize": 5,
"DesiredCapacity": 3,
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}

```

YAML

Atau, Anda dapat menggunakan [create-auto-scaling-group](#) perintah berikut untuk membuat grup Auto Scaling. Ini mereferensikan file YAMG sebagai satu-satunya parameter untuk grup Auto Scaling Anda.

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

config.yamlFile berisi konten berikut.

```
---
AutoScalingGroupName: my-asg
```

```

MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
      - InstanceType: c5.large
      - InstanceType: c5a.large
      - InstanceType: m5.large
      - InstanceType: m5a.large
      - InstanceType: c4.large
      - InstanceType: m4.large
      - InstanceType: c3.large
      - InstanceType: m3.large
    InstancesDistribution:
      OnDemandBaseCapacity: 1
      OnDemandPercentageAboveBaseCapacity: 0
      SpotAllocationStrategy: capacity-optimized-prioritized
  MinSize: 1
  MaxSize: 5
  DesiredCapacity: 3
  VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782

```

Contoh 3: Luncurkan Instans Spot menggunakan strategi **lowest-price** alokasi yang terdiversifikasi pada dua kumpulan

Perintah [create-auto-scaling-group](#) berikut membuat grup Auto Scaling yang menentukan hal berikut:

- Persentase grup yang akan diluncurkan sebagai On-Demand Instances (50). (Ini tidak menentukan nomor dasar Instans Sesuai Permintaan untuk memulai.)
- Jenis instance untuk diluncurkan dalam urutan prioritas (*c5.large,c5a.large,m5.large,m5a.large,c4.large,m4.large,c3.large,m3.large*).
- Subnet untuk meluncurkan instance (*subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782*). Masing-masing sesuai dengan Availability Zone yang berbeda.
- Template peluncuran (*my-launch-template*) dan versi template peluncuran (*\$Latest*).

Ketika Amazon EC2 Auto Scaling berupaya memenuhi kapasitas Sesuai Permintaan Anda, maka akan meluncurkan instans *c5.large* terlebih dahulu. Untuk kapasitas Spot Anda, Auto Scaling Amazon EC2 mencoba meluncurkan Instans Spot secara merata di dua kumpulan dengan harga terendah di setiap Availability Zone.

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file:///~/config.json
```

`config.json` berisi konten berikut.

```
{
  "AutoScalingGroupName": "my-asg",
  "MixedInstancesPolicy": {
    "LaunchTemplate": {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "$Latest"
      },
      "Overrides": [
        {
          "InstanceType": "c5.large"
        },
        {
          "InstanceType": "c5a.large"
        },
        {
          "InstanceType": "m5.large"
        },
        {
          "InstanceType": "m5a.large"
        },
        {
          "InstanceType": "c4.large"
        },
        {
          "InstanceType": "m4.large"
        },
        {
          "InstanceType": "c3.large"
        },
        {
          "InstanceType": "m3.large"
        }
      ]
    },
    "InstancesDistribution": {
      "OnDemandPercentageAboveBaseCapacity": 50,

```

```

        "SpotAllocationStrategy": "lowest-price",
        "SpotInstancePools": 2
    }
},
"MinSize": 1,
"MaxSize": 5,
"DesiredCapacity": 3,
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}

```

YAML

Atau, Anda dapat menggunakan [create-auto-scaling-group](#) perintah berikut untuk membuat grup Auto Scaling. Ini mereferensikan file YAMG sebagai satu-satunya parameter untuk grup Auto Scaling Anda.

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

config.yamlFile berisi konten berikut.

```

---
AutoScalingGroupName: my-asg
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
      - InstanceType: c5.large
      - InstanceType: c5a.large
      - InstanceType: m5.large
      - InstanceType: m5a.large
      - InstanceType: c4.large
      - InstanceType: m4.large
      - InstanceType: c3.large
      - InstanceType: m3.large
  InstancesDistribution:
    OnDemandPercentageAboveBaseCapacity: 50
    SpotAllocationStrategy: lowest-price
    SpotInstancePools: 2
MinSize: 1
MaxSize: 5

```

```
DesiredCapacity: 3
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

Contoh 4: Meluncurkan Instans Spot menggunakan strategi alokasi **price-capacity-optimized**

Perintah [create-auto-scaling-group](#) berikut membuat grup Auto Scaling yang menentukan hal berikut:

- Persentase grup yang akan diluncurkan sebagai On-Demand Instances ()30. (Ini tidak menentukan nomor dasar Instans Sesuai Permintaan untuk memulai.)
- Jenis instance untuk diluncurkan dalam urutan prioritas (c5.large,c5a.large,m5.large,m5a.large,c4.large,m4.large,c3.large,m3.large).
- Subnet untuk meluncurkan instance (subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782). Masing-masing sesuai dengan Availability Zone yang berbeda.
- Template peluncuran (my-launch-template) dan versi template peluncuran (\$Latest).

Ketika Amazon EC2 Auto Scaling berupaya memenuhi kapasitas Sesuai Permintaan Anda, maka akan meluncurkan instans c5.large terlebih dahulu. Untuk kapasitas Spot Anda, Auto Scaling Amazon EC2 mencoba meluncurkan Instans Spot dari kumpulan Instans Spot dengan harga serendah mungkin, tetapi juga dengan kapasitas optimal untuk jumlah instans yang diluncurkan.

JSON

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

config.jsonFile berisi konten berikut.

```
{
  "AutoScalingGroupName": "my-asg",
  "MixedInstancesPolicy": {
    "LaunchTemplate": {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "$Latest"
      },
      "Overrides": [
        {
          "InstanceType": "c5.large"
        },
        {
          "InstanceType": "c5a.large"
        }
      ]
    }
  }
}
```

```

    },
    {
      "InstanceType": "m5.large"
    },
    {
      "InstanceType": "m5a.large"
    },
    {
      "InstanceType": "c4.large"
    },
    {
      "InstanceType": "m4.large"
    },
    {
      "InstanceType": "c3.large"
    },
    {
      "InstanceType": "m3.large"
    }
  ]
},
"InstancesDistribution": {
  "OnDemandPercentageAboveBaseCapacity": 30,
  "SpotAllocationStrategy": "price-capacity-optimized"
}
},
"MinSize": 1,
"MaxSize": 5,
"DesiredCapacity": 3,
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}

```

YAML

Atau, Anda dapat menggunakan [create-auto-scaling-group](#) perintah berikut untuk membuat grup Auto Scaling. Ini mereferensikan file YAMG sebagai satu-satunya parameter untuk grup Auto Scaling Anda.

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

config.yamlFile berisi konten berikut.

```
---
```

```

AutoScalingGroupName: my-asg
MixedInstancesPolicy:
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
      - InstanceType: c5.large
      - InstanceType: c5a.large
      - InstanceType: m5.large
      - InstanceType: m5a.large
      - InstanceType: c4.large
      - InstanceType: m4.large
      - InstanceType: c3.large
      - InstanceType: m3.large
    InstancesDistribution:
      OnDemandPercentageAboveBaseCapacity: 30
      SpotAllocationStrategy: price-capacity-optimized
  MinSize: 1
  MaxSize: 5
  DesiredCapacity: 3
  VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782

```

Konfigurasi grup Auto Scaling untuk menggunakan bobot instance

Bila Anda menggunakan beberapa jenis instans, Anda dapat menentukan berapa banyak unit yang akan dikaitkan dengan setiap jenis instans, lalu menentukan kapasitas grup Anda dengan unit pengukuran yang sama. Opsi spesifikasi kapasitas ini dikenal sebagai bobot.

Sebagai contoh, katakanlah Anda menjalankan aplikasi komputasi intensif yang berkinerja terbaik dengan setidaknya 8 vCPU dan 15 GiB RAM. Jika Anda menggunakan `c5.2xlarge` sebagai unit dasar Anda, setiap jenis instans EC2 berikut akan memenuhi kebutuhan aplikasi Anda.

Contoh jenis instans

Jenis instans	vCPU	Memori (GiB)
<code>c5.2xlarge</code>	8	16
<code>c5.4xlarge</code>	16	32
<code>c5.12xlarge</code>	48	96

Jenis instans	vCPU	Memori (GiB)
c5.18xlarge	72	144
c5.24xlarge	96	192

Secara default, semua jenis instance memiliki bobot yang sama terlepas dari ukurannya. Dengan kata lain, apakah Amazon EC2 Auto Scaling meluncurkan jenis instans besar atau kecil, setiap instans menghitung hal yang sama terhadap kapasitas yang diinginkan dari grup Auto Scaling.

Namun, dengan bobot, Anda menetapkan nilai angka yang menentukan berapa banyak unit yang akan dikaitkan dengan setiap jenis instance. Misalnya, jika instans memiliki ukuran berbeda, c5.2xlarge instans bisa memiliki berat 2, dan c5.4xlarge (yang dua kali lebih besar) dapat memiliki berat 4, dan seterusnya. Kemudian, saat Amazon EC2 Auto Scaling menskalakan grup, bobot ini diterjemahkan ke dalam jumlah unit yang dihitung setiap instans terhadap kapasitas yang Anda inginkan.

Bobot tidak mengubah jenis instans yang dipilih Amazon EC2 Auto Scaling untuk diluncurkan; sebaliknya, strategi alokasi melakukannya. Untuk informasi selengkapnya, lihat [Strategi alokasi](#).

Important

Untuk mengonfigurasi grup Auto Scaling agar memenuhi kapasitas yang diinginkan menggunakan jumlah vCPU atau jumlah memori dari setiap jenis instans, sebaiknya gunakan pemilihan tipe instans berbasis atribut. Menyetel `DesiredCapacityType` parameter secara otomatis menentukan jumlah unit yang akan dikaitkan dengan setiap jenis instance berdasarkan nilai yang Anda tetapkan untuk parameter ini. Untuk informasi selengkapnya, lihat [Buat grup instance campuran menggunakan pemilihan tipe instans berbasis atribut](#).

Daftar Isi

- [Pertimbangan](#)
- [Perilaku berat badan contoh](#)
- [Konfigurasi grup Auto Scaling untuk menggunakan bobot](#)
- [Contoh harga spot per satuan jam](#)

Pertimbangan

Bagian ini membahas pertimbangan utama untuk menerapkan bobot secara efektif.

- Pilih beberapa jenis instance yang sesuai dengan kebutuhan kinerja aplikasi Anda. Tentukan bobot yang harus dihitung setiap jenis instans terhadap kapasitas yang diinginkan dari grup Auto Scaling Anda berdasarkan kemampuannya. Bobot ini berlaku untuk instans saat ini dan masa depan.
- Hindari rentang yang besar di antara bobot. Misalnya, jangan tentukan bobot 1 untuk tipe instance ketika tipe instance berikutnya yang lebih besar memiliki bobot 200. Perbedaan antara bobot terkecil dan terbesar juga tidak boleh ekstrim. Perbedaan bobot yang ekstrim dapat berdampak negatif pada pengoptimalan biaya-kinerja.
- Tentukan kapasitas yang diinginkan grup dalam satuan, bukan instance. Misalnya, jika Anda menggunakan bobot berbasis VCPU, atur jumlah inti yang Anda inginkan dan juga minimum dan maksimum.
- Atur berat dan kapasitas yang Anda inginkan sehingga kapasitas yang diinginkan setidaknya dua hingga tiga kali lebih besar daripada berat terbesar Anda.

Perhatikan hal berikut saat memperbarui grup yang ada:

- Saat Anda menambahkan bobot ke grup yang ada, sertakan bobot untuk semua jenis instance yang sedang digunakan.
- Saat Anda menambah atau mengubah bobot, Auto Scaling Amazon EC2 akan meluncurkan atau menghentikan instans untuk mencapai kapasitas yang diinginkan berdasarkan nilai bobot baru.
- Jika Anda menghapus tipe instans, menjalankan instance dari tipe tersebut akan mempertahankan bobot terakhirnya, meskipun tidak lagi ditentukan.

Perilaku berat badan contoh

Saat Anda menggunakan bobot instans, Auto Scaling Amazon EC2 berperilaku dengan cara berikut:

- Kapasitas saat ini akan berada pada kapasitas yang diinginkan atau di atasnya. Kapasitas saat ini dapat melebihi kapasitas yang diinginkan jika instans diluncurkan yang melebihi unit kapasitas yang diinginkan yang tersisa. Misalnya, misalkan Anda menentukan dua jenis instance, `c5.2xlarge` dan `c5.12xlarge`, dan Anda menetapkan bobot instance 2 untuk `c5.2xlarge` dan 12 untuk `c5.12xlarge`. Jika ada lima unit tersisa untuk memenuhi kapasitas yang diinginkan, dan Amazon EC2 Auto Scaling menyediakan `c5.12xlarge` a, kapasitas yang diinginkan dilampaui tujuh unit.

- Saat meluncurkan instans, Auto Scaling Amazon EC2 memprioritaskan kapasitas distribusi di seluruh Availability Zone dan menghormati strategi alokasi melebihi kapasitas yang diinginkan.
- Auto Scaling Amazon EC2 dapat melebihi batas kapasitas maksimum untuk menjaga keseimbangan di seluruh Availability Zone, menggunakan strategi alokasi pilihan Anda. Batas keras yang diberlakukan oleh Amazon EC2 Auto Scaling adalah kapasitas yang Anda inginkan ditambah bobot terbesar Anda.

Konfigurasi grup Auto Scaling untuk menggunakan bobot

Anda dapat mengonfigurasi grup Auto Scaling untuk menggunakan bobot, seperti yang ditunjukkan pada contoh berikut. AWS CLI Untuk petunjuk tentang penggunaan konsol, lihat [Buat grup instance campuran dengan memilih jenis instance secara manual](#).

Untuk mengonfigurasi grup Auto Scaling baru untuk menggunakan bobot ()AWS CLI

Gunakan perintah [create-auto-scaling-group](#). Misalnya, perintah berikut membuat grup Auto Scaling baru dan menetapkan bobot dengan menentukan hal berikut:

- Persentase grup yang akan diluncurkan sebagai Instans Sesuai Permintaan () 0
- Strategi alokasi untuk Spot Instance di setiap Availability Zone (*capacity-optimized*)
- Jenis instans untuk diluncurkan dalam urutan prioritas (*m4.16xlarge*, *m5.24xlarge*)
- Berat contoh yang sesuai dengan perbedaan ukuran relatif (vCPU) antara jenis contoh (16, 24)
- Subnet untuk meluncurkan instans (*subnet-5ea0c127*, *subnet-6194ea3b*, *subnet-c934b782*), masing-masing sesuai dengan Availability Zone yang berbeda
- Templat peluncuran (*my-launch-template*) dan versi templat peluncuran (*\$Latest*)

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

*config.json*File berisi konten berikut.

```
{
  "AutoScalingGroupName": "my-asg",
  "MixedInstancesPolicy": {
    "LaunchTemplate": {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "$Latest"
      }
    }
  }
}
```



```

    },
    "Overrides": [
      {
        "InstanceType": "m4.16xlarge",
        "WeightedCapacity": "16"
      },
      {
        "InstanceType": "m5.24xlarge",
        "WeightedCapacity": "24"
      }
    ]
  },
  "InstancesDistribution": {
    "OnDemandPercentageAboveBaseCapacity": 0,
    "SpotAllocationStrategy": "capacity-optimized"
  }
},
"MinSize": 160,
"MaxSize": 720,
"DesiredCapacity": 480,
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
"Tags": []
}

```

Untuk mengonfigurasi grup Auto Scaling yang ada untuk menggunakan bobot ()AWS CLI

Gunakan perintah [update-auto-scaling-group](#). Misalnya, perintah berikut menetapkan bobot ke tipe instance dalam grup Auto Scaling yang ada dengan menentukan hal berikut:

- Jenis instans untuk diluncurkan dalam urutan prioritas (c5.18xlarge, c5.24xlarge, c5.2xlarge, c5.4xlarge)
- Berat instans yang sesuai dengan perbedaan ukuran relatif (vCPU) antara jenis contoh (18, 24, 2, 4)
- Kapasitas ditingkatkan baru yang diinginkan, yang lebih besar daripada berat terbesar

```
aws autoscaling update-auto-scaling-group --cli-input-json file://~/config.json
```

config.jsonFile berisi konten berikut.

```
{
```

```

"AutoScalingGroupName": "my-existing-asg",
"MixedInstancesPolicy": {
  "LaunchTemplate": {
    "Overrides": [
      {
        "InstanceType": "c5.18xlarge",
        "WeightedCapacity": "18"
      },
      {
        "InstanceType": "c5.24xlarge",
        "WeightedCapacity": "24"
      },
      {
        "InstanceType": "c5.2xlarge",
        "WeightedCapacity": "2"
      },
      {
        "InstanceType": "c5.4xlarge",
        "WeightedCapacity": "4"
      }
    ]
  }
},
"MinSize": 0,
"MaxSize": 100,
"DesiredCapacity": 100
}

```

Untuk memverifikasi bobot menggunakan baris perintah

Gunakan salah satu perintah berikut:

- [describe-auto-scaling-groups](#) (AWS CLI)
- [Dapatkan-sebagai \(AutoScalingGroup\)](#) AWS Tools for Windows PowerShell

Contoh harga spot per satuan jam

Tabel berikut membandingkan harga per jam untuk Instans Spot di Zona Ketersediaan yang berbeda di AS Timur (Virginia Utara) dengan harga Instans Sesuai Permintaan di Wilayah yang sama. Harga yang ditampilkan adalah contoh harga dan bukan harga saat ini. Ini adalah biaya Anda per jam instans.

Contoh: Harga spot per jam instans

Jenis instans	us-east-1a	us-east-1b	us-east-1c	Penetapan harga Sesuai Permintaan
c5.2xlarge	0,180 USD	0,191 USD	0,170 USD	0,34 USD
c5.4xlarge	0,341 USD	0,361 USD	0,318 USD	0,68 USD
c5.12xlarge	0,779 USD	0,777 USD	0,777 USD	2,04 USD
c5.18xlarge	1,207 USD	1,475 USD	1,357 USD	3,06 USD
c5.24xlarge	1,555 USD	1,555 USD	1,555 USD	4,08 USD

Dengan bobot instance, Anda dapat mengevaluasi biaya Anda berdasarkan apa yang Anda gunakan per satuan jam. Anda dapat menentukan harga per unit jam dengan membagi harga Anda untuk satu jenis instans dengan jumlah unit yang diwakilinya. Untuk Instans Sesuai Permintaan, harga per unit jam sama ketika menerapkan satu jenis instans yang sama dengan saat menerapkan ukuran yang berbeda dari jenis instans yang sama. Sebaliknya, harga Spot per unit jam bervariasi berdasarkan kumpulan Spot.

Contoh berikut menunjukkan bagaimana perhitungan harga Spot per satuan jam bekerja dengan bobot instance. Untuk memudahkan perhitungan, katakanlah Anda ingin meluncurkan Instans Spot hanya di us-east-1a. Harga per satuan jam ditangkap dalam tabel berikut.

Contoh: Contoh Harga Spot per satuan jam

Jenis instans	us-east-1a	Berat instans	Harga per unit jam
c5.2xlarge	0,180 USD	2	0,090 USD
c5.4xlarge	0,341 USD	4	0,085 USD

Jenis instans	us-east-1a	Berat instans	Harga per unit jam
c5.12xlarge	0,779 USD	12	0,065 USD
c5.18xlarge	1,207 USD	18	0,067 USD
c5.24xlarge	1,555 USD	24	0,065 USD

Gunakan template peluncuran yang berbeda untuk jenis instans

Selain menggunakan beberapa jenis instance, Anda juga dapat menggunakan beberapa templat peluncuran.

Misalnya, Anda mengonfigurasi grup Auto Scaling untuk aplikasi intensif komputasi dan ingin menyertakan campuran tipe instans C5, C5a, dan C6g. Namun, instance C6g menampilkan prosesor AWS Graviton berdasarkan arsitektur Arm 64-bit, sedangkan instance C5 dan C5a berjalan pada prosesor Intel x86 64-bit. AMI untuk instance C5 dan C5a keduanya berfungsi pada masing-masing instance tersebut, tetapi tidak pada instance C6g. Untuk mengatasi masalah ini, gunakan template peluncuran yang berbeda untuk instance C6g. Anda masih dapat menggunakan template peluncuran yang sama untuk instance C5 dan C5a.

Bagian ini berisi prosedur untuk menggunakan AWS CLI untuk melakukan tugas-tugas yang terkait dengan menggunakan beberapa template peluncuran. Saat ini, fitur ini hanya tersedia jika Anda menggunakan AWS CLI atau SDK, dan tidak tersedia dari konsol.

Daftar Isi

- [Konfigurasi grup Auto Scaling untuk menggunakan beberapa templat peluncuran](#)
- [Sumber daya terkait](#)

Konfigurasi grup Auto Scaling untuk menggunakan beberapa templat peluncuran

Anda dapat mengonfigurasi grup Auto Scaling untuk menggunakan beberapa templat peluncuran, seperti yang ditunjukkan pada contoh berikut.

Untuk mengonfigurasi grup Auto Scaling baru untuk menggunakan beberapa templat peluncuran
()AWS CLI

Gunakan perintah [create-auto-scaling-group](#). Misalnya, perintah berikut membuat grup Auto Scaling baru. Ini menentukan tipe `c5.large`, `c5a.large`, dan `c6g.large` instance dan mendefinisikan

template peluncuran baru untuk jenis `c6g.large` instance guna memastikan bahwa AMI yang sesuai digunakan untuk meluncurkan instance ARM. Auto Scaling Amazon EC2 menggunakan urutan jenis instans untuk menentukan jenis instans mana yang akan digunakan terlebih dahulu saat memenuhi kapasitas Sesuai Permintaan.

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

`config.json` berisi konten berikut.

```
{
  "AutoScalingGroupName": "my-asg",
  "MixedInstancesPolicy": {
    "LaunchTemplate": {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template-for-x86",
        "Version": "$Latest"
      },
    },
    "Overrides": [
      {
        "InstanceType": "c6g.large",
        "LaunchTemplateSpecification": {
          "LaunchTemplateName": "my-launch-template-for-arm",
          "Version": "$Latest"
        }
      },
      {
        "InstanceType": "c5.large"
      },
      {
        "InstanceType": "c5a.large"
      }
    ]
  },
  "InstancesDistribution": {
    "OnDemandBaseCapacity": 1,
    "OnDemandPercentageAboveBaseCapacity": 50,
    "SpotAllocationStrategy": "capacity-optimized"
  },
  "MinSize": 1,
  "MaxSize": 5,
  "DesiredCapacity": 3,
}
```

```

"VPCZoneIdentifier": "subnet-5ea0c127, subnet-6194ea3b, subnet-c934b782",
"Tags": [ ]
}

```

Untuk mengonfigurasi grup Auto Scaling yang ada untuk menggunakan beberapa templat peluncuran (AWS CLI)

Gunakan perintah [update-auto-scaling-group](#). Misalnya, perintah berikut menetapkan template peluncuran bernama *my-launch-template-for-arm* ke jenis *c6g.large* instance untuk grup Auto Scaling bernama *my-asg*

```
aws autoscaling update-auto-scaling-group --cli-input-json file://~/config.json
```

`config.json` berisi konten berikut.

```

{
  "AutoScalingGroupName": "my-asg",
  "MixedInstancesPolicy": {
    "LaunchTemplate": {
      "Overrides": [
        {
          "InstanceType": "c6g.large",
          "LaunchTemplateSpecification": {
            "LaunchTemplateName": "my-launch-template-for-arm",
            "Version": "$Latest"
          }
        },
        {
          "InstanceType": "c5.large"
        },
        {
          "InstanceType": "c5a.large"
        }
      ]
    }
  }
}

```

Untuk memverifikasi template peluncuran untuk grup Auto Scaling

Gunakan salah satu perintah berikut:

- [describe-auto-scaling-groups](#) (AWS CLI)
- [Dapatkan-sebagai \(AutoScalingGroup\)](#) AWS Tools for Windows PowerShell

Sumber daya terkait

[Anda dapat menemukan contoh menentukan beberapa templat peluncuran menggunakan pemilihan jenis instans berbasis atribut dalam AWS CloudFormation templat di re:Post.AWS](#)

Buat grup Auto Scaling menggunakan konfigurasi peluncuran

Important

Anda tidak dapat menelepon `CreateLaunchConfiguration` dengan jenis instans Amazon EC2 baru yang dirilis setelah 31 Desember 2022. Selain itu, akun baru apa pun yang dibuat pada atau setelah 1 Juni 2023 tidak akan memiliki opsi untuk membuat konfigurasi peluncuran baru melalui konsol. Di masa depan, akun baru tidak akan dapat membuat konfigurasi peluncuran baru dengan menggunakan konsol, API, CLI, dan CloudFormation Migrasi untuk meluncurkan template untuk memastikan bahwa Anda tidak perlu membuat konfigurasi peluncuran baru sekarang atau di masa mendatang. Untuk informasi tentang memigrasi grup Auto Scaling untuk meluncurkan templat, lihat [Migrasi grup Auto Scaling Anda untuk meluncurkan template](#)

Jika Anda telah membuat konfigurasi peluncuran atau instans EC2, Anda dapat membuat grup Auto Scaling yang menggunakan konfigurasi peluncuran sebagai templat konfigurasi untuk instans EC2-nya. Konfigurasi peluncuran menentukan informasi seperti ID AMI, tipe instans, key pair, grup keamanan, dan pemetaan perangkat blok untuk instans Anda. Untuk informasi tentang membuat konfigurasi peluncuran, lihat [Buat konfigurasi peluncuran](#).

Anda harus memiliki izin yang cukup untuk membuat grup Auto Scaling. Anda juga harus memiliki izin yang cukup untuk membuat peran terkait layanan yang digunakan Auto Scaling Amazon EC2 untuk melakukan tindakan atas nama Anda jika belum ada. Untuk contoh kebijakan IAM yang dapat digunakan administrator sebagai referensi untuk memberikan izin kepada Anda, lihat [Contoh kebijakan berbasis identitas](#)

Daftar Isi

- [Buat grup Auto Scaling menggunakan konfigurasi peluncuran](#)

- [Buat grup Auto Scaling menggunakan parameter dari instance yang ada](#)

Buat grup Auto Scaling menggunakan konfigurasi peluncuran

Important

Kami memberikan informasi tentang konfigurasi peluncuran untuk pelanggan yang belum bermigrasi dari konfigurasi peluncuran ke templat peluncuran. Untuk informasi tentang memigrasi grup Auto Scaling untuk meluncurkan templat, lihat [Migrasi grup Auto Scaling Anda untuk meluncurkan template](#)

Saat membuat grup Auto Scaling, Anda harus menentukan informasi yang diperlukan untuk mengonfigurasi instans Amazon EC2, Availability Zone, dan subnet VPC untuk instans, kapasitas yang diinginkan, dan batas kapasitas minimum dan maksimum.

Prosedur berikut menunjukkan cara membuat grup Auto Scaling menggunakan konfigurasi peluncuran. Anda tidak dapat mengubah konfigurasi peluncuran setelah dibuat, tetapi Anda dapat mengganti konfigurasi peluncuran untuk grup Auto Scaling. Untuk informasi selengkapnya, lihat [Mengubah konfigurasi peluncuran untuk grup Auto Scaling](#).

Prasyarat

- Anda harus telah membuat konfigurasi peluncuran. Untuk informasi selengkapnya, lihat [Buat konfigurasi peluncuran](#).

Untuk membuat grup Auto Scaling menggunakan konfigurasi peluncuran (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pada bilah navigasi di bagian atas layar, pilih yang sama dengan Wilayah AWS yang Anda gunakan saat membuat konfigurasi peluncuran.
3. Pilih Buat grup Auto Scaling.
4. Pada halaman Pilih templat peluncuran atau konfigurasi, untuk Nama grup Auto Scaling, masukkan nama untuk grup Auto Scaling Anda.
5. Untuk memilih konfigurasi peluncuran, lakukan hal berikut:

- a. Untuk template Luncurkan, pilih Beralih untuk meluncurkan konfigurasi.
 - b. Untuk Konfigurasi peluncuran, pilih konfigurasi peluncuran yang ada.
 - c. Verifikasi bahwa konfigurasi peluncuran Anda mendukung semua opsi yang Anda rencanakan untuk digunakan, lalu pilih Selanjutnya.
6. Pada halaman Configure instance launch options, di bawah Jaringan, untuk VPC, pilih VPC. Grup Auto Scaling harus dibuat dalam VPC yang sama dengan grup keamanan yang Anda tentukan dalam konfigurasi peluncuran.
7. Untuk Zona Ketersediaan dan subnet, pilih satu atau lebih subnet di VPC yang ditentukan. Gunakan subnet di beberapa Availability Zone untuk ketersediaan tinggi. Untuk informasi selengkapnya, lihat [Pertimbangan saat memilih subnet VPC](#).
8. Pilih Berikutnya.

Atau, Anda dapat menerima pengaturan default lainnya, dan memilih Lewati untuk meninjau.

9. (Opsional) Pada Konfigurasi opsi lanjutan, konfigurasi opsi berikut, lalu pilih Selanjutnya:
- a. Untuk mendaftarkan instans Amazon EC2 Anda dengan penyeimbang beban, pilih penyeimbang beban yang ada atau buat yang baru. Untuk informasi selengkapnya, lihat [Gunakan Elastic Load Balancing untuk mendistribusikan lalu lintas di seluruh instans dalam grup Auto Scaling](#). Untuk membuat penyeimbang beban baru, ikuti prosedur di [Konfigurasi Application Load Balancer atau Network Load Balancer dari konsol Auto Scaling Amazon EC2](#).
 - b. (Opsional) Untuk pemeriksaan Kesehatan, Jenis pemeriksaan kesehatan tambahan, pilih Aktifkan pemeriksaan kesehatan Elastic Load Balancing.
 - c. (Opsional) Untuk Health check masa tenggang, masukkan jumlah waktu, dalam hitungan detik. Ini adalah berapa lama Amazon EC2 Auto Scaling perlu menunggu sebelum memeriksa status kesehatan suatu instans setelah memasuki status. InService Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).
 - d. Di bawah Pengaturan tambahan, Pemantauan, pilih apakah akan mengaktifkan pengumpulan metrik CloudWatch grup. Metrik ini memberikan pengukuran yang dapat menjadi indikator potensi masalah, seperti jumlah instance penghentian atau jumlah instance yang tertunda. Untuk informasi selengkapnya, lihat [Pantau CloudWatch metrik untuk grup dan instans Auto Scaling](#).

- e. Untuk Aktifkan pemanasan instance default, pilih opsi ini dan pilih waktu pemanasan untuk aplikasi Anda. Jika Anda membuat grup Auto Scaling yang memiliki kebijakan penskalaan, fitur pemanasan instans default akan meningkatkan CloudWatch metrik Amazon yang digunakan untuk penskalaan dinamis. Untuk informasi selengkapnya, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#).
10. (Opsional) Di halaman Konfigurasi ukuran grup dan kebijakan penskalaan, konfigurasi opsi berikut, lalu pilih Selanjutnya:
 - a. Di bawah Ukuran grup, untuk kapasitas yang diinginkan, masukkan jumlah instans awal yang akan diluncurkan.
 - b. Di bagian Penskalaan, di bawah batas Penskalaan, jika nilai baru Anda untuk kapasitas yang diinginkan lebih besar dari kapasitas yang diinginkan Min dan kapasitas yang diinginkan Maks, kapasitas Maks yang diinginkan secara otomatis ditingkatkan ke nilai kapasitas baru yang diinginkan. Anda dapat mengubah batasan ini sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).
 - c. Untuk Penskalaan otomatis, pilih apakah Anda ingin membuat kebijakan penskalaan pelacakan target. Anda juga dapat membuat kebijakan ini setelah membuat grup Auto Scaling.


Jika Anda memilih kebijakan penskalaan pelacakan target, ikuti petunjuk [Buat kebijakan penskalaan pelacakan target](#) untuk membuat kebijakan.
 - d. Untuk kebijakan pemeliharaan Instance, pilih apakah Anda ingin membuat kebijakan pemeliharaan instans. Anda juga dapat membuat kebijakan ini setelah membuat grup Auto Scaling. Ikuti petunjuk [Tetapkan kebijakan pemeliharaan instans](#) untuk membuat kebijakan.
 - e. Di bawah Perlindungan penskalaan instans, pilih apakah akan mengaktifkan perlindungan skala instans. Untuk informasi selengkapnya, lihat [Gunakan perlindungan skala dalam instance](#).
 11. (Opsional) Untuk menerima notifikasi, untuk Tambahkan notifikasi, konfigurasi notifikasi, lalu pilih Selanjutnya. Untuk informasi selengkapnya, lihat [Opsi pemberitahuan Amazon SNS untuk Auto Scaling Amazon EC2](#).
 12. (Opsional) Untuk menambahkan tag, pilih Tambahkan tag, berikan kunci tag dan nilai tag untuk setiap tag, lalu pilih Selanjutnya. Untuk informasi selengkapnya, lihat [Menandai grup dan instance Auto Scaling](#).
 13. Di halaman Peninjauan, pilih Buat grup Auto Scaling.

Untuk membuat grup Auto Scaling menggunakan jalur perintah

Anda dapat menggunakan salah satu perintah berikut:

- [create-auto-scaling-group](#) (AWS CLI)
- [Baru-AS \(AutoScalingGroup\)](#) AWS Tools for Windows PowerShell

Buat grup Auto Scaling menggunakan parameter dari instance yang ada

 Important

Kami memberikan informasi tentang konfigurasi peluncuran untuk pelanggan yang belum bermigrasi dari konfigurasi peluncuran ke templat peluncuran. Untuk informasi tentang memigrasi grup Auto Scaling untuk meluncurkan templat, lihat [Migrasi grup Auto Scaling Anda untuk meluncurkan template](#)

Jika ini adalah pertama kalinya Anda membuat grup Auto Scaling, kami sarankan Anda menggunakan konsol untuk membuat template peluncuran dari instans EC2 yang ada. Lalu gunakan templat peluncuran untuk membuat grup Auto Scaling baru. Untuk prosedur ini, lihat [Membuat grup Auto Scaling menggunakan wizard peluncuran Amazon EC2](#).

Prosedur berikut menunjukkan cara membuat grup Auto Scaling dengan menentukan instance yang ada untuk digunakan sebagai dasar untuk meluncurkan instance lain. Beberapa parameter diperlukan untuk membuat instans EC2, seperti ID Amazon Machine Image (AMI), tipe instans, key pair, dan grup keamanan. Semua informasi ini juga digunakan oleh Amazon EC2 Auto Scaling untuk meluncurkan instans atas nama Anda ketika ada kebutuhan untuk skala. Informasi ini disimpan baik dalam template peluncuran atau konfigurasi peluncuran.

Saat Anda menggunakan instans yang ada, Auto Scaling Amazon EC2 akan membuat grup Auto Scaling yang meluncurkan instans berdasarkan konfigurasi peluncuran yang dibuat pada saat yang bersamaan. Konfigurasi peluncuran baru memiliki nama yang sama dengan grup Auto Scaling, dan mencakup detail konfigurasi tertentu dari instance yang diidentifikasi.

Detail konfigurasi berikut disalin dari instance yang diidentifikasi ke dalam konfigurasi peluncuran:

- ID AMI
- Jenis instans

- Pasangan kunci
- Grup keamanan
- Jenis alamat IP (publik atau pribadi)
- Profil instans IAM, jika berlaku
- Pemantauan (benar atau salah)
- EBS dioptimalkan (benar atau salah)
- Pengaturan penghunian, jika diluncurkan ke VPC (bersama atau khusus)
- ID Kernel dan ID disk RAM, jika ada
- Data pengguna, jika ditentukan
- Harga spot (maksimum)

Subnet VPC dan Availability Zone disalin dari instance yang diidentifikasi ke definisi sumber daya grup Auto Scaling sendiri.

Jika instance yang diidentifikasi berada dalam grup penempatan, grup Auto Scaling baru akan meluncurkan instance ke grup penempatan yang sama dengan instance yang diidentifikasi. Karena pengaturan konfigurasi peluncuran tidak mengizinkan grup penempatan ditentukan, grup penempatan disalin ke PlacementGroup atribut grup Auto Scaling yang baru.

Detail konfigurasi berikut tidak disalin dari instance yang Anda identifikasi:

- Penyimpanan: Perangkat blok (volume EBS dan volume penyimpanan instans) tidak disalin dari instance yang diidentifikasi. Sebaliknya, pemetaan perangkat blok yang dibuat sebagai bagian dari pembuatan AMI menentukan perangkat mana yang digunakan.
- Jumlah antarmuka jaringan: Antarmuka jaringan tidak disalin dari instance yang Anda identifikasi. Sebagai gantinya, Amazon EC2 Auto Scaling menggunakan pengaturan defaultnya untuk membuat satu antarmuka jaringan, yang merupakan antarmuka jaringan utama (eth0).
- Opsi metadata instans: Pengaturan batas hop hop yang dapat diakses metadata, versi metadata, dan respons token tidak disalin dari instance yang diidentifikasi. Sebagai gantinya, Amazon EC2 Auto Scaling menggunakan pengaturan defaultnya. Untuk informasi selengkapnya, lihat [Mengonfigurasi opsi metadata instans](#).
- Load balancer: Jika instans yang diidentifikasi terdaftar dengan satu atau beberapa penyeimbang beban, informasi tentang penyeimbang beban tidak disalin ke penyeimbang beban atau atribut grup target dari grup Auto Scaling yang baru.

- **Tag:** Jika instance yang diidentifikasi memiliki tag, tag tidak disalin ke Tags atribut grup Auto Scaling yang baru.

Prasyarat

Instans EC2 harus memenuhi kriteria berikut:

- Instance ini bukanlah anggota dari grup Auto Scaling lainnya.
- Instans berada dalam status `running`.
- AMI yang digunakan untuk meluncurkan instans harus tetap ada.

Buat grup Auto Scaling dari instans EC2 (konsol)

Untuk membuat grup Auto Scaling dari instans EC2

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pada panel navigasi, di bawah Instans, pilih Instans, lalu pilih instance.
3. Pilih Tindakan, Pengaturan instance, Lampirkan ke Grup Auto Scaling.
4. Pada halaman Melampirkan ke grup Auto Scaling, untuk Grup Auto Scaling, masukkan nama untuk grup, lalu pilih Lampirkan.

Setelah instance dilampirkan, itu dianggap sebagai bagian dari grup Auto Scaling. Grup Auto Scaling baru dibuat menggunakan konfigurasi peluncuran baru dengan nama yang sama yang Anda tentukan untuk grup Auto Scaling. Grup Auto Scaling memiliki kapasitas yang diinginkan dan ukuran maksimum. 1

5. (Opsional) Untuk mengedit pengaturan grup Auto Scaling, pada panel navigasi, di bawah Auto Scaling, pilih Grup Auto Scaling. Pilih kotak centang di samping grup Auto Scaling baru, pilih tombol Edit yang berada di atas daftar grup, ubah pengaturan sesuai kebutuhan, lalu pilih Perbarui.

Buat grup Auto Scaling dari instans EC2 () AWS CLI

Prosedur berikut menunjukkan cara menggunakan perintah CLI untuk membuat grup Auto Scaling dari instance EC2.

Prosedur ini tidak menambahkan instance ke grup Auto Scaling. Untuk instance yang akan dilampirkan, Anda harus menjalankan perintah [attach-instance](#) setelah grup Auto Scaling dibuat.

[Sebelum memulai, cari ID instans EC2 menggunakan konsol Amazon EC2 atau perintah `describe-instances`.](#)

Untuk menggunakan instans Anda saat ini sebagai templat

- Gunakan perintah [create-auto-scaling-group](#) berikut untuk membuat grup Auto Scaling, `my-asg-from-instance`, dari instans EC2 `i-0e69cc3f05f825f4f`.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg-from-instance \  
  --instance-id i-0e69cc3f05f825f4f --min-size 1 --max-size 2 --desired-capacity 2
```

Untuk memverifikasi bahwa grup Auto Scaling Anda telah meluncurkan instans

- Gunakan perintah [describe-auto-scaling-groups](#) berikut untuk memverifikasi bahwa grup Auto Scaling berhasil dibuat.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg-from-instance
```

Contoh respons berikut menunjukkan bahwa kapasitas yang diinginkan grup adalah 2 dan grup tersebut memiliki 2 instans yang berjalan, dan konfigurasi peluncuran dinamai `my-asg-from-instance`.

```
{  
  "AutoScalingGroups": [  
    {  
      "AutoScalingGroupName": "my-asg-from-instance",  
      "AutoScalingGroupARN": "arn",  
      "LaunchConfigurationName": "my-asg-from-instance",  
      "MinSize": 1,  
      "MaxSize": 2,  
      "DesiredCapacity": 2,  
      "DefaultCooldown": 300,  
      "AvailabilityZones": [  
        "us-west-2a"  
      ],  
    },  
  ],  
}
```

```
"LoadBalancerNames": [],
"TargetGroupARNs": [],
"HealthCheckType": "EC2",
"HealthCheckGracePeriod": 0,
"Instances": [
  {
    "InstanceId": "i-06905f55584de02da",
    "InstanceType": "t2.micro",
    "AvailabilityZone": "us-west-2a",
    "LifecycleState": "InService",
    "HealthStatus": "Healthy",
    "LaunchConfigurationName": "my-asg-from-instance",
    "ProtectedFromScaleIn": false
  },
  {
    "InstanceId": "i-087b42219468eacde",
    "InstanceType": "t2.micro",
    "AvailabilityZone": "us-west-2a",
    "LifecycleState": "InService",
    "HealthStatus": "Healthy",
    "LaunchConfigurationName": "my-asg-from-instance",
    "ProtectedFromScaleIn": false
  }
],
"CreatedTime": "2020-10-28T02:39:22.152Z",
"SuspendedProcesses": [],
"VPCZoneIdentifier": "subnet-6bea5f06",
"EnabledMetrics": [],
"Tags": [],
"TerminationPolicies": [
  "Default"
],
"NewInstancesProtectedFromScaleIn": false,
"ServiceLinkedRoleARN": "arn",
"TrafficSources": []
}
]
```

Untuk melihat konfigurasi peluncuran

- Gunakan perintah [describe-launch-configurations](#) berikut untuk melihat detail konfigurasi peluncuran.

```
aws autoscaling describe-launch-configurations --launch-configuration-names my-asg-from-instance
```

Berikut ini adalah output contoh:

```
{
  "LaunchConfigurations": [
    {
      "LaunchConfigurationName": "my-asg-from-instance",
      "LaunchConfigurationARN": "arn",
      "ImageId": "ami-0528a5175983e7f28",
      "KeyName": "my-key-pair-uswest2",
      "SecurityGroups": [
        "sg-05eaec502fcdadc2e"
      ],
      "ClassicLinkVPCSecurityGroups": [ ],
      "UserData": "",
      "InstanceType": "t2.micro",
      "KernelId": "",
      "RamdiskId": "",
      "BlockDeviceMappings": [ ],
      "InstanceMonitoring": {
        "Enabled": true
      },
      "CreatedTime": "2020-10-28T02:39:22.321Z",
      "EbsOptimized": false,
      "AssociatePublicIpAddress": true
    }
  ]
}
```

Untuk menghentikan instans

- Jika Anda tidak lagi membutuhkan instans, Anda dapat mengakhirinya. Perintah [terminate-instance](#) berikut mengakhiri instance. *i-0e69cc3f05f825f4f*


```
aws ec2 terminate-instances --instance-ids i-0e69cc3f05f825f4f
```

Setelah Anda mengakhiri instans Amazon EC2, Anda tidak dapat memulai ulang instans tersebut. Setelah pengakhiran, datanya hilang dan volumenya tidak dapat dilampirkan ke instans apa pun. Untuk mempelajari selengkapnya tentang menghentikan instans, lihat [Mengakhiri instance](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Memperbarui grup Auto Scaling

Anda dapat memperbarui sebagian besar detail grup Auto Scaling Anda. Anda tidak dapat memperbarui nama grup Auto Scaling atau mengubahnya. Wilayah AWS

Untuk memperbarui grup Auto Scaling (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih grup Auto Scaling Anda untuk menampilkan informasi tentang grup, dengan tab untuk Detail, Aktivitas, Penskalaan otomatis, Manajemen instans, Pemantauan, dan Penyegaran Instance.
3. Pilih tab untuk area konfigurasi yang Anda minati dan perbarui pengaturan sesuai kebutuhan. Untuk setiap pengaturan yang Anda edit, pilih Perbarui untuk menyimpan perubahan Anda ke konfigurasi grup Auto Scaling.

- Tab Detail

Ini adalah pengaturan umum untuk grup Auto Scaling Anda. Anda dapat mengedit dan mengelola ini dengan cara yang sama seperti saat pembuatan grup Auto Scaling.

Bagian Konfigurasi lanjutan memiliki beberapa opsi yang tidak tersedia saat membuat grup seperti [kebijakan penghentian](#), [cooldown](#), [proses yang ditangguhkan](#), dan masa pakai [instans maksimum](#). Anda juga dapat melihat, tetapi tidak mengedit grup penempatan dan [peran terkait layanan](#) dari grup Auto Scaling.

Jika grup dikaitkan dengan sumber daya Elastic Load Balancing, lihat [Menambahkan dan menghapus Availability Zones](#) sebelum mengubah Availability Zones. Beberapa batasan pada penyeimbang beban mungkin mencegah Anda menerapkan perubahan pada Availability Zone grup Anda ke Availability Zone penyeimbang beban Anda.

- Tab aktivitas
 - Pemberitahuan aktivitas - Pemberitahuan [Amazon SNS](#)
- Tab penskalaan otomatis
 - Kebijakan penskalaan dinamis — Kebijakan [penskalaan dinamis](#)
 - Kebijakan penskalaan prediktif — Kebijakan penskalaan [prediktif](#)
 - Tindakan terjadwal - [Tindakan terjadwal](#)
- Tab manajemen instans
 - Kait siklus hidup - Kait [Siklus Hidup](#)
 - Kolam hangat — [Kolam hangat](#)
- Tab pemantauan
 - Hanya ada satu opsi di tab ini, yang memungkinkan Anda mengaktifkan atau menonaktifkan [CloudWatchkumpulan metrik](#) grup.

Untuk memperbarui grup Auto Scaling menggunakan baris perintah

Anda dapat menggunakan salah satu perintah berikut:

- [update-auto-scaling-group](#) (AWS CLI)
- [Perbaruan-AS \(\) AutoScalingGroup](#) AWS Tools for Windows PowerShell

Perbarui instans Auto Scaling

Jika Anda mengaitkan template peluncuran baru atau konfigurasi peluncuran dengan grup Auto Scaling, semua instance baru akan mendapatkan konfigurasi yang diperbarui. Instance yang ada terus berjalan dengan konfigurasi yang awalnya diluncurkan. Untuk menerapkan perubahan pada instans yang ada, Anda memiliki opsi berikut:

- Mulai penyegaran instance untuk mengganti instance yang lebih lama. Untuk informasi selengkapnya, lihat [Menggunakan penyegaran instans untuk memperbarui instance dalam grup Auto Scaling](#).
- [Tunggu aktivitas penskalaan untuk secara bertahap mengganti instans lama dengan instans yang lebih baru berdasarkan kebijakan penghentian Anda](#).
- Hentikan secara manual sehingga digantikan oleh grup Auto Scaling Anda.

Note

Anda dapat mengubah atribut instance berikut dengan menentukannya sebagai bagian dari template peluncuran atau konfigurasi peluncuran:

- Amazon Machine Image (AMI)
- memblokir perangkat
- pasangan kunci
- tipe instans
- kelompok keamanan
- data pengguna
- pemantauan
- Profil instans IAM
- penyewaan penempatan
- kernel
- ramdisk
- apakah instance memiliki alamat IP publik

Menandai grup dan instance Auto Scaling

Tag adalah label atribut kustom yang Anda tetapkan atau yang ditetapkan ke AWS sumber daya. AWS Setiap tag memiliki dua bagian:

- Kunci tag (misalnya, `costcenterenvironment`, atau `project`)
- Bidang opsional yang dikenal sebagai nilai tag (misalnya, `111122223333` atau `production`)

Tanda membantu Anda melakukan hal berikut:

- Lacak AWS biaya Anda. Anda mengaktifkan tag ini di AWS Billing and Cost Management dasbor. AWS menggunakan tag untuk mengkategorikan biaya Anda dan mengirimkan laporan alokasi biaya bulanan kepada Anda. Untuk informasi selengkapnya, lihat [Menggunakan tanda alokasi biaya](#) dalam Panduan Pengguna AWS Billing .

- Kontrol akses ke grup Auto Scaling berdasarkan tag. Anda dapat menggunakan kondisi dalam kebijakan IAM untuk mengontrol akses ke grup Auto Scaling berdasarkan tag pada grup tersebut. Untuk informasi selengkapnya, lihat [Tag untuk keamanan](#).
- Filter dan cari grup Auto Scaling berdasarkan tag yang Anda tambahkan. Untuk informasi selengkapnya, lihat [Gunakan tag untuk memfilter grup Auto Scaling](#).
- Identifikasi dan atur AWS sumber daya Anda. Banyak penandaan Layanan AWS dukungan, sehingga Anda dapat menetapkan tag yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya terkait.

Anda dapat menandai grup Auto Scaling baru atau yang sudah ada. Anda juga dapat menyebarkan tag dari grup Auto Scaling ke instans EC2 yang diluncurkan.

Tag tidak disebarkan ke volume Amazon EBS. Untuk menambahkan tag ke volume Amazon EBS, tentukan tag dalam template peluncuran. Untuk informasi selengkapnya, lihat [Membuat template peluncuran untuk grup Auto Scaling](#).

Anda dapat membuat dan mengelola tag melalui AWS Management Console, AWS CLI, atau SDK.

Daftar Isi

- [Membatasi penamaan dan penggunaan tag](#)
- [Siklus hidup penandaan instans EC2](#)
- [Tandai grup Auto Scaling Anda](#)
- [Hapus tag](#)
- [Tag untuk keamanan](#)
- [Kontrol akses ke tag](#)
- [Gunakan tag untuk memfilter grup Auto Scaling](#)

Membatasi penamaan dan penggunaan tag

Batasan dasar berikut berlaku untuk tanda:

- Jumlah maksimum tag per sumber daya adalah 50.
- Jumlah tag maksimum yang dapat Anda tambahkan atau hapus menggunakan satu panggilan adalah 25.

- Panjang kunci maksimum adalah 128 karakter Unicode.
- Panjang nilai maksimum adalah 256 karakter Unicode.
- Kunci dan nilai tanda peka huruf besar dan kecil. Sebagai praktik terbaik, putuskan strategi untuk memanfaatkan tag dan terapkan strategi tersebut secara konsisten di semua jenis sumber daya.
- Jangan gunakan `aws` : awalan dalam nama atau nilai tag Anda, karena ini dicadangkan untuk AWS digunakan. Anda tidak dapat mengedit atau menghapus nama atau nilai tag dengan awalan ini, dan mereka tidak dihitung terhadap tag Anda per kuota sumber daya.

Siklus hidup penandaan instans EC2

Jika Anda memilih untuk menyebarkan tag ke instans EC2 Anda, tag dikelola sebagai berikut:

- Ketika grup Auto Scaling meluncurkan instans, ini menambahkan tag ke instans selama pembuatan sumber daya alih-alih setelah sumber daya dibuat.
- Grup Auto Scaling secara otomatis menambahkan tag ke instance dengan kunci `aws:autoscaling:groupName` dan nilai nama grup Auto Scaling.
- Jika Anda menentukan tag instance di template peluncuran dan Anda memilih untuk menyebarkan tag grup Anda ke instansnya, semua tag digabungkan. Jika kunci tag yang sama ditentukan untuk tag di template peluncuran dan tag di grup Auto Scaling Anda, maka nilai tag dari grup akan diutamakan.
- Saat Anda melampirkan instans yang ada, grup Auto Scaling menambahkan tag ke instans, menimpa tag yang ada dengan kunci tag yang sama. Ini juga menambahkan tag dengan kunci `aws:autoscaling:groupName` dan nilai nama grup Auto Scaling.
- Ketika Anda memisahkan suatu instans dari grup Auto Scaling, hal itu hanya menghapus tag `aws:autoscaling:groupName`.

Tandai grup Auto Scaling Anda

Saat Anda menambahkan tag ke grup Auto Scaling, Anda dapat menentukan apakah tag tersebut harus ditambahkan ke instans yang diluncurkan di grup Auto Scaling. Jika Anda memodifikasi tag, versi tag yang diperbarui ditambahkan ke instans yang diluncurkan dalam grup Auto Scaling setelah perubahan. Jika Anda membuat atau memodifikasi tag untuk grup Auto Scaling, perubahan ini tidak dilakukan pada instans yang sudah berjalan di grup Auto Scaling.

Daftar Isi

- [Tambah atau ubah tag \(konsol\)](#)
- [Menambahkan atau memodifikasi tag \(AWS CLI\)](#)

Tambah atau ubah tag (konsol)

Untuk memberi tag grup Auto Scaling pada pembuatan

Saat Anda menggunakan konsol Amazon EC2 untuk membuat grup Auto Scaling, Anda dapat menentukan kunci dan nilai tag pada halaman Tambahkan tag dari panduan grup Buat Auto Scaling. Untuk menyebarkan tag ke instance yang diluncurkan di grup Auto Scaling, pastikan Anda tetap memilih opsi Tag instance baru untuk tag tersebut. Jika tidak, Anda dapat membatalkan pilihannya.

Untuk menambah atau mengubah tag untuk grup Auto Scaling yang ada

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Di tab Detail, pilih Tag, Edit.
4. Untuk memodifikasi tag yang ada, edit Kunci dan Nilai.
5. Untuk menambahkan tag baru, pilih Tambahkan tag dan edit Kunci dan Nilai. Anda dapat menyimpan Tag instans baru yang terpilih untuk menambahkan tag ke instans yang diluncurkan di grup Auto Scaling secara otomatis, dan membatalkan pilihan yang sebaliknya.
6. Setelah Anda selesai menambahkan tag, pilih Perbarui.

Menambahkan atau memodifikasi tag (AWS CLI)

Contoh berikut menunjukkan cara menggunakan tag AWS CLI untuk menambahkan saat Anda membuat grup Auto Scaling, dan untuk menambahkan atau memodifikasi tag untuk grup Auto Scaling yang ada.

Untuk memberi tag grup Auto Scaling pada pembuatan

Gunakan [create-auto-scaling-group](#) perintah untuk membuat grup Auto Scaling baru dan tambahkan tag, misalnya `environment=production`, ke grup Auto Scaling. Tag juga ditambahkan ke setiap instans yang diluncurkan dalam grup Auto Scaling.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \  
--launch-configuration-name my-launch-config --min-size 1 --max-size 3 \  
--vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \  
--tags Key=environment,Value=production,PropagateAtLaunch=true
```

Untuk membuat atau memodifikasi tag untuk grup Auto Scaling yang ada

Gunakan [create-or-update-tags](#) perintah untuk membuat atau memodifikasi tag. Misalnya, perintah berikut menambahkan tag **Name=my-asg** dan **costcenter=cc123**. Tag juga ditambahkan ke setiap instans yang diluncurkan di grup Auto Scaling setelah perubahan ini. Jika tag dengan salah satu kunci sudah ada, tag yang ada diganti. Konsol Amazon EC2 menghubungkan nama tampilan untuk setiap instans dengan nama yang ditentukan untuk kunci Name (peka huruf besar/kecil).

```
aws autoscaling create-or-update-tags \  
--tags ResourceId=my-asg,ResourceType=auto-scaling-group,Key=Name,Value=my-  
asg,PropagateAtLaunch=true \  
ResourceId=my-asg,ResourceType=auto-scaling-  
group,Key=costcenter,Value=cc123,PropagateAtLaunch=true
```

Jelaskan tag untuk grup Auto Scaling ()AWS CLI

Jika Anda ingin melihat tag yang diterapkan ke grup Auto Scaling tertentu, Anda dapat menggunakan salah satu dari perintah berikut:

- [describe-tags](#) - Anda memberikan nama grup Auto Scaling Anda untuk melihat daftar tag untuk grup yang ditentukan.

```
aws autoscaling describe-tags --filters Name=auto-scaling-group,Values=my-asg
```

Berikut ini adalah contoh respons.

```
{  
  "Tags": [  
    {  
      "ResourceType": "auto-scaling-group",  
      "ResourceId": "my-asg",  
      "PropagateAtLaunch": true,  
      "Value": "production",  
      "Key": "environment"  
    }  
  ]  
}
```

```
}
```

- [describe-auto-scaling-groups](#)— Anda memberikan nama grup Auto Scaling Anda untuk melihat atribut grup yang ditentukan, termasuk tag apa pun.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Berikut ini adalah contoh respons.

```
{
  "AutoScalingGroups": [
    {
      "AutoScalingGroupName": "my-asg",
      "AutoScalingGroupARN": "arn",
      "LaunchTemplate": {
        "LaunchTemplateId": "lt-0b97f1e282EXAMPLE",
        "LaunchTemplateName": "my-launch-template",
        "Version": "$Latest"
      },
      "MinSize": 1,
      "MaxSize": 5,
      "DesiredCapacity": 1,
      ...
      "Tags": [
        {
          "ResourceType": "auto-scaling-group",
          "ResourceId": "my-asg",
          "PropagateAtLaunch": true,
          "Value": "production",
          "Key": "environment"
        }
      ],
      ...
    }
  ]
}
```

Hapus tag

Anda dapat menghapus tag yang terkait dengan grup Auto Scaling Anda kapan saja.

Daftar Isi

- [Hapus tag \(konsol\)](#)
- [Hapus tag \(AWS CLI\)](#)

Hapus tag (konsol)

Untuk menghapus tanda

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup yang sudah ada.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Di tab Detail, pilih Tag, Edit.
4. Pilih Hapus di samping tag.
5. Pilih Perbarui.

Hapus tag (AWS CLI)

Gunakan perintah [delete-tags](#) untuk menghapus tag. Misalnya, perintah berikut menghapus tag dengan kunci **environment**.

```
aws autoscaling delete-tags --tags "ResourceId=my-asg,ResourceType=auto-scaling-group,Key=environment"
```

Anda harus menentukan kunci tag, tetapi Anda tidak perlu menentukan nilainya. Jika Anda menetapkan nilai dan nilai salah, tag tidak akan dihapus.

Tag untuk keamanan

Gunakan tag untuk memverifikasi bahwa pemohon (seperti pengguna atau peran IAM) memiliki izin untuk membuat, memodifikasi, atau menghapus grup Auto Scaling tertentu. Berikan informasi tag dalam elemen kondisi kebijakan IAM dengan menggunakan satu atau beberapa kunci kondisi berikut:

- Gunakan `autoscaling:ResourceTag/tag-key: tag-value` untuk memungkinkan (atau menolak) tindakan pengguna di Auto Scaling dengan tag khusus.

- Gunakan `aws:RequestTag/tag-key: tag-value` untuk meminta tag khusus dapat ada (atau tidak ada) dalam permintaan.
- Gunakan `aws:TagKeys [tag-key, ...]` untuk meminta kunci khusus dapat ada (atau tidak ada) dalam permintaan.

Misalnya, Anda dapat menolak akses ke semua grup Auto Scaling yang menyertakan tag dengan kunci **environment** dan nilainya **production**, seperti yang ditunjukkan pada contoh berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling>DeleteAutoScalingGroup"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {"autoscaling:ResourceTag/environment": "production"}
      }
    }
  ]
}
```

Untuk informasi selengkapnya tentang menggunakan tombol kondisi untuk mengontrol akses ke grup Auto Scaling, lihat [Cara kerja Amazon EC2 Auto Scaling dengan IAM](#)

Kontrol akses ke tag

Gunakan tag untuk memverifikasi bahwa pemohon (seperti pengguna atau peran IAM) memiliki izin untuk menambah, memodifikasi, atau menghapus tag untuk grup Auto Scaling.

Contoh berikut kebijakan IAM memberikan izin utama untuk menghapus hanya tag dengan **temporary** kunci dari grup Auto Scaling.

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

```
{
  "Effect": "Allow",
  "Action": "autoscaling:DeleteTags",
  "Resource": "*",
  "Condition": {
    "ForAllValues:StringEquals": { "aws:TagKeys": ["temporary"] }
  }
}
```

Untuk lebih banyak contoh kebijakan IAM yang menerapkan batasan pada tag yang ditentukan untuk grup Auto Scaling, lihat [Kontrol kunci tag dan nilai tag mana yang dapat digunakan](#)

Note

Bahkan jika Anda memiliki kebijakan yang membatasi pengguna Anda untuk melakukan operasi penandaan (atau untagging) pada grup Auto Scaling, ini tidak mencegah mereka mengubah tag secara manual pada instance setelah diluncurkan. Untuk contoh yang mengontrol akses ke tag pada instans EC2, lihat [Contoh: Menandai sumber daya](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Gunakan tag untuk memfilter grup Auto Scaling

Contoh berikut menunjukkan cara menggunakan filter dengan [describe-auto-scaling-groups](#) perintah untuk menggambarkan grup Auto Scaling dengan tag tertentu. Pemfilteran berdasarkan tag terbatas pada AWS CLI atau SDK, dan tidak tersedia dari konsol.

Pertimbangan penyaringan

- Anda dapat menentukan banyak filter dan banyak nilai filter dalam satu permintaan.
- Anda tidak dapat menggunakan wildcard dengan nilai filter.
- Nilai filter peka huruf besar/kecil.

Contoh: Jelaskan grup Auto Scaling dengan kunci tag dan pasangan nilai tertentu

Perintah berikut menunjukkan cara memfilter hasil untuk hanya menampilkan grup Auto Scaling dengan kunci tag dan pasangan nilai. **environment=production**

```
aws autoscaling describe-auto-scaling-groups \
  --filters Name=tag-key,Values=environment Name=tag-value,Values=production
```

Berikut ini adalah contoh respons.

```
{
  "AutoScalingGroups": [
    {
      "AutoScalingGroupName": "my-asg",
      "AutoScalingGroupARN": "arn",
      "LaunchTemplate": {
        "LaunchTemplateId": "lt-0b97f1e282EXAMPLE",
        "LaunchTemplateName": "my-launch-template",
        "Version": "$Latest"
      },
      "MinSize": 1,
      "MaxSize": 5,
      "DesiredCapacity": 1,
      "...",
      "Tags": [
        {
          "ResourceType": "auto-scaling-group",
          "ResourceId": "my-asg",
          "PropagateAtLaunch": true,
          "Value": "production",
          "Key": "environment"
        }
      ],
      "...",
    },
    ... additional groups ...
  ]
}
```

Atau, Anda dapat menentukan tag menggunakan tag **<key>** filter. Misalnya, perintah berikut menunjukkan cara memfilter hasil untuk hanya menampilkan grup Auto Scaling dengan kunci tag dan pasangan nilai **environment=production**. Filter ini diformat sebagai berikut: **Name=tag:<key>,Values=<value>**, dengan **<key>** dan **<value>** mewakili kunci tag dan pasangan nilai.

```
aws autoscaling describe-auto-scaling-groups \  
  --filters Name=tag:environment,Values=production
```

Anda juga dapat memfilter AWS CLI output dengan menggunakan `--query` opsi. Contoh berikut menunjukkan cara membatasi AWS CLI output untuk perintah sebelumnya ke nama grup, ukuran minimum, ukuran maksimum, dan atribut kapasitas yang diinginkan saja.

```
aws autoscaling describe-auto-scaling-groups \  
  --filters Name=tag:environment,Values=production \  
  --query "AutoScalingGroups[].{AutoScalingGroupName: AutoScalingGroupName, MinSize: MinSize, MaxSize: MaxSize, DesiredCapacity: DesiredCapacity}"
```

Berikut ini adalah contoh respons.

```
[  
  {  
    "AutoScalingGroupName": "my-asg",  
    "MinSize": 0,  
    "MaxSize": 10,  
    "DesiredCapacity": 1  
  },  
  ... additional groups ...  
]
```

Untuk informasi selengkapnya tentang pemfilteran, lihat [Memfilter AWS CLI output](#) di AWS Command Line Interface Panduan Pengguna.

Contoh: Jelaskan grup Auto Scaling dengan tag yang cocok dengan kunci tag yang ditentukan

Perintah berikut menunjukkan cara memfilter hasil agar hanya menampilkan grup Auto Scaling dengan **environment** tag, terlepas dari nilai tag.

```
aws autoscaling describe-auto-scaling-groups \  
  --filters Name=tag-key,Values=environment
```

Contoh: Jelaskan grup Auto Scaling dengan tag yang cocok dengan kumpulan kunci tag yang ditentukan

Perintah berikut menunjukkan cara memfilter hasil agar hanya menampilkan grup Auto Scaling dengan tag untuk **environment** dan **project**, terlepas dari nilai tag.

```
aws autoscaling describe-auto-scaling-groups \  
  --filters Name=tag-key,Values=environment Name=tag-key,Values=project
```

Contoh: Jelaskan grup Auto Scaling dengan tag yang cocok dengan setidaknya salah satu kunci tag yang ditentukan

Perintah berikut menunjukkan cara memfilter hasil agar hanya menampilkan grup Auto Scaling dengan tag untuk **environment** atau **project**, terlepas dari nilai tag.

```
aws autoscaling describe-auto-scaling-groups \  
  --filters Name=tag-key,Values=environment,project
```

Contoh: Jelaskan grup Auto Scaling dengan nilai tag yang ditentukan

Perintah berikut menunjukkan cara memfilter hasil agar hanya menampilkan grup Auto Scaling dengan nilai tag **production**, terlepas dari kunci tag.

```
aws autoscaling describe-auto-scaling-groups \  
  --filters Name=tag-value,Values=production
```

Contoh: Jelaskan grup Auto Scaling dengan kumpulan nilai tag yang ditentukan

Perintah berikut menunjukkan cara memfilter hasil agar hanya menampilkan grup Auto Scaling dengan nilai tag **production** dan **development**, terlepas dari kunci tag.

```
aws autoscaling describe-auto-scaling-groups \  
  --filters Name=tag-value,Values=production Name=tag-value,Values=development
```

Contoh: Jelaskan grup Auto Scaling dengan tag yang cocok dengan setidaknya salah satu nilai tag yang ditentukan

Perintah berikut menunjukkan cara memfilter hasil agar hanya menampilkan grup Auto Scaling dengan nilai tag **production** atau **development**, terlepas dari kunci tag.

```
aws autoscaling describe-auto-scaling-groups \  
  --filters Name=tag-value,Values=production,development
```

```
--filters Name=tag-value,Values=production,development
```

Contoh: Jelaskan grup Auto Scaling dengan tag yang cocok dengan beberapa kunci dan nilai tag

Anda juga dapat menggabungkan filter untuk membuat kustom AND dan OR logika untuk melakukan pemfilteran yang lebih kompleks.

Perintah berikut menunjukkan cara memfilter hasil agar hanya menampilkan grup Auto Scaling dengan kumpulan tag tertentu. Salah satu kunci tag adalah **environment** AND nilai tag adalah (**productionORdevelopment**) AND kunci tag lainnya adalah **costcenter** AND nilai tag **cc123**.

```
aws autoscaling describe-auto-scaling-groups \  
  --filters Name=tag:environment,Values=production,development \  
  Name=tag:costcenter,Values=cc123
```

Kebijakan pemeliharaan instans

Anda dapat mengonfigurasi kebijakan pemeliharaan instans untuk grup Auto Scaling agar memenuhi persyaratan kapasitas tertentu selama peristiwa yang menyebabkan instance diganti, seperti penyegaran instans atau proses pemeriksaan kesehatan.

Misalnya, Anda memiliki grup Auto Scaling yang memiliki sejumlah kecil instance. Anda ingin menghindari potensi gangguan dari penghentian dan kemudian mengganti instance ketika pemeriksaan kesehatan menunjukkan contoh yang terganggu. Dengan kebijakan pemeliharaan instans, Anda dapat memastikan bahwa Auto Scaling Amazon EC2 meluncurkan instans baru terlebih dahulu dan kemudian menunggu instans siap sepenuhnya sebelum menghentikan instans yang tidak sehat.

Kebijakan pemeliharaan instans juga membantu Anda meminimalkan potensi gangguan dalam kasus di mana beberapa instance diganti secara bersamaan. Anda menetapkan parameter persentase sehat minimum dan maksimum untuk kebijakan tersebut, dan grup Auto Scaling Anda hanya dapat menambah dan mengurangi kapasitas dalam rentang maksimum minimum tersebut saat mengganti instans. Rentang yang lebih besar meningkatkan jumlah instance yang dapat diganti pada saat yang bersamaan.

Daftar Isi

- [Ikhtisar kebijakan pemeliharaan instans](#)
- [Menetapkan kebijakan pemeliharaan instans pada grup Auto Scaling](#)

Ikhtisar kebijakan pemeliharaan instans

Topik ini memberikan ikhtisar opsi yang tersedia dan menjelaskan apa yang harus dipertimbangkan saat Anda membuat kebijakan pemeliharaan instans.

Daftar Isi

- [Gambaran Umum](#)
- [Konsep inti](#)
- [Pemanasan contoh](#)
- [Masa tenggang pemeriksaan kesehatan](#)
- [Skalakan grup Auto Scaling Anda](#)
- [Contoh alur perencanaan](#)

Gambaran Umum

Saat Anda membuat kebijakan pemeliharaan instans untuk grup Auto Scaling, kebijakan tersebut memengaruhi peristiwa Auto Scaling Amazon EC2 yang menyebabkan instance diganti. Ini menghasilkan perilaku penggantian yang lebih konsisten dalam grup Auto Scaling yang sama. Ini juga memungkinkan Anda mengoptimalkan grup Anda untuk ketersediaan atau biaya tergantung pada kebutuhan Anda.

Di konsol, opsi konfigurasi berikut tersedia:

- Peluncuran sebelum mengakhiri - Sebuah instance baru harus disediakan terlebih dahulu sebelum instance yang ada dapat dihentikan. Pendekatan ini adalah pilihan yang baik untuk aplikasi yang mendukung ketersediaan daripada penghematan biaya.
- Hentikan dan luncurkan - Instans baru disediakan pada saat yang sama instans Anda yang ada dihentikan. Pendekatan ini adalah pilihan yang baik untuk aplikasi yang mendukung penghematan biaya daripada ketersediaan. Ini juga merupakan pilihan yang baik untuk aplikasi yang seharusnya tidak meluncurkan kapasitas lebih dari yang tersedia saat ini, bahkan ketika mengganti instance.
- Kebijakan khusus — Opsi ini memungkinkan Anda menyiapkan kebijakan dengan rentang minimum dan maksimum khusus untuk jumlah kapasitas yang Anda inginkan saat mengganti instans. Pendekatan ini dapat membantu Anda mencapai keseimbangan yang tepat antara biaya dan ketersediaan.

Default untuk grup Auto Scaling adalah tidak memiliki kebijakan pemeliharaan instans, yang menyebabkannya merespons peristiwa pemeliharaan instance dengan perilaku default. Perilaku default dijelaskan dalam tabel berikut.

Perilaku default acara pemeliharaan instans

Peristiwa	Deskripsi	Perilaku default
Kegagalan pemeriksaan kesehatan	Terjadi secara otomatis ketika instance gagal dalam pemeriksaan kesehatan mereka. Auto Scaling Amazon EC2 menggantikan instans yang gagal dalam pemeriksaan kesehatannya. Untuk memahami penyebab kegagalan pemeriksaan kesehatan, lihat Health memeriksa instans dalam grup Auto Scaling .	Hentikan dan luncurkan.
Refresh instans	Terjadi saat Anda memulai penyegaran instance. Bergantung pada konfigurasi Anda, penyegaran instance menggantikan instance satu per satu, beberapa sekaligus, atau sekaligus. Untuk informasi selengkapnya, lihat Menggunakan penyegaran instans untuk memperbaiki instance dalam grup Auto Scaling .	Hentikan dan luncurkan.
Masa aktif instans maksimum	Terjadi secara otomatis ketika instance mencapai masa pakai instans maksimum yang Anda tentukan untuk	Hentikan dan luncurkan.

Peristiwa	Deskripsi	Perilaku default
	<p>grup Auto Scaling Anda. Auto Scaling Amazon EC2 menggantikan instans yang mencapai masa pakai instans maksimumnya. Untuk informasi selengkapnya, lihat Ganti instans Auto Scaling berdasarkan masa pakai instans maksimum.</p>	

Peristiwa	Deskripsi	Perilaku default
Menyeimbangkan kembali	<p>Terjadi secara otomatis jika ada perubahan mendasar yang menyebabkan grup menjadi tidak seimbang. Auto Scaling Amazon EC2 menyeimbangkan kembali grup dalam situasi berikut:</p> <ul style="list-style-type: none">• Availability Zone yang sebelumnya memiliki kapasitas tidak mencukupi pulih, atau Anda menambahkan atau menghapus Availability Zone dari grup. Ketika ini terjadi, grup Auto Scaling Anda mencoba menyeimbangkan dirinya secara merata di seluruh Availability Zone. Untuk informasi selengkapnya, lihat Aktivitas rebalancing.• Anda mengaktifkan Rebalancing Kapasitas pada grup Auto Scaling, dan mencoba meluncurkan Instans Spot baru sebelum yang sudah ada terputus karena ketersediaan Instans Spot berubah. Untuk informasi selengkapnya, lihat Gunakan Rebalancing Kapasitas untuk menangani interupsi Spot Amazon EC2.	<p>Luncurkan sebelum mengakhiri.</p> <p>Auto Scaling Amazon EC2 dapat melebihi batas ukuran grup Anda hingga 10 persen dari kapasitas maksimumnya. Namun, jika Anda menggunakan Capacity Rebalancing, itu hanya dapat melebihi batas ini hingga 10 persen dari kapasitas yang diinginkan.</p>

Peristiwa	Deskripsi	Perilaku default
	<ul style="list-style-type: none"> Anda memperbarui grup Auto Scaling, dan secara bertahap menggantikan instance agar sesuai dengan opsi pembelian baru yang Anda pilih saat memperbarui kebijakan instans campuran. Untuk informasi selengkapnya, lihat Memperbarui grup Auto Scaling. 	

Auto Scaling Amazon EC2 akan terus default untuk dihentikan dan diluncurkan dalam situasi berikut. Oleh karena itu, ketika salah satu situasi ini terjadi, kapasitas grup Anda mungkin kurang dari ambang batas bawah kebijakan pemeliharaan instans Anda.

- Ketika sebuah instance berakhir secara tak terduga, misalnya, karena tindakan manusia. Auto Scaling Amazon EC2 segera menggantikan instans yang tidak lagi berjalan. Untuk informasi selengkapnya, lihat [Pemeriksaan kesehatan Amazon EC2](#).
- Saat Amazon EC2 melakukan reboot, menghentikan, atau menghentikan instans sebagai bagian dari acara terjadwal sebelum Auto Scaling Amazon EC2 dapat meluncurkan instans pengganti. Untuk informasi selengkapnya tentang peristiwa ini, lihat [Acara terjadwal untuk instans Anda](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Ketika Layanan Spot Amazon EC2 memulai interupsi Instans Spot dan Instans Spot kemudian dihentikan secara paksa.

Dengan Instans Spot, jika Anda mengaktifkan Rebalancing Kapasitas pada grup Auto Scaling, maka instance tersebut mungkin sudah memiliki instance yang tertunda dari kumpulan Spot lain yang kami luncurkan sebelum kami memulai interupsi Spot. Untuk detail tentang cara kerja Penyeimbangan Kembali Kapasitas, lihat. [Gunakan Rebalancing Kapasitas untuk menangani interupsi Spot Amazon EC2](#)

Namun, karena Instans Spot tidak dijamin akan tetap tersedia dan dapat dihentikan dengan pemberitahuan interupsi Instans Spot selama dua menit, ambang batas bawah kebijakan pemeliharaan instans Anda dapat dilampaui jika instans terputus sebelum instans baru diluncurkan.

Konsep inti

Sebelum memulai, biasakan diri Anda dengan konsep dan istilah inti berikut:

Kapasitas yang diinginkan

Kapasitas yang diinginkan adalah kapasitas grup Auto Scaling pada saat pembuatan. Ini juga merupakan kapasitas yang coba dipertahankan kelompok ketika tidak ada kondisi penskalaan yang melekat pada grup.

Kebijakan pemeliharaan instans

Kebijakan pemeliharaan instans mengontrol apakah sebuah instance disediakan terlebih dahulu sebelum instance yang ada dihentikan untuk kejadian pemeliharaan instance. Ini juga menentukan seberapa jauh di bawah dan di atas kapasitas yang Anda inginkan grup Auto Scaling Anda mungkin pergi untuk mengganti beberapa instance secara bersamaan.

Persentase sehat maksimum

Persentase sehat maksimum adalah persentase kapasitas yang diinginkan yang dapat ditingkatkan oleh grup Auto Scaling Anda saat mengganti instance. Ini mewakili persentase maksimum kelompok yang dapat melayani dan sehat, atau tertunda, untuk mendukung beban kerja Anda. Di konsol, Anda dapat mengatur persentase sehat maksimum saat menggunakan opsi Launch before terminating atau opsi Kebijakan kustom. Nilai yang valid adalah 100-200 persen.

Persentase sehat minimum

Persentase sehat minimum adalah persentase kapasitas yang diinginkan untuk tetap dalam pelayanan, sehat, dan siap digunakan untuk mendukung beban kerja Anda saat mengganti instance. Sebuah instance dianggap sehat dan siap digunakan setelah berhasil menyelesaikan pemeriksaan kesehatan pertamanya dan waktu pemanasan yang ditentukan berlalu. Di konsol, Anda dapat menyetel persentase sehat minimum saat menggunakan opsi Hentikan dan peluncuran atau opsi Kebijakan khusus. Nilai yang valid adalah 0-100 persen.

Note

Untuk mengganti instance lebih cepat, Anda dapat menentukan persentase sehat minimum yang rendah. Namun, jika tidak ada cukup contoh sehat yang berjalan, itu dapat mengurangi ketersediaan. Sebaiknya pilih nilai yang wajar untuk menjaga ketersediaan dalam situasi di mana beberapa instance akan diganti.

Pemanasan contoh

Jika instans Anda memerlukan waktu untuk diinisialisasi setelah masuk ke InService status, aktifkan pemanasan instans default untuk grup Auto Scaling Anda. Dengan pemanasan instans default, Anda dapat mencegah instance dihitung ke persentase sehat minimum sebelum siap. Ini memastikan bahwa Auto Scaling Amazon EC2 mempertimbangkan berapa lama waktu yang dibutuhkan untuk memiliki kapasitas yang cukup untuk mendukung beban kerja sebelum menghentikan instans yang ada.

Sebagai manfaat tambahan, Anda dapat meningkatkan CloudWatch metrik Amazon yang digunakan untuk penskalaan dinamis saat mengaktifkan pemanasan instans default. Jika grup Auto Scaling Anda memiliki kebijakan penskalaan apa pun, saat grup diskalakan keluar, grup tersebut menggunakan periode pemanasan default yang sama untuk mencegah instance dihitung terhadap CloudWatch metrik sebelum selesai diinisialisasi.

Untuk informasi selengkapnya, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#).

Masa tenggang pemeriksaan kesehatan

Auto Scaling Amazon EC2 menentukan apakah instans sehat berdasarkan status pemeriksaan kesehatan yang digunakan grup Auto Scaling Anda. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

Untuk memastikan bahwa pemeriksaan kesehatan ini dimulai sesegera mungkin, jangan atur tenggang pemeriksaan kesehatan kelompok terlalu tinggi, tetapi cukup tinggi untuk pemeriksaan kesehatan Elastic Load Balancing Anda untuk menentukan apakah target tersedia untuk menangani permintaan. Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).

Skalakan grup Auto Scaling Anda

Kebijakan pemeliharaan instans hanya berlaku untuk peristiwa pemeliharaan instans dan tidak mencegah grup diskalakan secara manual atau otomatis.

Jika ada kebijakan penskalaan atau tindakan terjadwal yang dilampirkan ke grup Auto Scaling Anda, kebijakan tersebut dapat berjalan secara paralel saat peristiwa pemeliharaan instance terjadi. Dalam hal ini, mereka dapat meningkatkan atau mengurangi kapasitas yang diinginkan grup tetapi hanya dalam batas penskalaan yang Anda tentukan. Untuk informasi lebih lanjut tentang batasan ini, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).

Contoh alur perencanaan

Dalam skenario tipikal, kebijakan pemeliharaan instans Anda dan kapasitas yang diinginkan mungkin terlihat seperti ini:

- Persentase sehat minimum = 90 persen
- Persentase sehat maksimum = 120 persen
- Kapasitas yang diinginkan = 100

Selama acara pemeliharaan instans apa pun, grup Auto Scaling Anda mungkin memiliki sedikitnya 90 instans dan sebanyak 120. Setelah acara, grup kembali memiliki 100 instance.

Bila Anda menggunakan kebijakan pemeliharaan instans dengan grup Auto Scaling yang memiliki kolam hangat, persentase sehat minimum dan maksimum diterapkan secara terpisah ke grup Auto Scaling dan kolam hangat.

Misalnya, anggap ini adalah konfigurasi Anda:

- Persentase sehat minimum = 90 persen
- Persentase sehat maksimum = 120 persen
- Kapasitas yang diinginkan = 100
- Ukuran kolam hangat = 10

Jika Anda memulai penyegaran instans untuk mendaur ulang instans grup, Auto Scaling Amazon EC2 menggantikan instans di grup Auto Scaling terlebih dahulu, lalu instance di kolam hangat. Meskipun Auto Scaling Amazon EC2 masih bekerja untuk mengganti instans di grup Auto Scaling, grup ini mungkin memiliki sedikitnya 90 instans dan sebanyak 120. Setelah selesai dengan grup,

Amazon EC2 Auto Scaling dapat bekerja mengganti instance di kolam hangat. Sementara ini terjadi, kolam hangat mungkin memiliki sedikitnya 9 kali dan sebanyak 12.

Menetapkan kebijakan pemeliharaan instans pada grup Auto Scaling

Anda dapat membuat kebijakan pemeliharaan instans saat membuat grup Auto Scaling. Anda juga dapat membuatnya untuk grup yang ada.

Dengan menetapkan kebijakan pemeliharaan instans pada grup Auto Scaling, Anda tidak perlu lagi menentukan nilai untuk parameter persentase sehat minimum dan maksimum untuk fitur penyegaran instans kecuali Anda ingin mengganti kebijakan pemeliharaan instans.

Di konsol, Amazon EC2 Auto Scaling menyediakan opsi untuk membantu Anda memulai.

Daftar Isi

- [Tetapkan kebijakan pemeliharaan instans](#)
- [Hapus kebijakan pemeliharaan instans](#)

Tetapkan kebijakan pemeliharaan instans

Untuk menetapkan kebijakan pemeliharaan instans pada grup Auto Scaling, gunakan salah satu metode berikut:

Console

Untuk menyetel kebijakan pemeliharaan instans pada grup baru (konsol)

1. Ikuti instruksi [Buat grup Auto Scaling menggunakan templat peluncuran](#) dan selesaikan setiap langkah dalam prosedur, hingga langkah 11.
2. Pada kebijakan ukuran grup dan penskalaan Konfigurasi, untuk kapasitas yang diinginkan, masukkan jumlah awal instance yang akan diluncurkan.
3. Di bagian Penskalaan, di bawah batas Penskalaan, jika nilai baru Anda untuk kapasitas yang diinginkan lebih besar dari kapasitas yang diinginkan Min dan kapasitas yang diinginkan Maks, kapasitas Maks yang diinginkan secara otomatis ditingkatkan ke nilai kapasitas baru yang diinginkan. Anda dapat mengubah batasan ini sesuai kebutuhan.
4. Untuk Penskalaan otomatis, pilih apakah Anda ingin membuat kebijakan penskalaan pelacakan target. Anda juga dapat membuat kebijakan ini setelah membuat grup Auto Scaling.

Jika Anda memilih kebijakan penskalaan pelacakan target, ikuti petunjuk [Buat kebijakan penskalaan pelacakan target](#) untuk membuat kebijakan.

5. Di bagian Kebijakan pemeliharaan instans, pilih salah satu opsi yang tersedia:
 - Luncurkan sebelum penghentian: Instance baru harus disediakan terlebih dahulu sebelum instance yang ada dapat dihentikan. Ini adalah pilihan yang baik untuk aplikasi yang mendukung ketersediaan daripada penghematan biaya.
 - Hentikan dan luncurkan: Instans baru disediakan pada saat yang sama instans Anda yang ada dihentikan. Ini adalah pilihan yang baik untuk aplikasi yang mendukung penghematan biaya daripada ketersediaan. Ini juga merupakan pilihan yang baik untuk aplikasi yang seharusnya tidak meluncurkan kapasitas lebih dari yang tersedia saat ini.
 - Kebijakan khusus: Opsi ini memungkinkan Anda menyiapkan kebijakan dengan rentang minimum dan maksimum khusus untuk jumlah kapasitas yang Anda inginkan saat mengganti instans. Ini dapat membantu Anda mencapai keseimbangan yang tepat antara biaya dan ketersediaan.
6. Untuk Tetapkan persentase sehat, masukkan nilai untuk satu atau kedua bidang berikut. Bidang yang diaktifkan bervariasi tergantung pada opsi yang Anda pilih pada langkah sebelumnya.
 - Min: Menetapkan persentase sehat minimum yang diperlukan untuk melanjutkan penggantian instance.
 - Maks: Menetapkan persentase sehat maksimum yang mungkin saat mengganti instance.
7. Perluas kapasitas Tampilan selama penggantian berdasarkan bagian kapasitas yang Anda inginkan untuk mengonfirmasi bagaimana nilai untuk Min dan Maks berlaku untuk grup Anda. Nilai pasti yang digunakan tergantung pada nilai kapasitas yang diinginkan, yang akan berubah jika skala grup.
8. Lanjutkan dengan langkah-langkah masuk [Buat grup Auto Scaling menggunakan templat peluncuran](#).

AWS CLI

Untuk menetapkan kebijakan pemeliharaan instans pada grup baru (AWS CLI)

Tambahkan `--instance-maintenance-policy` opsi ke [create-auto-scaling-group](#) perintah. Contoh berikut menetapkan kebijakan pemeliharaan instance pada grup Auto Scaling baru bernama `my-asg`

```
aws autoscaling create-auto-scaling-group \  
  --launch-template LaunchTemplateName=my-launch-template,Version='1' \  
  --auto-scaling-group-name my-asg \  
  --min-size 1 \  
  --max-size 10 \  
  --desired-capacity 5 \  
  --default-instance-warmup 20 \  
  --instance-maintenance-policy '{  
    "MinHealthyPercentage": 90,  
    "MaxHealthyPercentage": 120  
  }' \  
  --vpc-zone-identifier "subnet-5e6example,subnet-613example,subnet-c93example"
```

Console

Untuk menyetel kebijakan pemeliharaan instans pada grup (konsol) yang ada

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pada bilah navigasi di bagian atas layar, pilih tempat Wilayah AWS Anda membuat grup Auto Scaling.
3. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

4. Pada tab Detail, pilih Kebijakan pemeliharaan instans, Edit.
5. Untuk menetapkan kebijakan pemeliharaan instans pada grup, pilih salah satu opsi yang tersedia:
 - Luncurkan sebelum penghentian: Instance baru harus disediakan terlebih dahulu sebelum instance yang ada dapat dihentikan. Ini adalah pilihan yang baik untuk aplikasi yang mendukung ketersediaan daripada penghematan biaya.
 - Hentikan dan luncurkan: Instans baru disediakan pada saat yang sama instans Anda yang ada dihentikan. Ini adalah pilihan yang baik untuk aplikasi yang mendukung penghematan biaya daripada ketersediaan. Ini juga merupakan pilihan yang baik untuk aplikasi yang seharusnya tidak meluncurkan kapasitas lebih dari yang tersedia saat ini.
 - Kebijakan khusus: Opsi ini memungkinkan Anda menyiapkan kebijakan dengan rentang minimum dan maksimum khusus untuk jumlah kapasitas yang Anda inginkan saat

mengganti instans. Ini dapat membantu Anda mencapai keseimbangan yang tepat antara biaya dan ketersediaan.

6. Untuk Tetapkan persentase sehat, masukkan nilai untuk satu atau kedua bidang berikut. Bidang yang diaktifkan bervariasi tergantung pada opsi yang Anda pilih pada langkah sebelumnya.
 - Min: Menetapkan persentase sehat minimum yang diperlukan untuk melanjutkan penggantian instance.
 - Maks: Menetapkan persentase sehat maksimum yang mungkin saat mengganti instance.
7. Perluas kapasitas Tampilan selama penggantian berdasarkan bagian kapasitas yang Anda inginkan untuk mengonfirmasi bagaimana nilai untuk Min dan Maks berlaku untuk grup Anda. Nilai pasti yang digunakan tergantung pada nilai kapasitas yang diinginkan, yang akan berubah jika skala grup.
8. Pilih Perbarui.

AWS CLI

Untuk menetapkan kebijakan pemeliharaan instans pada grup (AWS CLI) yang ada

Tambahkan `--instance-maintenance-policy` opsi ke [update-auto-scaling-group](#) perintah. Contoh berikut menetapkan kebijakan pemeliharaan instance pada grup Auto Scaling yang ditentukan.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
--instance-maintenance-policy '{  
  "MinHealthyPercentage": 90,  
  "MaxHealthyPercentage": 120  
}'
```

Hapus kebijakan pemeliharaan instans

Jika Anda ingin berhenti menggunakan kebijakan pemeliharaan instans dengan grup Auto Scaling, Anda dapat menghapusnya.

Console

Untuk menghapus kebijakan pemeliharaan instance (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pada bilah navigasi di bagian atas layar, pilih tempat Wilayah AWS Anda membuat grup Auto Scaling.
3. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

4. Pada tab Detail, pilih Kebijakan pemeliharaan instans, Edit.
5. Pilih Tidak ada kebijakan pemeliharaan instans.
6. Pilih Perbarui.

AWS CLI

Untuk menghapus kebijakan pemeliharaan instance (AWS CLI)

Tambahkan `--instance-maintenance-policy` opsi ke [update-auto-scaling-group](#) perintah. Contoh berikut menghapus kebijakan pemeliharaan instance dari grup Auto Scaling yang ditentukan.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
  --instance-maintenance-policy '{  
    "MinHealthyPercentage": -1,  
    "MaxHealthyPercentage": -1  
  }'
```

Hook siklus aktif Amazon EC2 Auto Scaling

Auto Scaling Amazon EC2 menawarkan kemampuan untuk menambahkan kait siklus hidup ke grup Auto Scaling Anda. Hook ini memungkinkan Anda membuat solusi yang mengetahui peristiwa dalam siklus hidup instans Auto Scaling, dan kemudian melakukan tindakan kustom pada instance saat peristiwa siklus hidup terkait terjadi. Pengait siklus hidup menyediakan jumlah waktu tertentu (satu jam secara default) untuk menunggu tindakan selesai sebelum instance bertransisi ke status berikutnya.

Sebagai contoh penggunaan kait siklus hidup dengan instance Auto Scaling:

- Ketika peristiwa scale-out terjadi, instance Anda yang baru diluncurkan menyelesaikan urutan startup dan transisi ke status tunggu. Sementara instance dalam keadaan menunggu, ia menjalankan skrip untuk mengunduh dan menginstal paket perangkat lunak yang diperlukan untuk aplikasi Anda, memastikan bahwa instance Anda sepenuhnya siap sebelum mulai menerima lalu lintas. Ketika skrip selesai menginstal perangkat lunak, ia mengirimkan `complete-lifecycle-action` perintah untuk melanjutkan.
- Ketika peristiwa scale-in terjadi, kait siklus hidup menjeda instance sebelum dihentikan dan mengirimkan pemberitahuan kepada Anda menggunakan Amazon EventBridge. Saat instance dalam status tunggu, Anda dapat memanggil AWS Lambda fungsi atau terhubung ke instance untuk mengunduh log atau data lain sebelum instance dihentikan sepenuhnya.

Penggunaan kait siklus hidup yang populer adalah untuk mengontrol saat instance terdaftar dengan Elastic Load Balancing. Dengan menambahkan hook siklus hidup peluncuran ke grup Auto Scaling Anda, Anda dapat memastikan bahwa skrip bootstrap Anda telah berhasil diselesaikan dan aplikasi pada instance siap menerima lalu lintas sebelum terdaftar ke penyeimbang beban di akhir kait siklus hidup.

Daftar Isi

- [Ketersediaan kait siklus hidup](#)
- [Pertimbangan dan batasan untuk kait siklus hidup](#)
- [Sumber daya terkait](#)
- [Cara kerja hook siklus aktif](#)
- [Bersiaplah untuk menambahkan kait siklus hidup ke grup Auto Scaling Anda](#)
- [Ambil status siklus hidup target melalui metadata instance](#)
- [Tambahkan hook siklus aktif](#)
- [Selesaikan tindakan siklus hidup](#)
- [Tutorial: Konfigurasi data pengguna untuk mengambil status siklus hidup target melalui metadata instance](#)
- [Tutorial: Konfigurasi kait siklus hidup yang memanggil fungsi Lambda](#)

Ketersediaan kait siklus hidup

Tabel berikut mencantumkan kait siklus hidup yang tersedia untuk berbagai skenario.

Peristiwa	Peluncuran atau penghentian instance ¹	Masa Pakai Instance Maksimum : Instans penggantian	Penyegaran Instance : Instans penggantian	Penyeimbangan Kembali Kapasitas : Contoh penggantian	Kolam Hangat : Contoh masuk dan meninggalkan kolam hangat
Peluncuran instance	✓	✓	✓	✓	✓
Pengakhiran instance	✓	✓	✓	✓	✓

¹ Berlaku untuk semua peluncuran dan penghentian, apakah mereka dimulai secara otomatis atau manual seperti ketika Anda memanggil atau operasi. `SetDesiredCapacity` `TerminateInstanceInAutoScalingGroup` Tidak berlaku saat Anda melampirkan atau melepaskan instance, memindahkan instance masuk dan keluar dari mode siaga, atau menghapus grup dengan opsi hapus paksa.

Pertimbangan dan batasan untuk kait siklus hidup

Saat bekerja dengan kait siklus hidup, ingatlah catatan dan batasan berikut:

- Auto Scaling Amazon EC2 menyediakan siklus hidupnya sendiri untuk membantu pengelolaan grup Auto Scaling. Siklus hidup ini berbeda dari instans EC2 lainnya. Untuk informasi selengkapnya, lihat [Siklus aktif instance Amazon EC2 Auto Scaling](#). Contoh di kolam hangat juga memiliki siklus hidup mereka sendiri, seperti yang dijelaskan dalam [Transisi status siklus hidup untuk instance di kolam hangat](#)
- Anda dapat menggunakan kait siklus hidup dengan Instans Spot, tetapi pengait siklus hidup tidak mencegah instance berhenti jika kapasitas tidak lagi tersedia, yang dapat terjadi kapan saja dengan pemberitahuan interupsi dua menit. Untuk informasi selengkapnya, lihat [Interupsi Instans Spot](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux. Namun, Anda dapat mengaktifkan

Penyeimbangan Kembali Kapasitas untuk secara proaktif mengganti Instans Spot yang telah menerima rekomendasi penyeimbangan kembali dari layanan Spot Amazon EC2, sinyal yang dikirim saat Instans Spot berisiko tinggi mengalami gangguan. Untuk informasi selengkapnya, lihat [Gunakan Rebalancing Kapasitas untuk menangani interupsi Spot Amazon EC2](#).

- Instance dapat tetap berada dalam status tunggu selama periode waktu tertentu. Batas waktu default untuk pengait siklus hidup adalah satu jam (batas waktu detak jantung). Ada juga batas waktu global yang menentukan jumlah waktu maksimum yang dapat Anda simpan instance dalam status tunggu. Batas waktu global adalah 48 jam atau 100 kali batas waktu detak jantung, mana yang lebih kecil.
- Hasil kait siklus hidup dapat ditinggalkan atau dilanjutkan. Jika instans diluncurkan, lanjutkan menunjukkan bahwa tindakan Anda berhasil, dan Auto Scaling Amazon EC2 dapat menempatkan instans ke dalam layanan. Jika tidak, abaikan menunjukkan bahwa tindakan kustom Anda tidak berhasil, dan kami dapat menghentikan dan mengganti instance. Jika sebuah instance berakhir, keduanya mengabaikan dan melanjutkan mengizinkan instance untuk dihentikan. Namun, abaikan menghentikan tindakan yang tersisa, seperti kait siklus hidup lainnya, dan lanjutkan memungkinkan pengait siklus hidup lainnya untuk diselesaikan.
- Auto Scaling Amazon EC2 membatasi kecepatan peluncuran instans jika kait siklus hidup gagal secara konsisten, jadi pastikan untuk menguji dan memperbaiki kesalahan permanen apa pun dalam tindakan siklus hidup Anda.
- Membuat dan memperbarui kait siklus hidup menggunakan AWS CLI, AWS CloudFormation, atau SDK menyediakan opsi yang tidak tersedia saat membuat kait siklus hidup dari. AWS Management Console Misalnya, bidang untuk menentukan ARN topik SNS atau antrean SQS tidak muncul di konsol, karena Amazon EC2 Auto Scaling sudah mengirimkan peristiwa ke Amazon. EventBridge Peristiwa ini dapat difilter dan diarahkan ke AWS layanan seperti Lambda, Amazon SNS, dan Amazon SQS sesuai kebutuhan.
- Anda dapat menambahkan beberapa kait siklus hidup ke grup Auto Scaling saat Anda membuatnya, dengan memanggil [CreateAutoScalingGroup](#) API menggunakan, atau SDK AWS CLI. AWS CloudFormation Namun, setiap hook harus memiliki target notifikasi dan peran IAM yang sama, jika ditentukan. Untuk membuat kait siklus hidup dengan target notifikasi yang berbeda dan peran yang berbeda, buat kait siklus hidup satu per satu dalam panggilan terpisah ke API. [PutLifecycleHook](#)
- Jika Anda menambahkan kait siklus hidup untuk peluncuran misalnya, masa tenggang pemeriksaan kesehatan dimulai segera setelah instance mencapai status. InService Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).

Pertimbangan penskalaan

- Kebijakan penskalaan dinamis masuk dan keluar sebagai respons terhadap data CloudWatch metrik, seperti CPU dan I/O jaringan, yang digabungkan di beberapa instance. Saat penskalaan keluar, Auto Scaling Amazon EC2 tidak langsung menghitung instance baru ke metrik instans gabungan dari grup Auto Scaling. Itu menunggu sampai instance mencapai InService status dan pemanasan instance telah selesai. Untuk informasi selengkapnya, lihat [Pertimbangan kinerja penskalaan](#) di topik pemanasan instance default.
- Pada skala dalam, metrik instans agregat mungkin tidak langsung mencerminkan penghapusan instance penghentian. Instans penghentian berhenti menghitung metrik instans agregat grup sesaat setelah alur kerja penghentian Auto Scaling Amazon EC2 dimulai.
- Dalam kebanyakan kasus ketika kait siklus hidup dipanggil, aktivitas penskalaan karena kebijakan penskalaan sederhana dijeda hingga tindakan siklus hidup selesai dan periode cooldown telah kedaluwarsa. Menetapkan interval panjang untuk periode cooldown berarti akan memakan waktu lebih lama untuk melanjutkan penskalaan. Untuk informasi lebih lanjut, lihat [Kait siklus hidup dapat menyebabkan penundaan tambahan](#) di topik cooldown. Secara umum, sebaiknya jangan menggunakan kebijakan penskalaan sederhana jika Anda dapat menggunakan kebijakan penskalaan langkah atau penskalaan pelacakan target.

Sumber daya terkait

Untuk video pengantar, lihat [AWS re:Invent 2018: Manajemen Kapasitas Menjadi Mudah dengan Auto Scaling Amazon EC2 aktif](#). YouTube

Kami menyediakan beberapa cuplikan template JSON dan YAMAL yang dapat Anda gunakan untuk memahami cara mendeklarasikan kait siklus hidup di templat tumpukan Anda. AWS CloudFormation Untuk informasi selengkapnya, lihat [AWS::AutoScaling::LifecycleHook](#) referensi di Panduan AWS CloudFormation Pengguna.

Anda juga dapat mengunjungi [GitHubrepositori](#) kami untuk mengunduh contoh templat dan skrip data pengguna untuk kait siklus hidup.

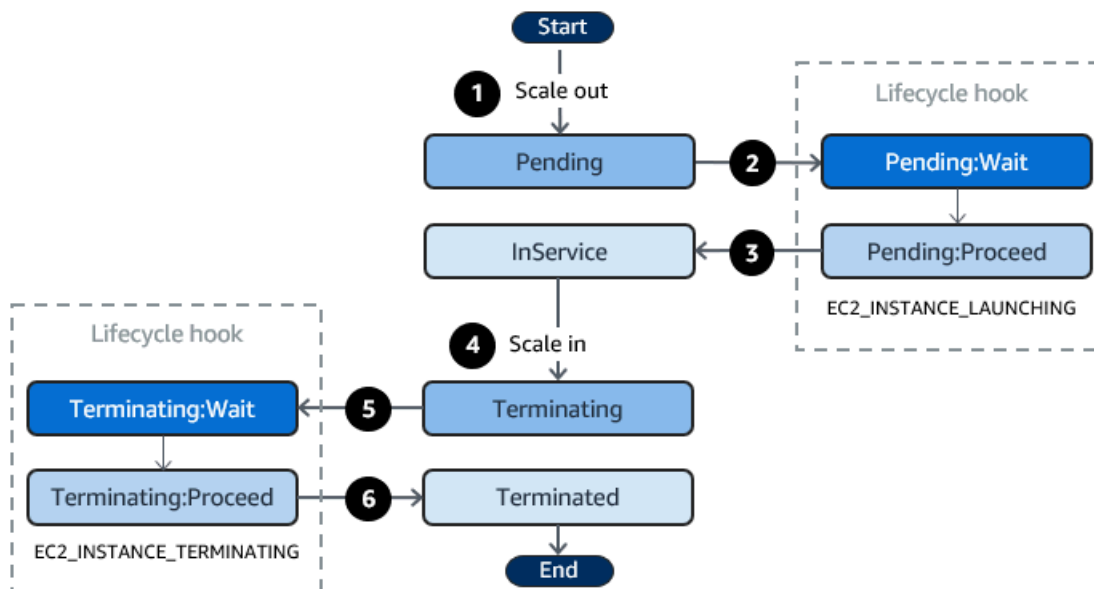
Untuk contoh penggunaan kait siklus hidup, lihat posting blog berikut.

- [Membangun Sistem Cadangan untuk Instans Berskala menggunakan Lambda dan Amazon EC2 Run Command](#)
- [Jalankan kode sebelum menghentikan instance EC2 Auto Scaling](#).

Cara kerja hook siklus aktif

Instans Amazon EC2 bertransisi melalui status yang berbeda dari saat diluncurkan hingga dihentikan. Anda dapat membuat tindakan kustom untuk grup Auto Scaling untuk bertindak saat instance bertransisi ke status tunggu karena pengait siklus hidup.

Ilustrasi berikut menunjukkan transisi antara status instans Auto Scaling saat Anda menggunakan kait siklus hidup untuk skala keluar dan menskalakan.



Seperti yang ditunjukkan pada diagram sebelumnya:

1. Grup Auto Scaling merespons peristiwa scale-out dan mulai meluncurkan instance.
2. Pengait siklus hidup menempatkan instance ke status tunggu (`Pending:Wait`) dan kemudian melakukan tindakan khusus.

Instance tetap dalam status tunggu hingga Anda menyelesaikan tindakan siklus hidup, atau periode batas waktu berakhir. Secara default, instance tetap dalam status tunggu selama satu jam, dan kemudian grup Auto Scaling melanjutkan proses peluncuran (`Pending:Proceed`). Jika Anda memerlukan waktu lebih banyak, Anda dapat memulai ulang periode jeda dengan merekam detak jantung. Jika Anda menyelesaikan tindakan siklus hidup ketika tindakan kustom telah selesai dan periode batas waktu belum kedaluwarsa, periode berakhir dan grup Auto Scaling melanjutkan proses peluncuran.

3. Instance memasuki `InService` negara bagian dan masa tenggang pemeriksaan kesehatan dimulai. Namun, sebelum instance mencapai `InService` status, jika grup Auto Scaling dikaitkan dengan penyeimbang beban Elastic Load Balancing, instans terdaftar dengan penyeimbang

beban, dan penyeimbang beban mulai memeriksa kesehatannya. Setelah masa tenggang pemeriksaan kesehatan berakhir, Auto Scaling Amazon EC2 mulai memeriksa kondisi kesehatan instans.

4. Grup Auto Scaling merespons peristiwa scale-in dan mulai menghentikan sebuah instance. Jika grup Auto Scaling digunakan dengan Elastic Load Balancing, instance terminating pertama kali didaftarkan dari load balancer. Jika pengurusan koneksi diaktifkan untuk penyeimbang beban, instance berhenti menerima koneksi baru dan menunggu koneksi yang ada mengalir sebelum menyelesaikan proses deregistrasi.
5. Pengait siklus hidup menempatkan instance ke status tunggu (`Terminating:Wait`) dan kemudian melakukan tindakan khusus.

Instance tetap dalam status tunggu baik sampai Anda menyelesaikan tindakan siklus hidup, atau sampai periode batas waktu berakhir (satu jam secara default). Setelah Anda menyelesaikan hook siklus hidup atau periode batas waktu kedaluwarsa, instance akan beralih ke status berikutnya (`Terminating:Proceed`).

6. Instance dihentikan.

Important

Instans di kolam hangat juga memiliki siklus hidupnya sendiri dengan status tunggu yang sesuai, seperti yang dijelaskan dalam [Transisi status siklus hidup untuk instance di kolam hangat](#)

Bersiaplah untuk menambahkan kait siklus hidup ke grup Auto Scaling Anda

Sebelum menambahkan hook siklus hidup ke grup Auto Scaling, pastikan skrip data pengguna atau target notifikasi sudah diatur dengan benar.

- Untuk menggunakan skrip data pengguna untuk melakukan tindakan kustom pada instance Anda saat diluncurkan, Anda tidak perlu mengonfigurasi target notifikasi. Namun, Anda harus sudah membuat template peluncuran atau konfigurasi peluncuran yang menentukan skrip data pengguna Anda dan menghubungkannya dengan grup Auto Scaling Anda. Untuk informasi selengkapnya tentang skrip data pengguna, lihat [Menjalankan perintah pada instans Linux Anda saat diluncurkan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

- Untuk memberi sinyal Auto Scaling Amazon EC2 saat tindakan siklus hidup selesai, Anda harus menambahkan [CompleteLifecycleAction](#) panggilan API ke skrip, dan Anda harus membuat peran IAM secara manual dengan kebijakan yang memungkinkan instance Auto Scaling memanggil API ini. Template peluncuran atau konfigurasi peluncuran Anda harus menentukan peran ini menggunakan profil instans IAM yang dilampirkan ke instans Amazon EC2 Anda saat peluncuran. Lihat informasi yang lebih lengkap di [Selesaikan tindakan siklus hidup](#) dan [Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2](#).
- Untuk menggunakan layanan seperti Lambda untuk melakukan tindakan kustom, Anda harus sudah membuat EventBridge aturan dan menetapkan fungsi Lambda sebagai targetnya. Untuk informasi selengkapnya, lihat [Konfigurasi target notifikasi untuk notifikasi siklus hidup](#).
- Untuk mengizinkan Lambda memberi sinyal Auto Scaling Amazon EC2 saat tindakan siklus hidup selesai, Anda harus [CompleteLifecycleAction](#) menambahkan panggilan API ke kode fungsi. Anda juga harus melampirkan kebijakan IAM ke peran eksekusi fungsi yang memberi Lambda izin untuk menyelesaikan tindakan siklus hidup. Untuk informasi selengkapnya, lihat [Tutorial: Konfigurasi kait siklus hidup yang memanggil fungsi Lambda](#).
- Untuk menggunakan layanan seperti Amazon SNS atau Amazon SQS untuk melakukan tindakan kustom, Anda harus sudah membuat topik SNS atau antrian SQS dan telah menyiapkan Nama Sumber Daya Amazon (ARN). Anda juga harus telah membuat peran IAM yang memberikan akses Auto Scaling Amazon EC2 ke topik SNS atau target SQS Anda dan telah menyiapkan ARN-nya. Untuk informasi selengkapnya, lihat [Konfigurasi target notifikasi untuk notifikasi siklus hidup](#).

Note

Secara default, saat Anda menambahkan kait siklus hidup di konsol, Auto Scaling Amazon EC2 mengirimkan pemberitahuan peristiwa siklus hidup ke Amazon EventBridge. Menggunakan EventBridge atau skrip data pengguna adalah praktik terbaik yang disarankan. Untuk membuat hook siklus hidup yang mengirimkan notifikasi langsung ke Amazon SNS atau Amazon SQS, gunakan AWS CloudFormation, atau SDK AWS CLI untuk menambahkan kait siklus hidup.

Konfigurasi target notifikasi untuk notifikasi siklus hidup

Anda dapat menambahkan kait siklus hidup ke grup Auto Scaling untuk melakukan tindakan kustom saat instance memasuki status tunggu. Anda dapat memilih layanan target untuk melakukan tindakan ini tergantung pada pendekatan pengembangan pilihan Anda.

Pendekatan pertama menggunakan Amazon EventBridge untuk menjalankan fungsi Lambda yang melakukan tindakan yang Anda inginkan. Pendekatan kedua melibatkan pembuatan topik Amazon Simple Notification Service (Amazon SNS) tempat notifikasi diterbitkan. Klien dapat berlangganan topik SNS dan menerima pesan yang dipublikasikan menggunakan protokol yang didukung. Pendekatan terakhir melibatkan penggunaan Amazon Simple Queue Service (Amazon Simple Queue Service), sistem pesan yang digunakan oleh aplikasi terdistribusi untuk bertukar pesan melalui model polling.

Sebagai praktik terbaik, kami sarankan Anda menggunakannya EventBridge. Pemberitahuan yang dikirim ke Amazon SNS dan Amazon SQS berisi informasi yang sama dengan notifikasi yang dikirimkan oleh Amazon EC2 Auto Scaling. EventBridge Sebelumnya EventBridge, praktik standar adalah mengirim pemberitahuan ke SNS atau SQS dan mengintegrasikan layanan lain dengan SNS atau SQS untuk melakukan tindakan terprogram. Hari ini, EventBridge memberi Anda lebih banyak opsi untuk layanan mana yang dapat Anda targetkan dan membuatnya lebih mudah untuk menangani acara menggunakan arsitektur tanpa server.

Prosedur berikut mencakup cara mengatur target notifikasi Anda.

Ingat, jika Anda memiliki skrip data pengguna di template peluncuran atau konfigurasi peluncuran yang mengonfigurasi instans Anda saat diluncurkan, Anda tidak perlu menerima pemberitahuan untuk melakukan tindakan khusus pada instance Anda.

Daftar Isi

- [Pemberitahuan rute ke Lambda menggunakan EventBridge](#)
- [Terima notifikasi menggunakan Amazon SNS](#)
- [Menerima pemberitahuan menggunakan Amazon SQS](#)
- [Contoh pesan pemberitahuan untuk Amazon SNS dan Amazon SQS](#)

Important

EventBridge Aturan, fungsi Lambda, topik Amazon SNS, dan antrean Amazon SQS yang Anda gunakan dengan kait siklus hidup harus selalu berada di Wilayah yang sama tempat Anda membuat grup Auto Scaling.

Pemberitahuan rute ke Lambda menggunakan EventBridge

Anda dapat mengonfigurasi EventBridge aturan untuk menjalankan fungsi Lambda saat instance memasuki status tunggu. Auto Scaling Amazon EC2 memancarkan pemberitahuan peristiwa siklus hidup tentang instans yang diluncurkan atau dihentikan dan token yang dapat Anda gunakan EventBridge untuk mengontrol tindakan siklus hidup. Untuk contoh peristiwa ini, lihat [Referensi acara Auto Scaling Amazon EC2](#).

Note

Saat Anda menggunakan aturan AWS Management Console untuk membuat acara, konsol secara otomatis menambahkan izin IAM yang diperlukan untuk memberikan EventBridge izin memanggil fungsi Lambda Anda. Jika Anda membuat aturan peristiwa menggunakan AWS CLI, Anda harus memberikan izin ini secara eksplisit.

Untuk informasi tentang cara membuat aturan peristiwa di EventBridge konsol, lihat [Membuat EventBridge aturan Amazon yang bereaksi terhadap peristiwa](#) di Panduan EventBridge Pengguna Amazon.

— atau —

Untuk tutorial pengantar yang ditujukan untuk pengguna konsol, lihat [Tutorial: Konfigurasi kait siklus hidup yang memanggil fungsi Lambda](#). Tutorial ini menunjukkan cara membuat fungsi Lambda sederhana yang mendengarkan acara peluncuran dan menuliskannya ke log Log. CloudWatch

Untuk membuat EventBridge aturan yang memanggil fungsi Lambda

1. Buat fungsi Lambda dengan menggunakan [konsol Lambda](#) dan catat Nama Sumber Daya Amazon (ARN). Misalnya, `arn:aws:lambda:region:123456789012:function:my-function`. Anda membutuhkan ARN untuk membuat target. EventBridge Untuk informasi selengkapnya, lihat [Memulai Lambda](#) di Panduan AWS Lambda Pengembang.
2. Untuk membuat aturan yang cocok dengan peristiwa misalnya peluncuran, gunakan perintah [put-rule](#) berikut.

```
aws events put-rule --name my-rule --event-pattern file://pattern.json --state ENABLED
```

Contoh berikut menunjukkan `pattern.json` untuk tindakan siklus hidup peluncuran instance. Ganti teks dalam *huruf miring* dengan nama grup Auto Scaling Anda.

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
    "AutoScalingGroupName": [ "my-asg" ]
  }
}
```

Jika perintah berjalan dengan sukses, EventBridge merespons dengan ARN aturan. Perhatikan ARN ini. Anda harus memasukkannya di langkah 4.

Untuk membuat aturan yang cocok untuk acara lain, ubah pola acara. Untuk informasi selengkapnya, lihat [Gunakan EventBridge untuk menangani peristiwa Auto Scaling](#).

3. Untuk menentukan fungsi Lambda yang akan digunakan sebagai target aturan, gunakan perintah [put-target](#) berikut.

```
aws events put-targets --rule my-rule --targets
  Id=1,Arn=arn:aws:lambda:region:123456789012:function:my-function
```

Pada perintah sebelumnya, *aturan* saya adalah nama yang Anda tentukan untuk aturan di langkah 2, dan nilai untuk parameter `Arn` adalah ARN dari fungsi yang Anda buat di langkah 1.

4. [Untuk menambahkan izin yang memungkinkan aturan menjalankan fungsi Lambda Anda, gunakan perintah izin tambahan Lambda berikut.](#) Perintah ini mempercayai EventBridge service principal (`events.amazonaws.com`) dan cakupan izin untuk aturan yang ditentukan.

```
aws lambda add-permission --function-name my-function --statement-id my-unique-id \
  --action 'lambda:InvokeFunction' --principal events.amazonaws.com --source-arn
  arn:aws:events:region:123456789012:rule/my-rule
```

Dalam perintah sebelumnya:

- *my-function* adalah nama fungsi Lambda yang Anda ingin aturan untuk digunakan sebagai target.
- *my-unique-id* adalah pengidentifikasi unik yang Anda tentukan untuk menggambarkan pernyataan dalam kebijakan fungsi Lambda.
- `source-arn` adalah ARN dari aturan tersebut. EventBridge

Jika perintah berjalan dengan berhasil, Anda menerima output yang mirip dengan berikut ini.

```
{
  "Statement": "{\"Sid\":\"my-unique-id\",
    \"Effect\":\"Allow\",
    \"Principal\":{\"Service\":\"events.amazonaws.com\"},
    \"Action\":\"lambda:InvokeFunction\",
    \"Resource\":\"arn:aws:lambda:us-west-2:123456789012:function:my-function\",
    \"Condition\":
      {\"ArnLike\":
        {\"AWS:SourceArn\":
          \"arn:aws:events:us-west-2:123456789012:rule/my-rule\"}}}"
}
```

Nilai Statement adalah versi string JSON dari pernyataan yang ditambahkan ke kebijakan fungsi Lambda.

5. Setelah Anda mengikuti petunjuk ini, lanjutkan ke [Tambahkan hook siklus aktif](#) sebagai langkah berikutnya.

Terima notifikasi menggunakan Amazon SNS

Anda dapat menggunakan Amazon SNS untuk menyiapkan target notifikasi (topik SNS) untuk menerima notifikasi saat tindakan siklus hidup terjadi. Amazon SNS kemudian mengirimkan notifikasi ke penerima berlangganan. Sampai langganan dikonfirmasi, tidak ada pemberitahuan yang dipublikasikan ke topik yang dikirim ke penerima.

Untuk mengatur menyiapkan notifikasi menggunakan Amazon SNS

1. [Buat topik Amazon SNS dengan menggunakan konsol Amazon SNS atau perintah create-topic berikut](#). Pastikan topik berada di Wilayah yang sama dengan grup Auto Scaling yang Anda gunakan. Untuk informasi lebih lanjut, lihat [Memulai dengan Amazon SNS](#) di Panduan Developer Amazon Simple Notification Service.

```
aws sns create-topic --name my-sns-topic
```

2. Perhatikan topik Amazon Resource Name (ARN), misalnya, `arn:aws:sns:region:123456789012:my-sns-topic` Anda membutuhkannya untuk membuat kait siklus hidup.

3. Buat peran layanan IAM untuk memberikan akses Auto Scaling Amazon EC2 ke target notifikasi Amazon SNS Anda.

Untuk memberikan Amazon EC2 Auto Scaling akses ke topik SNS Anda

- a. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
 - b. Pada panel navigasi di sebelah kiri, pilih Peran.
 - c. Pilih Buat peran.
 - d. Untuk Pilih entitas tepercaya, pilih AWS layanan.
 - e. Untuk kasus penggunaan Anda, di bawah Kasus penggunaan untuk AWS layanan lain, pilih EC2 Auto Scaling dan kemudian Akses Pemberitahuan Penskalaan Otomatis EC2.
 - f. Pilih Berikutnya dua kali untuk membuka halaman Nama, ulasan, dan buat.
 - g. Untuk nama Peran, masukkan nama untuk peran (misalnya, **my-notification-role**) dan pilih Buat peran.
 - h. Pada halaman Peran, pilih peran yang baru Anda buat untuk membuka Ringkasan yang baru. Catat peran ARN. Misalnya, `arn:aws:iam::123456789012:role/my-notification-role`. Anda membutuhkannya untuk membuat kait siklus hidup.
4. Setelah Anda mengikuti petunjuk ini, lanjutkan ke [Tambahkan kait siklus hidup \(\)AWS CLI](#) sebagai langkah berikutnya.

Menerima pemberitahuan menggunakan Amazon SQS

Anda dapat menggunakan Amazon SQS untuk menyiapkan target notifikasi agar menerima pesan saat tindakan siklus hidup terjadi. Konsumen antrian kemudian harus melakukan polling antrian SQS untuk bertindak atas notifikasi ini.

Important

Antrean FIFO tidak kompatibel dengan hook siklus aktif.

Untuk menyiapkan notifikasi menggunakan Amazon SQS

1. [Buat antrean Amazon SQS dengan menggunakan konsol Amazon SQS](#). Pastikan antrean berada di Wilayah yang sama dengan grup Auto Scaling yang Anda gunakan. Untuk informasi lebih lanjut, lihat [Memulai dengan Amazon SQS](#) di Panduan Pengembang Amazon Simple Queue Service.

2. Perhatikan antrian ARN, misalnya, `arn:aws:sqs:us-west-2:123456789012:my-sqs-queue` Anda membutuhkannya untuk membuat kait siklus hidup.
3. Buat peran layanan IAM untuk memberikan akses Auto Scaling Amazon EC2 ke target notifikasi Amazon SQS Anda.

Untuk memberikan akses Auto Scaling Amazon EC2 ke antrean SQS Anda

- a. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
 - b. Pada panel navigasi di sebelah kiri, pilih Peran.
 - c. Pilih Buat peran.
 - d. Untuk Pilih entitas tepercaya, pilih AWS layanan.
 - e. Untuk kasus penggunaan Anda, di bawah Kasus penggunaan untuk AWS layanan lain, pilih EC2 Auto Scaling dan kemudian Akses Pemberitahuan Penskalaan Otomatis EC2.
 - f. Pilih Berikutnya dua kali untuk membuka halaman Nama, ulasan, dan buat.
 - g. Untuk nama Peran, masukkan nama untuk peran (misalnya, **my-notification-role**) dan pilih Buat peran.
 - h. Pada halaman Peran, pilih peran yang baru Anda buat untuk membuka Ringkasan yang baru. Catat peran ARN. Misalnya, `arn:aws:iam::123456789012:role/my-notification-role`. Anda membutuhkannya untuk membuat kait siklus hidup.
4. Setelah Anda mengikuti petunjuk ini, lanjutkan ke [Tambahkan kait siklus hidup \(\)AWS CLI](#) sebagai langkah berikutnya.

Contoh pesan pemberitahuan untuk Amazon SNS dan Amazon SQS

Saat instance dalam keadaan menunggu, pesan dipublikasikan ke target notifikasi Amazon SNS atau Amazon SQS. Pesan ini mencakup informasi berikut:

- `LifecycleActionToken` — Token tindakan siklus aktif.
- `AccountId`— Akun AWS ID.
- `AutoScalingGroupName` — Nama grup Auto Scaling.
- `LifecycleHookName` — Nama hook siklus aktif.
- `EC2InstanceId` — ID instance EC2.
- `LifecycleTransition` — Jenis hook siklus aktif.
- `NotificationMetadata`— Metadata notifikasi.

Berikut ini adalah contoh pesan pemberitahuan.

```
Service: AWS Auto Scaling
Time: 2021-01-19T00:36:26.533Z
RequestId: 18b2ec17-3e9b-4c15-8024-ff2e8ce8786a
LifecycleActionToken: 71514b9d-6a40-4b26-8523-05e7ee35fa40
AccountId: 123456789012
AutoScalingGroupName: my-asg
LifecycleHookName: my-hook
EC2InstanceId: i-0598c7d356eba48d7
LifecycleTransition: autoscaling:EC2_INSTANCE_LAUNCHING
NotificationMetadata: hook message metadata
```

Contoh pesan pemberitahuan uji

Saat pertama kali menambahkan kait siklus hidup, pesan notifikasi pengujian akan dipublikasikan ke target notifikasi. Berikut ini adalah contoh pesan pemberitahuan pengujian.

```
Service: AWS Auto Scaling
Time: 2021-01-19T00:35:52.359Z
RequestId: 18b2ec17-3e9b-4c15-8024-ff2e8ce8786a
Event: autoscaling:TEST_NOTIFICATION
AccountId: 123456789012
AutoScalingGroupName: my-asg
AutoScalingGroupARN: arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:042cba90-ad2f-431c-9b4d-6d9055bcc9fb:autoScalingGroupName/my-asg
```

Note

Untuk contoh peristiwa yang dikirimkan dari Amazon EC2 Auto Scaling EventBridge ke, lihat [Referensi acara Auto Scaling Amazon EC2](#)

Ambil status siklus hidup target melalui metadata instance

Setiap instans Auto Scaling yang Anda luncurkan melewati beberapa status siklus hidup. Untuk memanggil tindakan kustom dari dalam instance yang bertindak pada transisi status siklus hidup tertentu, Anda harus mengambil status siklus hidup target melalui metadata instance.

Misalnya, Anda mungkin memerlukan mekanisme untuk mendeteksi penghentian instance dari dalam instance untuk menjalankan beberapa kode pada instance sebelum dihentikan. Anda dapat melakukan ini dengan menulis kode yang melakukan polling untuk status siklus hidup sebuah instance langsung dari instance. Anda kemudian dapat menambahkan hook siklus hidup ke grup Auto Scaling untuk menjaga instance tetap berjalan hingga kode Anda mengirimkan complete-lifecycle-action perintah untuk melanjutkan.

Siklus hidup instans Auto Scaling memiliki dua kondisi tunak utama— InService dan Terminated —dan dua kondisi tunak sisi—dan. Detached Standby Jika Anda menggunakan kolom hangat, siklus hidup memiliki empat kondisi tunak tambahan—Warmed:Hibernated,, Warmed:RunningWarmed:Stopped, dan. Warmed:Terminated

Saat instance bersiap untuk bertransisi ke salah satu status stabil sebelumnya, Auto Scaling Amazon EC2 memperbarui nilai item metadata instans. `autoscaling/target-lifecycle-state` Untuk mendapatkan status siklus hidup target dari dalam instance, Anda harus menggunakan Layanan Metadata Instance untuk mengambilnya dari metadata instance.

Note

Metadata instans adalah data tentang instans Amazon EC2 yang dapat digunakan aplikasi untuk menanyakan informasi instans. Layanan Metadata Instance adalah komponen on-instance yang digunakan kode lokal untuk mengakses metadata instance. Kode lokal dapat mencakup skrip data pengguna atau aplikasi yang berjalan pada instance.

Kode lokal dapat mengakses metadata instance dari instance yang sedang berjalan menggunakan salah satu dari dua metode: Instance Metadata Service Version 1 (IMDSv1) atau Instance Metadata Service Version 2 (IMDSv2). IMDSv2 menggunakan permintaan berorientasi sesi dan mengurangi beberapa jenis kerentanan yang dapat digunakan untuk mencoba mengakses metadata instance. Untuk detail tentang kedua metode ini, lihat [Menggunakan IMDSv2](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

IMDSv2

```
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" ` \  
&& curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-  
data/autoscaling/target-lifecycle-state
```

IMDSv1

```
curl http://169.254.169.254/latest/meta-data/autoscaling/target-lifecycle-state
```

Berikut ini adalah output contoh.

```
InService
```

Status siklus hidup target adalah status yang menjadi transisi instance. Status siklus hidup saat ini adalah status tempat instance berada. Ini bisa sama setelah tindakan siklus hidup selesai dan instance menyelesaikan transisinya ke status siklus hidup target. Anda tidak dapat mengambil status siklus hidup instans saat ini dari metadata instance.

Auto Scaling Amazon EC2 mulai menghasilkan status siklus hidup target pada 10 Maret 2022. Jika instance Anda bertransisi ke salah satu status siklus hidup target setelah tanggal tersebut, item status siklus hidup target ada dalam metadata instans Anda. Jika tidak, itu tidak ada, dan Anda menerima kesalahan HTTP 404.

Untuk informasi selengkapnya tentang mengambil metadata instans, lihat [Mengambil metadata instans di Panduan Pengguna Amazon EC2 untuk Instans Linux](#).

Untuk tutorial yang menunjukkan cara membuat hook siklus hidup dengan tindakan kustom dalam skrip data pengguna yang menggunakan status siklus hidup target, lihat [Tutorial: Konfigurasi data pengguna untuk mengambil status siklus hidup target melalui metadata instance](#)

Important

Untuk memastikan bahwa Anda dapat memanggil tindakan kustom sesegera mungkin, kode lokal Anda harus sering melakukan polling IMDS dan mencoba lagi kesalahan.

Tambahkan hook siklus aktif

Untuk menempatkan instans Auto Scaling ke status tunggu dan melakukan tindakan kustom padanya, Anda dapat menambahkan kait siklus hidup ke grup Auto Scaling Anda. Tindakan kustom dilakukan saat instance diluncurkan atau sebelum dihentikan. Instance tetap dalam status tunggu hingga Anda menyelesaikan tindakan siklus hidup, atau periode batas waktu berakhir.

Setelah membuat grup Auto Scaling dari grup AWS Management Console, Anda dapat menambahkan satu atau beberapa kait siklus hidup ke grup tersebut, hingga total 50 kait siklus hidup. Anda juga dapat menggunakan AWS CLI, AWS CloudFormation, atau SDK untuk menambahkan kait siklus hidup ke grup Auto Scaling saat Anda membuatnya.

Secara default, saat Anda menambahkan kait siklus hidup di konsol, Auto Scaling Amazon EC2 mengirimkan pemberitahuan peristiwa siklus hidup ke Amazon EventBridge. Menggunakan EventBridge atau skrip data pengguna adalah praktik terbaik yang disarankan. Untuk membuat hook siklus hidup yang mengirimkan notifikasi langsung ke Amazon SNS atau Amazon SQS, Anda dapat menggunakan perintah, seperti yang ditunjukkan [put-lifecycle-hook](#) pada contoh dalam topik ini.

Daftar Isi

- [Tambahkan hook siklus aktif \(konsol\)](#)
- [Tambahkan kait siklus hidup \(\)AWS CLI](#)

Tambahkan hook siklus aktif (konsol)

Ikuti langkah-langkah berikut untuk menambahkan kait siklus hidup ke grup Auto Scaling Anda. Untuk menambahkan kait siklus hidup untuk penskalaan (peluncuran instance) dan penskalaan (instance mengakhiri atau kembali ke kolam hangat), Anda harus membuat dua kait terpisah.

Sebelum memulai, konfirmasikan bahwa Anda telah menyiapkan tindakan khusus, sesuai kebutuhan, seperti yang dijelaskan dalam [Bersiaplah untuk menambahkan kait siklus hidup ke grup Auto Scaling Anda](#).

Untuk menambahkan kait siklus hidup untuk skala keluar

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda. Panel split terbuka di bagian bawah halaman.
3. Pada Manajemen Instance, di Hook siklus aktif, pilih Buat hook siklus aktif.
4. Untuk menentukan hook siklus hidup untuk skala keluar (peluncuran instance), lakukan hal berikut:
 - a. Untuk Nama hook siklus aktif, sebutkan nama untuk hook siklus aktif.
 - b. Untuk transisi Siklus Hidup, pilih Peluncuran instans.

- c. Untuk batas waktu detak jantung, tentukan jumlah waktu, dalam detik, agar instance tetap dalam keadaan menunggu saat penskalaan keluar sebelum waktu hook habis. Rentangnya adalah dari 30 hingga 7200 detik. Menetapkan periode batas waktu yang lama akan memberikan lebih banyak waktu untuk menyelesaikan tindakan kustom Anda. Kemudian, jika Anda selesai sebelum periode batas waktu berakhir, kirim [complete-lifecycle-action](#) perintah untuk mengizinkan instance melanjutkan ke status berikutnya.
 - d. Untuk hasil Default, tentukan tindakan yang akan diambil saat batas waktu pengait siklus hidup berlalu atau saat terjadi kegagalan yang tidak terduga. Anda dapat memilih untuk melanjutkan atau meninggalkan.
 - Jika Anda memilih LANJUTKAN, grup Auto Scaling dapat melanjutkan dengan kait siklus hidup lainnya dan kemudian memasukkan instance ke dalam layanan.
 - Jika Anda memilih ABANDOND, grup Auto Scaling menghentikan tindakan yang tersisa dan segera menghentikan instance.
 - e. (Opsional) Untuk metadata Pemberitahuan, tentukan informasi lain yang ingin Anda sertakan saat Auto Scaling Amazon EC2 mengirim pesan ke target notifikasi.
5. Pilih Buat.

Untuk menambahkan kait siklus hidup untuk skala di

1. Pilih Buat kait siklus hidup untuk melanjutkan tempat yang Anda tinggalkan setelah membuat hook siklus hidup untuk skala keluar.
2. Untuk menentukan hook siklus hidup untuk skala (instance yang mengakhiri atau kembali ke kolam hangat), lakukan hal berikut:
 - a. Untuk Nama hook siklus aktif, sebutkan nama untuk hook siklus aktif.
 - b. Untuk Transisi siklus aktif, pilih Penghentian instance.
 - c. Untuk batas waktu detak jantung, tentukan jumlah waktu, dalam detik, agar instance tetap dalam keadaan menunggu saat penskalaan keluar sebelum waktu hook habis. Kami merekomendasikan periode waktu tunggu singkat 30 hingga 120 detik, tergantung pada berapa banyak waktu yang Anda butuhkan untuk melakukan tugas akhir, seperti menarik log EC2 dari CloudWatch
 - d. Untuk hasil Default, tentukan tindakan yang dilakukan grup Auto Scaling saat batas waktu berlalu atau jika terjadi kegagalan yang tidak terduga. Baik ABANDOND dan CONTINUE membiarkan instance berakhir.

- Jika Anda memilih LANJUTKAN, grup Auto Scaling dapat melanjutkan dengan tindakan yang tersisa, seperti kait siklus hidup lainnya, sebelum penghentian.
 - Jika Anda memilih ABANDOND, grup Auto Scaling akan segera menghentikan instance.
- e. (Opsional) Untuk metadata Pemberitahuan, tentukan informasi lain yang ingin Anda sertakan saat Auto Scaling Amazon EC2 mengirim pesan ke target notifikasi.

3. Pilih Buat.

Tambahkan kait siklus hidup ()AWS CLI

Buat dan perbarui kait siklus hidup menggunakan perintah. [put-lifecycle-hook](#)

Untuk melakukan tindakan pada peningkatan skala, gunakan perintah berikut.

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-launch-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-transition autoscaling:EC2_INSTANCE_LAUNCHING
```

Untuk melakukan tindakan pada penurunan skala, gunakan perintah berikut.

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-termination-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-transition autoscaling:EC2_INSTANCE_TERMINATING
```

Untuk menerima pemberitahuan menggunakan Amazon SNS atau Amazon SQS, tambahkan `--notification-target-arn` opsi dan `--role-arn`

Contoh berikut membuat hook siklus hidup yang menentukan topik SNS bernama *my-sns-topic* sebagai target notifikasi.

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-termination-hook \  
  --auto-scaling-group-name my-asg \  
  --lifecycle-transition autoscaling:EC2_INSTANCE_TERMINATING \  
  --notification-target-arn arn:aws:sns:region:123456789012:my-sns-topic \  
  --role-arn arn:aws:iam::123456789012:role/my-notification-role
```

Topik tersebut menerima pemberitahuan uji dengan pasangan nilai kunci berikut.

```
"Event": "autoscaling:TEST_NOTIFICATION"
```

Secara default, [put-lifecycle-hook](#) perintah membuat kait siklus hidup dengan batas waktu detak jantung 3600 detik (satu jam).

Untuk mengubah batas waktu detak jantung untuk hook siklus hidup yang ada, tambahkan `--heartbeat-timeout` opsi, seperti yang ditunjukkan pada contoh berikut.

```
aws autoscaling put-lifecycle-hook --lifecycle-hook-name my-termination-hook \  
  --auto-scaling-group-name my-asg --heartbeat-timeout 120
```

Jika sebuah instance sudah dalam status tunggu, Anda dapat mencegah pengait siklus hidup dari waktu habis dengan merekam detak jantung, menggunakan perintah CLI [record-lifecycle-action-heartbeat](#). Ini memperpanjang periode batas waktu berdasarkan nilai batas waktu yang ditentukan ketika Anda membuat hook siklus aktif. Jika Anda selesai sebelum periode batas waktu berakhir, Anda dapat mengirim perintah [complete-lifecycle-action](#) CLI untuk memungkinkan instance melanjutkan ke status berikutnya. Untuk informasi selengkapnya dan contoh tambahan, lihat [Selesaikan tindakan siklus hidup](#).

Selesaikan tindakan siklus hidup

Saat grup Auto Scaling merespons peristiwa siklus hidup, grup tersebut akan menempatkan instance dalam status tunggu dan mengirimkan pemberitahuan peristiwa. Anda dapat melakukan tindakan kustom saat instance dalam status tunggu.

Menyelesaikan tindakan siklus hidup dengan hasil CONTINUE sangat membantu jika Anda selesai sebelum periode batas waktu berakhir. Jika Anda tidak menyelesaikan tindakan siklus hidup, kait siklus hidup akan masuk ke status yang Anda tentukan untuk hasil Default setelah periode batas waktu berakhir.

Daftar Isi

- [Selesaikan tindakan siklus hidup \(manual\)](#)
- [Selesaikan tindakan siklus hidup \(otomatis\)](#)

Selesaikan tindakan siklus hidup (manual)

Prosedur berikut adalah untuk antarmuka baris perintah dan tidak didukung di konsol. Informasi yang harus diganti, seperti ID instance atau nama grup Auto Scaling, ditampilkan dalam huruf miring.

Untuk menyelesaikan tindakan siklus hidup ()AWS CLI

1. Jika Anda memerlukan lebih banyak waktu untuk menyelesaikan tindakan kustom, gunakan [record-lifecycle-action-heartbeat](#) perintah untuk memulai ulang periode batas waktu dan menjaga instance dalam keadaan menunggu. Misalnya, jika periode batas waktu adalah satu jam, dan Anda memanggil perintah ini setelah 30 menit, instance tetap dalam keadaan menunggu selama satu jam tambahan, atau total 90 menit.

Anda dapat menentukan token tindakan siklus hidup yang Anda terima dengan [notifikasi](#), seperti yang ditunjukkan pada perintah berikut.

```
aws autoscaling record-lifecycle-action-heartbeat --lifecycle-hook-name my-launch-hook \  
  --auto-scaling-group-name my-asg --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

Atau, Anda dapat menentukan ID dari instance yang Anda terima dengan [notifikasi](#), seperti yang ditunjukkan pada perintah berikut.

```
aws autoscaling record-lifecycle-action-heartbeat --lifecycle-hook-name my-launch-hook \  
  --auto-scaling-group-name my-asg --instance-id i-1a2b3c4d
```

2. Jika Anda menyelesaikan tindakan kustom sebelum periode batas waktu berakhir, gunakan [complete-lifecycle-action](#) perintah agar grup Auto Scaling dapat melanjutkan peluncuran atau penghentian instance. Anda dapat menentukan token tindakan siklus aktif, seperti ditunjukkan dalam perintah berikut.

```
aws autoscaling complete-lifecycle-action --lifecycle-action-result CONTINUE \  
  --lifecycle-hook-name my-launch-hook --auto-scaling-group-name my-asg \  
  --lifecycle-action-token bcd2f1b8-9a78-44d3-8a7a-4dd07d7cf635
```

Atau, Anda dapat menentukan ID dari instance, seperti yang ditunjukkan pada perintah berikut.

```
aws autoscaling complete-lifecycle-action --lifecycle-action-result CONTINUE \  
  --instance-id i-1a2b3c4d --lifecycle-hook-name my-launch-hook \  
  --auto-scaling-group-name my-asg
```

Selesaikan tindakan siklus hidup (otomatis)

Jika Anda memiliki skrip data pengguna yang mengonfigurasi instance Anda setelah diluncurkan, Anda tidak perlu menyelesaikan tindakan siklus hidup secara manual. Anda dapat menambahkan [complete-lifecycle-action](#) perintah ke skrip. Skrip dapat mengambil ID instance dari metadata instance dan memberi sinyal Auto Scaling Amazon EC2 ketika skrip bootstrap telah berhasil diselesaikan.

Jika Anda belum melakukannya, perbarui skrip Anda untuk mengambil data ID instance dari metadata instance. Untuk informasi selengkapnya, lihat [Mengambil metadata instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

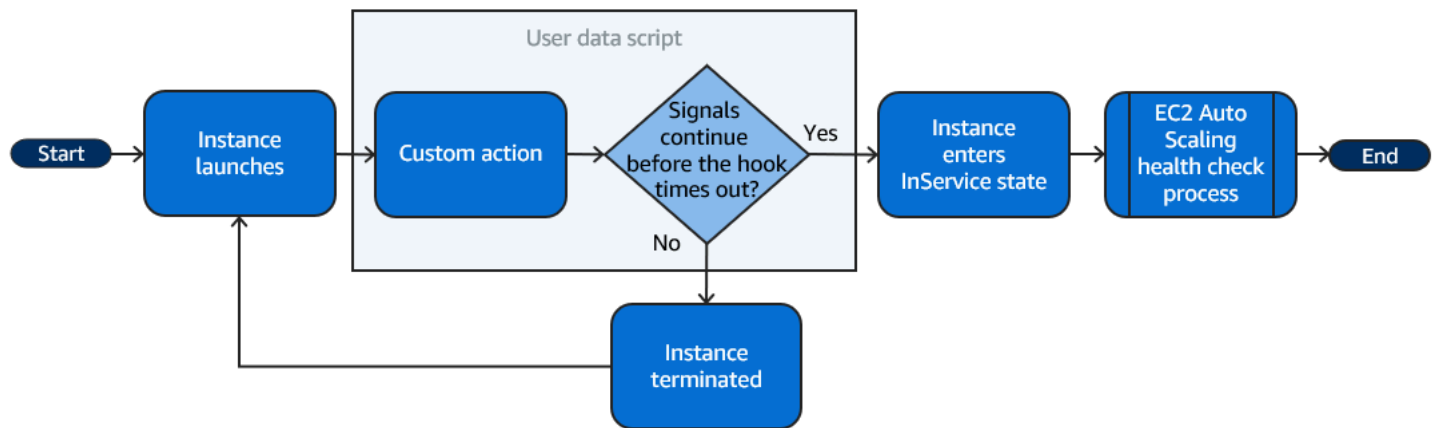
Jika Anda menggunakan Lambda, Anda juga dapat menyiapkan callback dalam kode fungsi agar siklus hidup instance dilanjutkan jika tindakan kustom berhasil. Untuk informasi selengkapnya, lihat [Tutorial: Konfigurasi kait siklus hidup yang memanggil fungsi Lambda](#).

Tutorial: Konfigurasi data pengguna untuk mengambil status siklus hidup target melalui metadata instance

Cara umum untuk membuat tindakan kustom untuk kait siklus hidup adalah dengan menggunakan notifikasi yang dikirimkan oleh Amazon EC2 Auto Scaling ke layanan lain, seperti Amazon EventBridge. Namun, Anda dapat menghindari keharusan membuat infrastruktur tambahan dengan menggunakan skrip data pengguna untuk memindahkan kode yang mengonfigurasi instance dan menyelesaikan tindakan siklus hidup ke dalam instance itu sendiri.

Tutorial berikut menunjukkan kepada Anda bagaimana untuk memulai menggunakan script data pengguna dan metadata instance. Anda membuat konfigurasi grup Auto Scaling dasar dengan skrip data pengguna yang membaca [status siklus hidup target](#) instans dalam grup Anda dan melakukan tindakan panggilan balik pada fase tertentu dari siklus hidup instans untuk melanjutkan proses peluncuran.

Ilustrasi berikut merangkum alur untuk acara scale-out saat Anda menggunakan skrip data pengguna untuk melakukan tindakan kustom. Setelah instance diluncurkan, siklus hidup instans dijeda hingga pengait siklus hidup selesai, baik dengan waktu habis atau oleh Amazon EC2 Auto Scaling yang menerima sinyal untuk melanjutkan.



Daftar Isi

- [Langkah 1: Buat peran IAM dengan izin untuk menyelesaikan tindakan siklus hidup](#)
- [Langkah 2: Buat template peluncuran dan sertakan peran IAM dan skrip data pengguna](#)
- [Langkah 3: Buat grup Auto Scaling](#)
- [Langkah 4: Tambahkan kait siklus hidup](#)
- [Langkah 5: Uji dan verifikasi fungsionalitas](#)
- [Langkah 6: Bersihkan](#)
- [Sumber daya terkait](#)

Langkah 1: Buat peran IAM dengan izin untuk menyelesaikan tindakan siklus hidup

Bila Anda menggunakan AWS CLI atau AWS SDK untuk mengirim panggilan balik untuk menyelesaikan tindakan siklus hidup, Anda harus menggunakan peran IAM dengan izin untuk menyelesaikan tindakan siklus hidup.

Untuk membuat kebijakan

1. Buka [halaman Kebijakan](#) konsol IAM, lalu pilih Buat kebijakan.
2. Pilih tab JSON.
3. Di kotak Dokumen Kebijakan, salin dan tempel dokumen kebijakan berikut ke dalam kotak. Ganti *teks contoh* dengan nomor akun Anda dan nama grup Auto Scaling yang ingin Anda buat (**TestAutoScalingEvent-group**).

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

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "autoscaling:CompleteLifecycleAction"  
    ],  
    "Resource":  
      "arn:aws:autoscaling:*:123456789012:autoScalingGroup:*:autoScalingGroupName/  
TestAutoScalingEvent-group"  
  }  
]  
}
```

4. Pilih Berikutnya.
5. Untuk Nama kebijakan, masukkan **TestAutoScalingEvent-policy**. Pilih Buat kebijakan.

Setelah selesai membuat kebijakan, Anda dapat membuat peran yang menggunakannya.

Untuk membuat peran

1. Pada panel navigasi di sebelah kiri, pilih Peran.
2. Pilih Buat peran.
3. Untuk Pilih entitas tepercaya, pilih AWS layanan.
4. Untuk kasus penggunaan Anda, pilih EC2 dan kemudian pilih Berikutnya.
5. Di bawah Tambahkan izin, pilih kebijakan yang Anda buat (TestAutoScalingEvent-policy). Lalu, pilih Selanjutnya.
6. Pada halaman Nama, tinjau, dan buat, untuk nama Peran, masukkan **TestAutoScalingEvent-role** dan pilih Buat peran.

Langkah 2: Buat template peluncuran dan sertakan peran IAM dan skrip data pengguna

Buat template peluncuran untuk digunakan dengan grup Auto Scaling Anda. Sertakan peran IAM yang Anda buat dan skrip data pengguna sampel yang disediakan.

Untuk membuat templat peluncuran

1. Buka [halaman Peluncuran templat](#) konsol Amazon EC2.

2. Pilih Buat templat peluncuran.
3. Untuk nama template Luncurkan, masukkan **TestAutoScalingEvent-template**.
4. Di bawah panduan Auto Scaling, pilih kotak centang.
5. Untuk Gambar Aplikasi dan OS (Gambar Mesin Amazon), pilih Amazon Linux 2 (HVM), Jenis Volume SSD, 64-bit (x86) dari daftar Mulai Cepat.
6. Untuk jenis Instance, pilih jenis instans Amazon EC2 (misalnya, "t2.micro").
7. Untuk Detail lanjutan, perluas bagian tersebut untuk melihat kolomnya.
8. Untuk profil instans IAM, pilih nama profil instans IAM dari peran IAM Anda (-role) **TestAutoScalingEvent**. Profil instance adalah kontainer untuk peran IAM yang memungkinkan Amazon EC2 melewati peran IAM ke suatu instance saat instance diluncurkan.

Saat Anda menggunakan konsol IAM untuk membuat peran IAM, konsol secara otomatis membuat profil instance dengan nama yang sama dengan peran yang sesuai.

9. Untuk data Pengguna, salin dan tempel contoh skrip data pengguna berikut ke dalam bidang. Ganti teks sampel `group_name` dengan nama grup Auto Scaling yang ingin Anda buat dan `region` dengan grup Auto Scaling yang Wilayah AWS Anda inginkan untuk digunakan.

```
#!/bin/bash

function get_target_state {
    echo $(curl -s http://169.254.169.254/latest/meta-data/autoscaling/target-lifecycle-state)
}

function get_instance_id {
    echo $(curl -s http://169.254.169.254/latest/meta-data/instance-id)
}

function complete_lifecycle_action {
    instance_id=$(get_instance_id)
    group_name='TestAutoScalingEvent-group'
    region='us-west-2'

    echo $instance_id
    echo $region
    echo $(aws autoscaling complete-lifecycle-action \
        --lifecycle-hook-name TestAutoScalingEvent-hook \
        --auto-scaling-group-name $group_name \
        --lifecycle-action-result CONTINUE \
```

```
--instance-id $instance_id \  
--region $region)  
}  
  
function main {  
  while true  
  do  
    target_state=$(get_target_state)  
    if [ \"$target_state\" = \"InService\" ]; then  
      # Change hostname  
      export new_hostname=\"${group_name}-${instance_id}\"  
      hostname $new_hostname  
      # Send callback  
      complete_lifecycle_action  
      break  
    fi  
    echo $target_state  
    sleep 5  
  done  
}  
  
main
```

Skrip data pengguna sederhana ini melakukan hal berikut:

- Memanggil metadata instance untuk mengambil status siklus hidup target dan ID instans dari metadata instance
- Mengambil status siklus hidup target berulang kali hingga berubah InService
- Mengubah nama host instance menjadi ID instans yang ditambahkan dengan nama grup Auto Scaling, jika status siklus hidup target InService
- Mengirim panggilan balik dengan memanggil perintah complete-lifecycle-action CLI untuk memberi sinyal Auto Scaling Amazon EC2 ke proses peluncuran EC2 CONTINUE

10. Pilih Buat templat peluncuran.

11. Pada halaman konfirmasi, pilih Buat grup Auto Scaling.

Note

Untuk contoh lain yang dapat Anda gunakan sebagai referensi untuk mengembangkan skrip data pengguna, lihat [GitHub repositori](#) untuk Amazon EC2 Auto Scaling.

Langkah 3: Buat grup Auto Scaling

Setelah Anda membuat template peluncuran, buat grup Auto Scaling.

Untuk membuat grup Auto Scaling

1. Pada halaman Choose launch template atau konfigurasi, untuk nama grup Auto Scaling, masukkan nama untuk grup Auto Scaling (). **TestAutoScalingEvent-group**
2. Pilih Berikutnya untuk pergi ke halaman opsi peluncuran Choose instance.
3. Untuk Jaringan, pilih VPC.
4. Untuk Availability Zones dan subnet, pilih satu atau beberapa subnet dari satu atau beberapa Availability Zone.
5. Di bagian Persyaratan tipe instans, gunakan pengaturan default untuk menyederhanakan langkah ini. (Jangan mengganti template peluncuran.) Untuk tutorial ini, Anda hanya akan meluncurkan satu On-Demand Instance menggunakan jenis instance yang ditentukan dalam template peluncuran Anda.
6. Pilih Lewati untuk meninjau di bagian bawah layar.
7. Pada halaman Tinjauan, tinjau detail grup Auto Scaling Anda, lalu pilih grup Buat Auto Scaling.

Langkah 4: Tambahkan kait siklus hidup

Tambahkan kait siklus hidup untuk menahan instance dalam status tunggu hingga tindakan siklus hidup Anda selesai.

Untuk menambahkan hook siklus aktif

1. Buka [halaman grup Auto Scaling](#) di konsol Amazon EC2.
2. Pilih kotak centang di samping grup Auto Scaling Anda. Panel split terbuka di bagian bawah halaman.
3. Di panel bawah, pada tab Manajemen instans, di kait Siklus Hidup, pilih Buat kait siklus hidup.

4. Untuk menentukan hook siklus hidup untuk skala keluar (peluncuran instance), lakukan hal berikut:
 - a. Untuk nama kait Siklus Hidup, masukkan **TestAutoScalingEvent-hook**
 - b. Untuk transisi Siklus Hidup, pilih Peluncuran instans.
 - c. Untuk batas waktu detak jantung, masukkan **300** jumlah detik untuk menunggu panggilan balik dari skrip data pengguna Anda.
 - d. Untuk hasil Default, pilih ABANDOND. Jika waktu hook habis tanpa menerima panggilan balik dari skrip data pengguna Anda, grup Auto Scaling akan menghentikan instance baru.
 - e. (Opsional) Biarkan metadata Pemberitahuan tetap kosong.
5. Pilih Buat.

Langkah 5: Uji dan verifikasi fungsionalitas

Untuk menguji fungsionalitas, perbarui grup Auto Scaling dengan meningkatkan kapasitas grup Auto Scaling yang diinginkan sebanyak 1. Skrip data pengguna berjalan dan mulai memeriksa status siklus hidup target instans segera setelah instance diluncurkan. Skrip mengubah nama host dan mengirimkan tindakan panggilan balik saat status siklus hidup target. InService Ini biasanya hanya membutuhkan beberapa detik untuk menyelesaikannya.

Untuk meningkatkan ukuran grup Auto Scaling

1. Buka [halaman grup Auto Scaling](#) di konsol Amazon EC2.
2. Pilih kotak centang di samping grup Auto Scaling Anda. Lihat detail di panel bawah sambil tetap melihat baris atas panel atas.
3. Di panel bawah, pada tab Detail, pilih Detail grup, Edit.
4. Untuk Kapasitas yang diinginkan, naikkan nilai saat ini sebesar 1.
5. Pilih Perbarui. Saat instance diluncurkan, kolom Status di panel atas menampilkan status Memperbarui kapasitas.

Setelah meningkatkan kapasitas yang diinginkan, Anda dapat memverifikasi bahwa instans Anda telah berhasil diluncurkan dan tidak dihentikan dari deskripsi aktivitas penskalaan.

Untuk melihat aktivitas penskalaan

1. Kembali ke halaman grup Auto Scaling dan pilih grup Anda.

2. Pada tab Aktivitas, di bawah Riwayat aktivitas, kolom Status menunjukkan apakah grup Auto Scaling Anda telah berhasil meluncurkan instance.
3. Jika skrip data pengguna gagal, setelah periode batas waktu berlalu, Anda akan melihat aktivitas penskalaan dengan status Canceled dan pesan status. Instance failed to complete user's Lifecycle Action: Lifecycle Action with token e85eb647-4fe0-4909-b341-a6c42EXAMPLE was abandoned: Lifecycle Action Completed with ABANDON Result

Langkah 6: Bersihkan

Jika Anda selesai bekerja dengan sumber daya yang Anda buat untuk tutorial ini, gunakan langkah-langkah berikut untuk menghapusnya.

Untuk menghapus kait siklus hidup

1. Buka [halaman grup Auto Scaling](#) di konsol Amazon EC2.
2. Pilih kotak centang di samping grup Auto Scaling Anda.
3. Pada tab Manajemen instans, di kait Siklus Hidup, pilih hook siklus hidup ().
TestAutoScalingEvent-hook
4. Pilih Tindakan, Hapus.
5. Pilih Hapus lagi untuk mengonfirmasi.

Untuk menghapus template peluncuran

1. Buka [halaman Peluncuran templat](#) konsol Amazon EC2.
2. Pilih template peluncuran Anda (TestAutoScalingEvent-template) dan kemudian pilih Tindakan, Hapus template.
3. Saat diminta konfirmasi, ketik **Delete** untuk mengonfirmasi penghapusan templat peluncuran yang ditentukan, lalu pilih Hapus.

Jika Anda sudah selesai bekerja dengan contoh grup Auto Scaling, hapus saja. Anda juga dapat menghapus peran IAM dan kebijakan izin yang Anda buat.

Untuk menghapus grup Auto Scaling

1. Buka [halaman grup Auto Scaling](#) di konsol Amazon EC2.

2. Pilih kotak centang di samping grup Auto Scaling (**TestAutoScalingEvent-group**) dan pilih Hapus.
3. Saat diminta konfirmasi, ketik **delete** untuk mengonfirmasi penghapusan grup Auto Scaling yang ditentukan, lalu pilih Hapus.

Ikon memuat di kolom Nama menunjukkan bahwa grup Auto Scaling dihapus. Dibutuhkan beberapa menit untuk menghentikan instance dan menghapus grup.

Untuk menghapus peran IAM

1. Buka [halaman Peran](#) konsol IAM.
2. Pilih peran fungsi (TestAutoScalingEvent-role)
3. Pilih Hapus.
4. Saat diminta konfirmasi, ketikkan nama peran, lalu pilih Hapus.

Untuk menghapus kebijakan IAM

1. Buka [halaman Kebijakan](#) konsol IAM.
2. Pilih kebijakan yang Anda buat (TestAutoScalingEvent-policy).
3. Pilih Tindakan, Hapus.
4. Saat diminta konfirmasi, ketikkan nama kebijakan, lalu pilih Hapus.

Sumber daya terkait

Topik terkait berikut dapat membantu saat Anda mengembangkan kode yang memanggil tindakan pada instance berdasarkan data yang tersedia dalam metadata instance.

- [Ambil status siklus hidup target melalui metadata instance](#). Bagian ini menjelaskan status siklus hidup untuk kasus penggunaan lainnya, seperti penghentian instance.
- [Tambahkan hook siklus aktif \(konsol\)](#). Prosedur ini menunjukkan kepada Anda cara menambahkan kait siklus hidup untuk skala keluar (peluncuran instance) dan skala (instance mengakhiri atau kembali ke kolam hangat).
- [Kategori metadata instans](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux. Topik ini mencantumkan semua kategori metadata instans yang dapat Anda gunakan untuk menjalankan tindakan pada instans EC2.

Untuk tutorial yang menunjukkan cara menggunakan Amazon EventBridge untuk membuat aturan yang menjalankan fungsi Lambda berdasarkan peristiwa yang terjadi pada instance di grup Auto Scaling, lihat. [Tutorial: Konfigurasi kait siklus hidup yang memanggil fungsi Lambda](#)

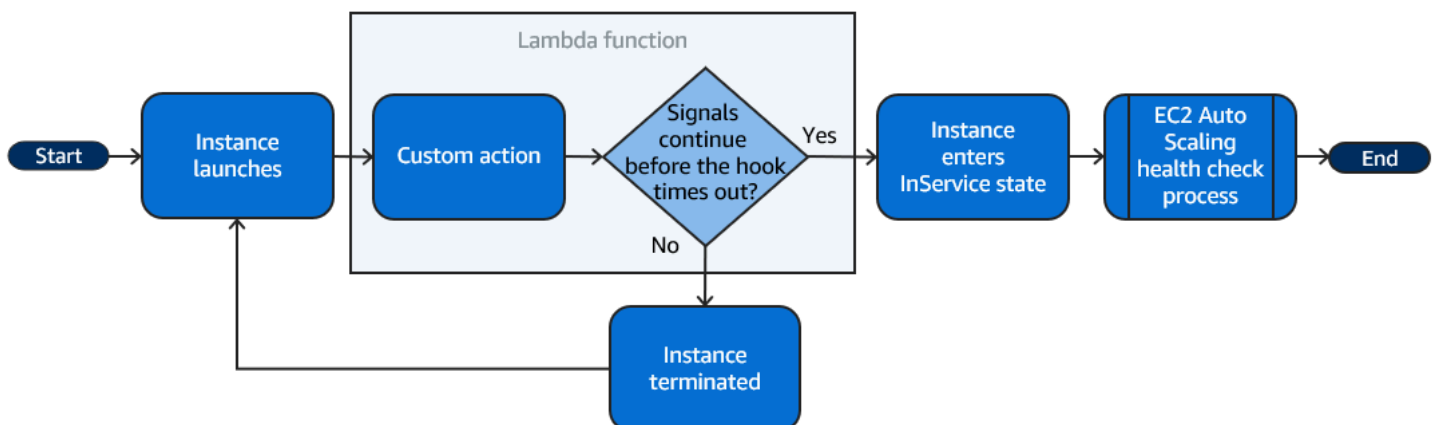
Tutorial: Konfigurasi kait siklus hidup yang memanggil fungsi Lambda

Dalam latihan ini, Anda membuat EventBridge aturan Amazon yang menyertakan pola filter yang bila dicocokkan, memanggil AWS Lambda fungsi sebagai target aturan. Kami menyediakan pola filter dan kode fungsi sampel untuk digunakan.

Jika semuanya dikonfigurasi dengan benar, di akhir tutorial ini, fungsi Lambda melakukan tindakan kustom saat instance diluncurkan. Tindakan kustom hanya mencatat peristiwa di aliran CloudWatch log Log yang terkait dengan fungsi Lambda.

Fungsi Lambda juga melakukan callback untuk membiarkan siklus hidup instance berjalan jika tindakan ini berhasil, tetapi memungkinkan instance meninggalkan peluncuran dan mengakhiri jika tindakan gagal.

Ilustrasi berikut merangkum alur untuk acara scale-out saat Anda menggunakan fungsi Lambda untuk melakukan tindakan kustom. Setelah instance diluncurkan, siklus hidup instans dijeda hingga pengait siklus hidup selesai, baik dengan waktu habis atau oleh Amazon EC2 Auto Scaling yang menerima sinyal untuk melanjutkan.



Daftar Isi

- [Prasyarat](#)
- [Langkah 1: Buat peran IAM dengan izin untuk menyelesaikan tindakan siklus hidup](#)
- [Langkah 2: Buat fungsi Lambda](#)
- [Langkah 3: Buat EventBridge aturan](#)

- [Langkah 4: Tambahkan kait siklus hidup](#)
- [Langkah 5: Uji dan verifikasi acara](#)
- [Langkah 6: Bersihkan](#)
- [Sumber daya terkait](#)

Prasyarat

Sebelum Anda memulai tutorial ini, buat grup Auto Scaling, jika Anda belum memilikinya. Untuk membuat grup Auto Scaling, buka halaman grup [Auto Scaling di konsol Amazon EC2](#) dan pilih grup Create Auto Scaling.

Langkah 1: Buat peran IAM dengan izin untuk menyelesaikan tindakan siklus hidup

Sebelum membuat fungsi Lambda, Anda harus terlebih dahulu membuat peran eksekusi dan kebijakan izin agar Lambda dapat menyelesaikan kait siklus hidup.

Untuk membuat kebijakan

1. Buka [halaman Kebijakan](#) konsol IAM, lalu pilih Buat kebijakan.
2. Pilih tab JSON.
3. Di kotak Dokumen Kebijakan, tempelkan dokumen kebijakan berikut ke dalam kotak, ganti teks yang *dicetak miring* dengan nomor akun dan nama grup Auto Scaling Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CompleteLifecycleAction"
      ],
      "Resource":
        "arn:aws:autoscaling:*:123456789012:autoScalingGroup:*:autoScalingGroupName/my-  
asg"
    }
  ]
}
```


4. Pilih Berikutnya.

5. Untuk Nama kebijakan, masukkan **LogAutoScalingEvent-policy**. Pilih Buat kebijakan.

Setelah selesai membuat kebijakan, Anda dapat membuat peran yang menggunakannya.

Untuk membuat peran

1. Pada panel navigasi di sebelah kiri, pilih Peran.
2. Pilih Buat peran.
3. Untuk Pilih entitas tepercaya, pilih AWS layanan.
4. Untuk kasus penggunaan Anda, pilih Lambda dan kemudian pilih Berikutnya.
5. Di bawah Tambahkan izin, pilih kebijakan yang Anda buat (LogAutoScalingEvent-policy) dan nama kebijakan tersebut. AWSLambdaBasicExecutionRole Lalu, pilih Selanjutnya.

 Note

AWSLambdaBasicExecutionRoleKebijakan ini memiliki izin yang diperlukan fungsi untuk menulis log ke CloudWatch Log.

6. Pada halaman Nama, tinjau, dan buat, untuk nama Peran, masukkan **LogAutoScalingEvent-role** dan pilih Buat peran.

Langkah 2: Buat fungsi Lambda

Buat fungsi Lambda untuk berfungsi sebagai target acara. Contoh fungsi Lambda, yang ditulis dalam Node.js, dipanggil EventBridge saat peristiwa pencocokan dipancarkan oleh Amazon EC2 Auto Scaling.

Untuk membuat fungsi Lambda

1. Buka [halaman Fungsi](#) di konsol Lambda.
2. Pilih Create function, Author dari awal.
3. Di bagian Informasi dasar, untuk Nama fungsi, masukkan **LogAutoScalingEvent**.
4. Untuk Runtime, pilih Node.js 18.x.
5. Gulir ke bawah dan pilih Ubah peran eksekusi default, lalu untuk peran Eksekusi, pilih Gunakan peran yang ada.
6. Untuk peran yang ada, pilih LogAutoScalingEvent-role.

7. Tinggalkan nilai default lainnya.
8. Pilih Buat fungsi. Anda dikembalikan ke kode dan konfigurasi fungsi.
9. Dengan LogAutoScalingEvent fungsi Anda masih terbuka di konsol, di bawah Sumber kode, di editor, tempelkan kode contoh berikut ke dalam file bernama index.mjs.

```
import { AutoScalingClient, CompleteLifecycleActionCommand } from "@aws-sdk/client-auto-scaling";
export const handler = async(event) => {
  console.log('LogAutoScalingEvent');
  console.log('Received event:', JSON.stringify(event, null, 2));
  var autoscaling = new AutoScalingClient({ region: event.region });
  var eventDetail = event.detail;
  var params = {
    AutoScalingGroupName: eventDetail['AutoScalingGroupName'], /* required */
    LifecycleActionResult: 'CONTINUE', /* required */
    LifecycleHookName: eventDetail['LifecycleHookName'], /* required */
    InstanceId: eventDetail['EC2InstanceId'],
    LifecycleActionToken: eventDetail['LifecycleActionToken']
  };
  var response;
  const command = new CompleteLifecycleActionCommand(params);
  try {
    var data = await autoscaling.send(command);
    console.log(data); // successful response
    response = {
      statusCode: 200,
      body: JSON.stringify('SUCCESS'),
    };
  } catch (err) {
    console.log(err, err.stack); // an error occurred
    response = {
      statusCode: 500,
      body: JSON.stringify('ERROR'),
    };
  }
  return response;
};
```

Kode ini hanya mencatat peristiwa sehingga, di akhir tutorial ini, Anda dapat melihat peristiwa muncul di aliran CloudWatch log Log yang terkait dengan fungsi Lambda ini.

10. Pilih Deploy.

Langkah 3: Buat EventBridge aturan

Buat EventBridge aturan untuk menjalankan fungsi Lambda Anda. Untuk informasi selengkapnya tentang penggunaan EventBridge, lihat [Gunakan EventBridge untuk menangani peristiwa Auto Scaling](#).

Untuk membuat aturan menggunakan konsol

1. Buka [konsol EventBridge](#).
2. Di panel navigasi, pilih Aturan.
3. Pilih Buat aturan.
4. Untuk Tentukan detail aturan, lakukan hal berikut:
 - a. Untuk Nama, masukkan **LogAutoScalingEvent-rule**.
 - b. Untuk Bus peristiwa, pilih default. Ketika Layanan AWS di akun Anda menghasilkan acara, itu selalu masuk ke bus acara default akun Anda.
 - c. Untuk Tipe aturan, pilih Aturan dengan pola peristiwa.
 - d. Pilih Selanjutnya.
5. Untuk Pola peristiwa build, lakukan hal berikut ini:
 - a. Untuk sumber Acara, pilih AWS acara atau acara EventBridge mitra.
 - b. Gulir ke bawah ke pola Acara, dan lakukan hal berikut:
 - c.
 - i. Untuk Sumber peristiwa, pilih Layanan AWS.
 - ii. Untuk Layanan AWS, pilih Auto Scaling.
 - iii. Untuk Jenis peristiwa, pilih Peluncuran dan Penghentian Instans.
 - iv. Secara default, aturan cocok dengan peristiwa scale-in atau scale-out apa pun. Untuk membuat aturan yang memberi tahu Anda saat ada peristiwa penskalaan dan instance dimasukkan ke dalam status tunggu karena kait siklus hidup, pilih Peristiwa instance Spesifik dan pilih Tindakan Siklus Hidup Peluncuran Instans EC2.
 - v. Secara default, aturan cocok dengan grup Auto Scaling di Wilayah. Untuk membuat aturan cocok dengan grup Auto Scaling tertentu, pilih Nama grup tertentu dan pilih grup.
 - vi. Pilih Selanjutnya.
6. Untuk Pilih target, lakukan hal berikut:
 - a. Untuk Tipe target, pilih Layanan AWS.

- b. Untuk Pilih target, pilih Fungsi Lambda.
 - c. Untuk Fungsi, pilih LogAutoScalingEvent.
 - d. Pilih Selanjutnya dua kali.
7. Pada halaman Tinjau dan buat, pilih Buat aturan.

Langkah 4: Tambahkan kait siklus hidup

Di bagian ini, Anda menambahkan kait siklus hidup sehingga Lambda menjalankan fungsi Anda pada instance saat peluncuran.

Untuk menambahkan hook siklus aktif

1. Buka [halaman grup Auto Scaling](#) di konsol Amazon EC2.
2. Pilih kotak centang di samping grup Auto Scaling Anda. Panel split terbuka di bagian bawah halaman.
3. Di panel bawah, pada tab Manajemen instans, di kait Siklus Hidup, pilih Buat kait siklus hidup.
4. Untuk menentukan hook siklus hidup untuk skala keluar (peluncuran instance), lakukan hal berikut:
 - a. Untuk nama kait Siklus Hidup, masukkan **LogAutoScalingEvent-hook**
 - b. Untuk transisi Siklus Hidup, pilih Peluncuran instans.
 - c. Untuk batas waktu Detak Jantung, masukkan **300** jumlah detik untuk menunggu panggilan balik dari fungsi Lambda Anda.
 - d. Untuk hasil Default, pilih ABANDOND. Ini berarti bahwa grup Auto Scaling akan menghentikan instance baru jika waktu pengait habis tanpa menerima panggilan balik dari fungsi Lambda Anda.
 - e. (Opsional) Biarkan metadata Pemberitahuan kosong. Data peristiwa yang kami berikan EventBridge berisi semua informasi yang diperlukan untuk menjalankan fungsi Lambda.
5. Pilih Buat.

Langkah 5: Uji dan verifikasi acara

Untuk menguji acara, perbarui grup Auto Scaling dengan meningkatkan kapasitas yang diinginkan dari grup Auto Scaling sebanyak 1. Fungsi Lambda Anda dipanggil dalam beberapa detik setelah meningkatkan kapasitas yang diinginkan.

Untuk meningkatkan ukuran grup Auto Scaling

1. Buka [halaman grup Auto Scaling](#) di konsol Amazon EC2.
2. Pilih kotak centang di samping grup Auto Scaling Anda untuk melihat detail di panel bawah dan masih melihat baris atas panel atas.
3. Di panel bawah, pada tab Detail, pilih Detail grup, Edit.
4. Untuk Kapasitas yang diinginkan, naikkan nilai saat ini sebesar 1.
5. Pilih Perbarui. Saat instance diluncurkan, kolom Status di panel atas menampilkan status Memperbarui kapasitas.

Setelah meningkatkan kapasitas yang diinginkan, Anda dapat memverifikasi bahwa fungsi Lambda Anda dipanggil.

Untuk melihat output dari fungsi Lambda Anda

1. Buka [halaman Grup log](#) CloudWatch konsol.
2. Pilih nama grup log untuk fungsi Lambda Anda (/aws/lambda/LogAutoScalingEvent).
3. Pilih nama aliran log untuk melihat data yang disediakan oleh fungsi untuk tindakan siklus hidup.

Selanjutnya, Anda dapat memverifikasi bahwa instans Anda telah berhasil diluncurkan dari deskripsi aktivitas penskalaan.

Untuk melihat aktivitas penskalaan

1. Kembali ke halaman grup Auto Scaling dan pilih grup Anda.
2. Pada tab Aktivitas, di bawah Riwayat aktivitas, kolom Status menunjukkan apakah grup Auto Scaling Anda telah berhasil meluncurkan instance.
 - Jika tindakan berhasil, aktivitas penskalaan akan memiliki status “Sukses”.
 - Jika gagal, setelah menunggu beberapa menit, Anda akan melihat aktivitas penskalaan dengan status “Dibatalkan” dan pesan status “Instance gagal menyelesaikan Tindakan Siklus Hidup pengguna: Tindakan Siklus Hidup dengan token E85EB647-4FE0-4909-B341-A6C42Example ditinggalkan: Tindakan Siklus Hidup Selesai dengan Hasil ABANDONOR”.

Untuk mengurangi ukuran grup Auto Scaling

Jika Anda tidak memerlukan instance tambahan yang Anda luncurkan untuk pengujian ini, Anda dapat membuka tab Detail dan mengurangi kapasitas yang diinginkan sebesar 1.

Langkah 6: Bersihkan

Jika Anda selesai bekerja dengan sumber daya yang Anda buat hanya untuk tutorial ini, gunakan langkah-langkah berikut untuk menghapusnya.

Untuk menghapus kait siklus hidup

1. Buka [halaman grup Auto Scaling](#) di konsol Amazon EC2.
2. Pilih kotak centang di samping grup Auto Scaling Anda.
3. Pada tab Manajemen instans, di kait Siklus Hidup, pilih hook siklus hidup (`LogAutoScalingEvent-hook`).
4. Pilih Tindakan, Hapus.
5. Pilih Hapus lagi untuk mengonfirmasi.

Untuk menghapus EventBridge aturan Amazon

1. Buka [halaman Aturan](#) di EventBridge konsol Amazon.
2. Di bawah bus acara, pilih bus acara yang terkait dengan aturan (`Default`).
3. Pilih kotak centang di sebelah aturan Anda (`LogAutoScalingEvent-rule`).
4. Pilih Hapus.
5. Saat diminta konfirmasi, ketikkan nama aturan lalu pilih Hapus.

Jika Anda sudah selesai bekerja dengan fungsi contoh, hapus fungsi tersebut. Anda juga dapat menghapus grup log yang menyimpan log fungsi, serta kebijakan peran dan izin eksekusi yang Anda buat.

Untuk menghapus fungsi Lambda

1. Buka [halaman Fungsi](#) di konsol Lambda.
2. Pilih fungsi (`LogAutoScalingEvent`).
3. Pilih Tindakan, Hapus.
4. Saat diminta konfirmasi, ketik **delete** untuk mengonfirmasi penghapusan fungsi yang ditentukan dan kemudian pilih Hapus.

Untuk menghapus grup log

1. Buka [halaman Grup log](#) CloudWatch konsol.
2. Pilih grup log fungsi (/aws/lambda/LogAutoScalingEvent).
3. Pilih Tindakan, Hapus grup log.
4. Di kotak dialog Hapus grup log, pilih Hapus.

Untuk menghapus peran eksekusi

1. Buka [halaman Peran](#) konsol IAM.
2. Pilih peran fungsi (LogAutoScalingEvent-role)
3. Pilih Hapus.
4. Saat diminta konfirmasi, ketikkan nama peran, lalu pilih Hapus.

Untuk menghapus kebijakan IAM

1. Buka [halaman Kebijakan](#) konsol IAM.
2. Pilih kebijakan yang Anda buat (LogAutoScalingEvent-policy).
3. Pilih Tindakan, Hapus.
4. Saat diminta konfirmasi, ketikkan nama kebijakan, lalu pilih Hapus.

Sumber daya terkait

Topik terkait berikut dapat membantu saat Anda membuat EventBridge aturan berdasarkan peristiwa yang terjadi pada instance di grup Auto Scaling Anda.

- [Gunakan EventBridge untuk menangani peristiwa Auto Scaling](#). Bagian ini menunjukkan contoh peristiwa untuk kasus penggunaan lainnya, termasuk peristiwa untuk skala.
- [Tambahkan hook siklus aktif \(konsol\)](#). Prosedur ini menunjukkan kepada Anda cara menambahkan kait siklus hidup untuk skala keluar (peluncuran instance) dan skala (instance mengakhiri atau kembali ke kolam hangat).

Untuk tutorial yang menunjukkan cara menggunakan Layanan Metadata Instance (IMDS) untuk menjalankan tindakan dari dalam instance itu sendiri, lihat. [Tutorial: Konfigurasi data pengguna untuk mengambil status siklus hidup target melalui metadata instance](#)

Kolam hangat untuk Auto Scaling Amazon EC2

Kolam hangat memberi Anda kemampuan untuk mengurangi latensi untuk aplikasi Anda yang memiliki waktu boot yang sangat lama, misalnya, karena instance perlu menulis sejumlah besar data ke disk. Dengan kolam hangat, Anda tidak perlu lagi menyediakan grup Auto Scaling secara berlebihan untuk mengelola latensi guna meningkatkan kinerja aplikasi. Untuk informasi selengkapnya, lihat posting blog berikut [Menskalakan aplikasi Anda lebih cepat dengan EC2 Auto Scaling Warm Pools](#).

Important

Membuat kolam hangat ketika tidak diperlukan dapat menyebabkan biaya yang tidak perlu. Jika waktu boot pertama Anda tidak menyebabkan masalah latensi yang nyata untuk aplikasi Anda, mungkin Anda tidak perlu menggunakan kolam hangat.

Topik

- [Konsep inti](#)
- [Prasyarat](#)
- [Perbarui contoh di kolam yang hangat](#)
- [Sumber daya terkait](#)
- [Batasan](#)
- [Gunakan kait siklus hidup dengan kolam hangat](#)
- [Buat kolam hangat untuk grup Auto Scaling](#)
- [Lihat status pemeriksaan kesehatan dan alasan kegagalan pemeriksaan kesehatan](#)
- [Contoh untuk membuat dan mengelola kolam hangat dengan AWS CLI](#)

Konsep inti

Sebelum memulai, biasakan diri Anda dengan konsep inti berikut:

Kolam hangat

Kolam hangat adalah kumpulan instans EC2 pra-inisialisasi yang berada di samping grup Auto Scaling. Kapan pun aplikasi Anda perlu diskalakan keluar, grup Auto Scaling dapat menggunakan

kolam hangat untuk memenuhi kapasitas baru yang diinginkan. Ini membantu Anda memastikan bahwa instance siap untuk mulai melayani lalu lintas aplikasi dengan cepat, mempercepat respons terhadap peristiwa penskalaan. Sebagai contoh meninggalkan kolam hangat, mereka menghitung ke arah kapasitas yang diinginkan dari kelompok. Ini dikenal sebagai awal yang hangat.

Saat instance berada di kolam hangat, kebijakan penskalaan Anda hanya dapat ditingkatkan jika nilai metrik dari instans yang berada di `InService` status lebih besar daripada ambang batas tinggi alarm kebijakan penskalaan (yang sama dengan pemanfaatan target kebijakan penskalaan pelacakan target).

Ukuran kolam hangat

Secara default, ukuran kolam hangat dihitung sebagai perbedaan antara kapasitas maksimum grup Auto Scaling dan kapasitas yang diinginkan. Misalnya, jika kapasitas yang diinginkan dari grup Auto Scaling Anda adalah 6 dan kapasitas maksimum adalah 10, ukuran kolam hangat Anda akan menjadi 4 saat Anda pertama kali mengatur kolam hangat dan kolam diinisialisasi.

Untuk menentukan kapasitas maksimum kolam hangat secara terpisah, tetapkan nilai untuk kapasitas persiapan maksimum yang lebih besar dari kapasitas grup saat ini. Ketika Anda menetapkan nilai untuk kapasitas disiapkan maksimum, ukuran kolam hangat dihitung sebagai perbedaan antara kapasitas maksimum yang disiapkan dan kapasitas grup yang diinginkan saat ini. Misalnya, jika kapasitas yang diinginkan dari grup Auto Scaling Anda adalah 6, jika kapasitas maksimum adalah 10, dan jika kapasitas maksimum yang disiapkan adalah 8, ukuran kolam hangat Anda akan menjadi 2 saat Anda pertama kali mengatur kolam hangat dan kolam sedang diinisialisasi.

Anda mungkin hanya perlu menggunakan opsi kapasitas maksimum yang disiapkan saat bekerja dengan grup Auto Scaling besar untuk mengelola manfaat biaya memiliki kolam hangat. Misalnya, grup Auto Scaling dengan 1.000 instans, kapasitas maksimum 1.500 (untuk menyediakan kapasitas ekstra untuk lonjakan lalu lintas darurat), dan kolam hangat 100 instance dapat membantu Anda mencapai tujuan Anda lebih baik daripada menyimpan 500 instans yang dicadangkan untuk penggunaan masa depan di dalam kolam hangat.

Ukuran kolam hangat minimum

Pertimbangkan untuk menggunakan pengaturan ukuran minimum untuk secara statis mengatur jumlah minimum instance untuk dipertahankan di kolam hangat. Tidak ada ukuran minimum yang ditetapkan secara default.

Status contoh kolam hangat

Anda dapat menyimpan contoh di kolam hangat di salah satu dari tiga negara bagian: `Stopped`, `Running`, atau `Hibernated`. Menjaga contoh dalam suatu `Stopped` keadaan adalah cara yang efektif untuk meminimalkan biaya. Dengan instans berhenti, Anda hanya membayar volume yang Anda gunakan dan alamat IP Elastis yang dilampirkan ke instans.

Atau, Anda dapat menyimpan instance dalam `Hibernated` keadaan untuk menghentikan instance tanpa menghapus konten memori (RAM) mereka. Ketika sebuah instans hibernasi, ini menandakan sistem operasi untuk menyimpan konten RAM Anda ke volume root Amazon EBS Anda. Ketika instance dimulai lagi, volume root dikembalikan ke keadaan sebelumnya dan konten RAM dimuat ulang. Saat instans dalam hibernasi, Anda hanya membayar untuk volume EBS, termasuk penyimpanan untuk konten RAM, dan alamat IP Elastis yang dilampirkan ke instans.

Menjaga keadaan dalam `Running` keadaan di dalam kolam hangat juga dimungkinkan, tetapi sangat tidak disarankan untuk menghindari biaya yang tidak perlu. Ketika instance dihentikan atau hibernasi, Anda menghemat biaya instans itu sendiri. Anda membayar untuk contoh hanya ketika mereka berjalan.

Kait siklus hidup

Anda menggunakan [kait siklus hidup](#) untuk menempatkan instance ke status tunggu sehingga Anda dapat melakukan tindakan kustom pada instance. Tindakan kustom dilakukan saat instance diluncurkan atau sebelum dihentikan.

Dalam konfigurasi kolam hangat, kait siklus hidup menunda instance agar tidak dihentikan atau dihibernasi dan dimasukkan ke dalam layanan selama acara scale-out hingga selesai diinisialisasi. Jika Anda menambahkan kolam hangat ke grup Auto Scaling tanpa pengait siklus hidup, instance yang membutuhkan waktu lama untuk menyelesaikan inisialisasi dapat dihentikan atau dihibernasi dan kemudian dimasukkan ke dalam layanan selama acara scale-out sebelum siap.

Kebijakan penggunaan kembali instans

Secara default, Auto Scaling Amazon EC2 menghentikan instans Anda saat grup Auto Scaling Anda masuk. Kemudian, ia meluncurkan instance baru ke kolam hangat untuk menggantikan instance yang dihentikan.

Jika Anda ingin mengembalikan instance ke kolam hangat, Anda dapat menentukan kebijakan penggunaan kembali instance. Ini memungkinkan Anda menggunakan kembali instance

yang sudah dikonfigurasi untuk melayani lalu lintas aplikasi. Untuk memastikan kolam hangat Anda tidak disediakan secara berlebihan, Auto Scaling Amazon EC2 dapat menghentikan instans di kolam hangat untuk mengurangi ukurannya saat lebih besar dari yang diperlukan berdasarkan pengaturannya. Saat mengakhiri instance di kolam hangat, ia menggunakan [kebijakan penghentian default](#) untuk memilih instance mana yang akan dihentikan terlebih dahulu.

Important

Jika Anda ingin hibernasi instance pada skala dan ada instance yang ada di grup Auto Scaling, instance tersebut harus memenuhi persyaratan misalnya hibernasi. Jika tidak, ketika contoh kembali ke kolam hangat, mereka akan mundur untuk dihentikan alih-alih hibernasi.

Note

Saat ini, Anda hanya dapat menentukan kebijakan penggunaan ulang instance dengan menggunakan AWS CLI atau SDK. Fitur ini tidak tersedia dari konsol.

Prasyarat

Sebelum Anda membuat kolam hangat untuk grup Auto Scaling, putuskan bagaimana Anda akan menggunakan kait siklus hidup untuk menginisialisasi instance baru dengan status awal yang sesuai.

Untuk melakukan tindakan kustom pada instance saat berada dalam status tunggu karena hook siklus hidup, Anda memiliki dua opsi:

- Untuk skenario sederhana di mana Anda ingin menjalankan perintah pada instance saat peluncuran, Anda dapat menyertakan skrip data pengguna saat membuat templat peluncuran atau konfigurasi peluncuran untuk grup Auto Scaling Anda. Skrip data pengguna hanyalah skrip shell normal atau direktif cloud-init yang dijalankan oleh [cloud-init](#) saat instance Anda dimulai. Skrip juga dapat mengontrol saat instance Anda bertransisi ke status berikutnya dengan menggunakan ID instance yang dijalkannya. Jika Anda belum melakukannya, perbarui skrip Anda untuk mengambil data ID instance dari metadata instance. Untuk informasi selengkapnya, lihat [Mengambil metadata instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

i Tip

Untuk menjalankan skrip data pengguna saat instance dimulai ulang, data pengguna harus dalam format multi-bagian MIME dan tentukan hal berikut di `#cloud-config` bagian data pengguna:

```
#cloud-config
cloud_final_modules:
  - [scripts-user, always]
```

- Untuk skenario lanjutan di mana Anda memerlukan layanan seperti AWS Lambda melakukan sesuatu saat instance memasuki atau meninggalkan kolam hangat, Anda dapat membuat kait siklus hidup untuk grup Auto Scaling dan mengonfigurasi layanan target untuk melakukan tindakan kustom berdasarkan pemberitahuan siklus hidup. Untuk informasi selengkapnya, lihat [Target notifikasi yang didukung](#).

Siapkan contoh untuk hibernasi

Untuk menyiapkan instans Auto Scaling agar menggunakan status *Hibernated* kumpulan, buat templat peluncuran baru atau konfigurasi peluncuran yang diatur dengan benar untuk mendukung hibernasi instans, seperti yang dijelaskan dalam [topik prasyarat Hibernasi di Panduan Pengguna Amazon EC2](#) untuk Instans Linux. Kemudian, kaitkan template peluncuran baru atau konfigurasi peluncuran dengan grup Auto Scaling dan mulai penyegaran instance untuk mengganti instance yang terkait dengan templat peluncuran atau konfigurasi peluncuran sebelumnya. Untuk informasi selengkapnya, lihat [Menggunakan penyegaran instans untuk memperbarui instance dalam grup Auto Scaling](#).

Perbarui contoh di kolam yang hangat

Untuk memperbarui instance di kolam hangat, Anda membuat templat peluncuran baru atau konfigurasi peluncuran dan mengaitkannya dengan grup Auto Scaling. Setiap instance baru diluncurkan menggunakan AMI baru dan pembaruan lain yang ditentukan dalam templat peluncuran atau konfigurasi peluncuran, tetapi instance yang ada tidak terpengaruh.

Untuk memaksa instance kumpulan hangat pengganti diluncurkan yang menggunakan templat peluncuran baru atau konfigurasi peluncuran, Anda dapat memulai penyegaran instans untuk melakukan pembaruan bergulir grup Anda. Penyegaran instance pertama menggantikan InService

instance. Kemudian menggantikan contoh di kolam hangat. Untuk informasi selengkapnya, lihat [Menggunakan penyegaran instans untuk memperbaiki instance dalam grup Auto Scaling](#).

Sumber daya terkait

Anda dapat mengunjungi [GitHubrepositori](#) kami untuk contoh kait siklus hidup untuk kolam hangat.

Batasan

- [Grup instance campuran](#) tidak didukung. Anda tidak dapat menambahkan kumpulan hangat ke grup Auto Scaling yang mengganti jenis instans yang ditentukan dalam template peluncuran atau yang disiapkan untuk meluncurkan Instans Spot.
- Auto Scaling Amazon EC2 dapat menempatkan instance dalam Stopped status Hibernated atau hanya jika memiliki volume Amazon EBS sebagai perangkat root. Instance yang menggunakan penyimpanan instance untuk perangkat root tidak dapat dihentikan atau dihibernasi.
- Auto Scaling Amazon EC2 hanya dapat menempatkan instans dalam status jika memenuhi semua persyaratan *Hibernated* yang tercantum dalam topik prasyarat [Hibernasi di Panduan Pengguna Amazon EC2 untuk Instans Linux](#).
- Jika kolam hangat Anda habis saat ada acara scale-out, instance akan diluncurkan langsung ke grup Auto Scaling (awal yang dingin). Anda juga bisa mengalami start dingin jika Availability Zone di luar kapasitas.
- Jika instance dalam kumpulan hangat mengalami masalah selama proses peluncuran, mencegahnya mencapai InService status, instance akan dianggap sebagai peluncuran yang gagal dan dihentikan. Ini berlaku terlepas dari penyebab yang mendasarinya, seperti kesalahan kapasitas yang tidak mencukupi atau faktor lainnya.
- Jika Anda mencoba menggunakan kolam hangat dengan grup node terkelola Amazon Elastic Kubernetes Service (Amazon EKS), instance yang masih diinisialisasi dapat mendaftar ke cluster Amazon EKS Anda. Akibatnya, cluster mungkin menjadwalkan pekerjaan pada sebuah instance karena sedang bersiap untuk dihentikan atau dihibernasi.
- Demikian juga, jika Anda mencoba menggunakan kolam hangat dengan kluster Amazon ECS, instance mungkin mendaftar ke cluster sebelum selesai diinisialisasi. Untuk mengatasi masalah ini, Anda harus mengonfigurasi templat peluncuran atau konfigurasi peluncuran yang menyertakan variabel konfigurasi agen khusus dalam data pengguna. Untuk informasi selengkapnya, lihat [Menggunakan kolam hangat untuk grup Auto Scaling Anda](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.

- Dukungan hibernasi untuk kolam hangat tersedia di semua iklan di Wilayah AWS mana Amazon EC2 Auto Scaling dan hibernasi tersedia, kecuali yang berikut:
 - Asia Pasifik (Hyderabad)
 - Asia Pasifik (Melbourne)
 - Kanada Barat (Calgary)
 - Wilayah Tiongkok (Beijing)
 - Wilayah China (Ningxia)
 - Eropa (Spanyol)
 - Israel (Tel Aviv)

Gunakan kait siklus hidup dengan kolam hangat

Instans di kolam hangat mempertahankan siklus hidup independennya sendiri untuk membantu Anda membuat tindakan kustom yang sesuai untuk setiap transisi. Siklus hidup ini dirancang untuk membantu Anda menjalankan tindakan dalam layanan target (misalnya, fungsi Lambda) saat instance masih diinisialisasi dan sebelum dimasukkan ke dalam layanan.

Note

Operasi API yang Anda gunakan untuk menambahkan dan mengelola kait siklus hidup dan tindakan siklus hidup lengkap tidak diubah. Hanya siklus hidup instance yang diubah.

Untuk informasi selengkapnya tentang menambahkan kait siklus hidup, lihat. [Tambahkan hook siklus aktif](#) Untuk informasi selengkapnya tentang menyelesaikan tindakan siklus hidup, lihat. [Selesaikan tindakan siklus hidup](#)

Untuk contoh memasuki kolam hangat, Anda mungkin memerlukan pengait siklus hidup karena salah satu alasan berikut:

- Anda ingin meluncurkan instans EC2 dari AMI yang membutuhkan waktu lama untuk menyelesaikan inisialisasi.
- Anda ingin menjalankan skrip data pengguna untuk mem-bootstrap instance EC2.

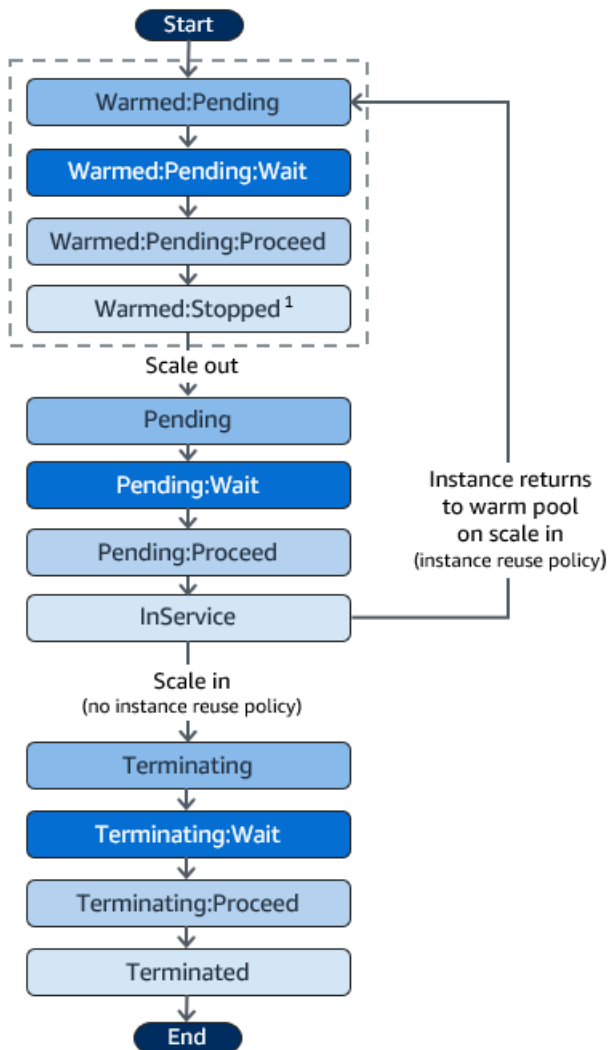
Untuk contoh meninggalkan kolam hangat, Anda mungkin memerlukan pengait siklus hidup karena salah satu alasan berikut:

- Anda dapat menggunakan beberapa waktu ekstra untuk menyiapkan instans EC2 untuk digunakan. Misalnya, Anda mungkin memiliki layanan yang harus dimulai saat instance dimulai ulang sebelum aplikasi Anda dapat bekerja dengan benar.
- Anda ingin mengisi data cache terlebih dahulu sehingga server baru tidak diluncurkan dengan cache kosong.
- Anda ingin mendaftarkan instance baru sebagai instance terkelola dengan layanan manajemen konfigurasi Anda.

Transisi status siklus hidup untuk instance di kolam hangat

Instance Auto Scaling dapat bertransisi melalui banyak status sebagai bagian dari siklus hidupnya.

Diagram berikut menunjukkan transisi antara status Auto Scaling saat Anda menggunakan kolam hangat:

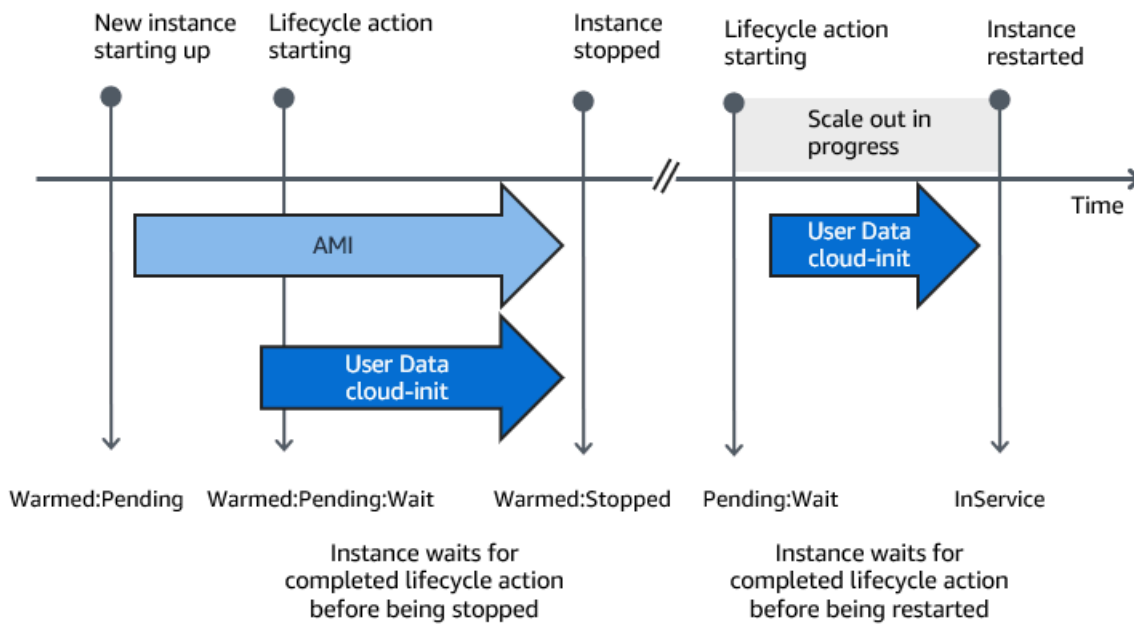


¹ Keadaan ini bervariasi berdasarkan pengaturan status kolam renang hangat. Jika status kumpulan diatur ke `Running`, maka status ini `Warmmed:Running` sebagai gantinya. Jika status kumpulan diatur ke `Hibernated`, maka status ini `Warmmed:Hibernated` sebagai gantinya.

Saat Anda menambahkan kait siklus hidup, pertimbangkan hal berikut:

- Saat kait siklus hidup dikonfigurasi untuk tindakan `autoscaling:EC2_INSTANCE_LAUNCHING` siklus hidup, instance yang baru diluncurkan akan berhenti sejenak untuk melakukan tindakan kustom saat mencapai `Warmmed:Pending:Wait` status, dan kemudian lagi saat instance dimulai ulang dan mencapai status. `Pending:Wait`
- Saat kait siklus hidup dikonfigurasi untuk tindakan `EC2_INSTANCE_TERMINATING` siklus hidup, instance penghentian akan berhenti untuk melakukan tindakan kustom saat mencapai status. `Terminating:Wait` Namun, jika Anda menetapkan kebijakan penggunaan kembali instance untuk mengembalikan instance ke kumpulan hangat pada skala alih-alih menghentikannya, maka instance yang kembali ke kumpulan hangat akan berhenti untuk melakukan tindakan kustom pada `Warmmed:Pending:Wait` status untuk tindakan siklus hidup. `EC2_INSTANCE_TERMINATING`
- Jika permintaan pada aplikasi Anda menghabiskan kolam hangat, Auto Scaling Amazon EC2 dapat meluncurkan instans langsung ke grup Auto Scaling selama grup belum mencapai kapasitas maksimumnya. Jika instans diluncurkan langsung ke grup, mereka hanya dijeda untuk melakukan tindakan kustom di negara bagian. `Pending:Wait`
- Untuk mengontrol berapa lama instans tetap dalam status tunggu sebelum transisi ke status berikutnya, konfigurasi tindakan kustom Anda untuk menggunakan `complete-lifecycle-action` perintah. Dengan kait siklus hidup, instance tetap dalam status tunggu hingga Anda memberi tahu Amazon EC2 Auto Scaling bahwa tindakan siklus hidup yang ditentukan telah selesai, atau hingga periode batas waktu berakhir (satu jam secara default).

Berikut ini merangkum alur untuk acara scale-out.



Saat instans mencapai status tunggu, Auto Scaling Amazon EC2 mengirimkan pemberitahuan. Contoh pemberitahuan ini tersedia di EventBridge bagian panduan ini. Untuk informasi selengkapnya, lihat [Acara dan pola contoh kolam hangat](#).

Target notifikasi yang didukung

Auto Scaling Amazon EC2 menyediakan dukungan untuk mendefinisikan salah satu hal berikut sebagai target notifikasi untuk notifikasi siklus hidup:

- EventBridge aturan
- Topik Amazon SNS
- Antrean Amazon SQS

⚠ Important

Ingat, jika Anda memiliki skrip data pengguna (cloud-init) di template peluncuran atau konfigurasi peluncuran yang mengonfigurasi instans Anda saat diluncurkan, Anda tidak perlu menerima pemberitahuan untuk melakukan tindakan khusus pada instance yang memulai atau memulai ulang.

Bagian berikut berisi tautan ke dokumentasi yang menjelaskan cara mengonfigurasi target notifikasi:

EventBridge aturan: Untuk menjalankan kode saat Amazon EC2 Auto Scaling menempatkan instance ke status tunggu, Anda dapat membuat EventBridge aturan dan menentukan fungsi Lambda sebagai targetnya. Untuk menjalankan fungsi Lambda yang berbeda berdasarkan pemberitahuan siklus hidup yang berbeda, Anda dapat membuat beberapa aturan dan mengaitkan setiap aturan dengan pola peristiwa tertentu dan fungsi Lambda. Untuk informasi selengkapnya, lihat [Buat EventBridge aturan untuk acara kolam hangat](#).

Topik Amazon SNS: Untuk menerima notifikasi saat instans dimasukkan ke dalam status tunggu, Anda membuat topik Amazon SNS dan kemudian mengatur pemfilteran pesan Amazon SNS untuk mengirimkan pemberitahuan siklus hidup secara berbeda berdasarkan atribut pesan. Untuk informasi selengkapnya, lihat [Terima notifikasi menggunakan Amazon SNS](#).

Antrian Amazon SQS: Untuk menyiapkan titik pengiriman pemberitahuan siklus hidup di mana konsumen yang relevan dapat mengambilnya dan memprosesnya, Anda dapat membuat antrian Amazon SQS dan konsumen antrian yang memproses pesan dari antrian SQS. Jika Anda ingin konsumen antrian memproses pemberitahuan siklus hidup secara berbeda berdasarkan atribut pesan, Anda juga harus mengatur konsumen antrian untuk mengurai pesan dan kemudian bertindak pada pesan ketika atribut tertentu cocok dengan nilai yang diinginkan. Untuk informasi selengkapnya, lihat [Menerima pemberitahuan menggunakan Amazon SQS](#).

Buat kolam hangat untuk grup Auto Scaling

Topik ini menjelaskan cara membuat kolam hangat untuk grup Auto Scaling Anda.

Important

Sebelum melanjutkan, lengkapi [prasyarat](#) untuk membuat kolam hangat dan konfirmasi bahwa Anda telah membuat kait siklus hidup untuk grup Auto Scaling Anda.

Buat kolam yang hangat

Gunakan prosedur berikut untuk membuat kolam hangat untuk grup Auto Scaling Anda.

Untuk membuat kolam hangat (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup yang sudah ada.

Panel split terbuka di bagian bawah halaman.

3. Pilih tab Manajemen instans.
4. Di bawah Kolam hangat, pilih Buat kolam hangat.
5. Untuk mengkonfigurasi kolam hangat, lakukan hal berikut:
 - a. Untuk status instance kolam hangat, pilih status mana yang ingin Anda transisikan instance saat mereka memasuki kolam hangat. Nilai default-nya Stopped.
 - b. Untuk ukuran kolam hangat Minimum, masukkan jumlah minimum instance untuk disimpan di kolam hangat.
 - c. Untuk penggunaan kembali Instance, pilih kotak centang Gunakan kembali pada skala untuk mengizinkan instance dalam grup Auto Scaling kembali ke kolam hangat pada skala masuk.
 - d. Untuk ukuran kolam hangat, pilih salah satu opsi yang tersedia:
 - Spesifikasi default: Ukuran kolam hangat ditentukan oleh perbedaan antara kapasitas maksimum dan yang diinginkan dari grup Auto Scaling. Opsi ini merampingkan manajemen kolam hangat. Setelah Anda membuat kolam hangat, ukurannya dapat dengan mudah diperbarui hanya dengan menyesuaikan kapasitas maksimum grup.
 - Spesifikasi khusus: Ukuran kolam hangat ditentukan oleh perbedaan antara nilai khusus dan kapasitas yang diinginkan dari grup Auto Scaling. Opsi ini memberi Anda fleksibilitas untuk mengelola ukuran kolam hangat Anda secara independen dari kapasitas maksimum grup.
6. Lihat bagian Perkiraan ukuran kolam hangat berdasarkan pengaturan saat ini untuk mengonfirmasi bagaimana spesifikasi default atau khusus berlaku untuk ukuran kolam hangat. Ingat, ukuran kolam hangat tergantung pada kapasitas yang diinginkan dari grup Auto Scaling, yang akan berubah jika skala grup.
7. Pilih Buat.

Hapus kolam hangat

Saat Anda tidak lagi membutuhkan kolam hangat, gunakan prosedur berikut untuk menghapusnya.

Untuk menghapus kolam hangat Anda (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.

2. Pilih kotak centang di samping grup yang sudah ada.

Panel split terbuka di bagian bawah halaman.

3. Pilih tab Manajemen instans.
4. Untuk kolam hangat, pilih Tindakan, Hapus.
5. Saat diminta konfirmasi, pilih Hapus.

Lihat status pemeriksaan kesehatan dan alasan kegagalan pemeriksaan kesehatan

Pemeriksaan kesehatan memungkinkan Amazon EC2 Auto Scaling untuk menentukan kapan instans tidak sehat dan harus dihentikan. Untuk contoh kolam hangat yang disimpan dalam Stopped keadaan, ia menggunakan pengetahuan bahwa Amazon EBS memiliki ketersediaan instans untuk mengidentifikasi Stopped contoh yang tidak sehat. Ini dilakukan dengan memanggil DescribeVolumeStatus API untuk menentukan status volume EBS yang dilampirkan ke instance. Untuk instance kolam hangat yang disimpan dalam Running keadaan, itu bergantung pada pemeriksaan status EC2 untuk menentukan kesehatan instance. Meskipun tidak ada masa tenggang pemeriksaan kesehatan untuk instans kolam hangat, Auto Scaling Amazon EC2 tidak mulai memeriksa kesehatan instans hingga pengait siklus hidup selesai.

Ketika sebuah instans ditemukan tidak sehat, Auto Scaling Amazon EC2 secara otomatis menghapus instans yang tidak sehat dan membuat yang baru untuk menggantikannya. Contoh biasanya dihentikan dalam beberapa menit setelah gagal dalam pemeriksaan kesehatan mereka. Untuk informasi selengkapnya, lihat [Lihat alasan kegagalan pemeriksaan kesehatan](#).

Pemeriksaan kesehatan khusus juga didukung. Ini dapat membantu jika Anda memiliki sistem pemeriksaan kesehatan sendiri yang dapat mendeteksi kesehatan instans dan mengirimkan informasi ini ke Amazon EC2 Auto Scaling. Untuk informasi selengkapnya, lihat [Pemeriksaan kesehatan khusus](#).

Di konsol Auto Scaling Amazon EC2, Anda dapat melihat status (sehat atau tidak sehat) instans kolam hangat Anda. Anda juga dapat melihat status kesehatan mereka menggunakan AWS CLI atau salah satu SDK.

Untuk melihat status instance kolam hangat Anda (konsol)


1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.

2. Pilih kotak centang di samping grup Auto Scaling.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Pada tab Manajemen instans, di bawah instance kumpulan Warm, kolom Siklus Hidup berisi status instance Anda.

Kolom status Kesehatan menunjukkan penilaian yang dibuat oleh Amazon EC2 Auto Scaling tentang kesehatan instans.

 Note

Contoh baru mulai sehat. Sampai kait siklus hidup selesai, kesehatan instans tidak diperiksa.

Untuk melihat alasan kegagalan pemeriksaan kesehatan (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Pada tab Aktivitas, di Riwayat aktivitas, kolom Status menunjukkan apakah grup Auto Scaling Anda telah berhasil meluncurkan atau menghentikan instance.

Jika menghentikan kasus yang tidak sehat, kolom Penyebab menunjukkan tanggal dan waktu penghentian dan alasan kegagalan pemeriksaan kesehatan. Misalnya, "Pada 2021-04-01T 21:48:35 Z sebuah instance dikeluarkan dari layanan sebagai tanggapan atas kegagalan pemeriksaan kesehatan volume EBS".

Untuk melihat status instance kolam hangat Anda ()AWS CLI

Lihat kolam hangat untuk grup Auto Scaling dengan menggunakan perintah berikut [describe-warm-pool](#).

```
aws autoscaling describe-warm-pool --auto-scaling-group-name my-asg
```

Contoh keluaran

```
{
  "WarmPoolConfiguration": {
    "MinSize": 0,
    "PoolState": "Stopped"
  },
  "Instances": [
    {
      "InstanceId": "i-0b5e5e7521cfaa46c",
      "InstanceType": "t2.micro",
      "AvailabilityZone": "us-west-2a",
      "LifecycleState": "Warmed:Stopped",
      "HealthStatus": "Healthy",
      "LaunchTemplate": {
        "LaunchTemplateId": "lt-08c4cd42f320d5dcd",
        "LaunchTemplateName": "my-template-for-auto-scaling",
        "Version": "1"
      }
    },
    {
      "InstanceId": "i-0e21af9dcfb7aa6bf",
      "InstanceType": "t2.micro",
      "AvailabilityZone": "us-west-2a",
      "LifecycleState": "Warmed:Stopped",
      "HealthStatus": "Healthy",
      "LaunchTemplate": {
        "LaunchTemplateId": "lt-08c4cd42f320d5dcd",
        "LaunchTemplateName": "my-template-for-auto-scaling",
        "Version": "1"
      }
    }
  ]
}
```

Untuk melihat alasan kegagalan pemeriksaan kesehatan (AWS CLI)

Gunakan perintah berikut [describe-scaling-activities](#).

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg
```

Berikut ini adalah contoh respons, yang `Description` menunjukkan bahwa grup Auto Scaling Anda telah menghentikan instance dan `Cause` menunjukkan alasan kegagalan pemeriksaan kesehatan.

Aktivitas penskalaan diurutkan berdasarkan waktu mulai. Aktivitas masih dalam proses dijelaskan terlebih dahulu.

```
{
  "Activities": [
    {
      "ActivityId": "4c65e23d-a35a-4e7d-b6e4-2eaa8753dc12",
      "AutoScalingGroupName": "my-asg",
      "Description": "Terminating EC2 instance: i-04925c838b6438f14",
      "Cause": "At 2021-04-01T21:48:35Z an instance was taken out of service in response to EBS volume health check failure.",
      "StartTime": "2021-04-01T21:48:35.859Z",
      "EndTime": "2021-04-01T21:49:18Z",
      "StatusCode": "Successful",
      "Progress": 100,
      "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-west-2a\"...}",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:283179a2-f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
    },
    ...
  ]
}
```

Contoh untuk membuat dan mengelola kolam hangat dengan AWS CLI

Anda dapat membuat dan mengelola kolam hangat menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau SDK.

Contoh berikut menunjukkan cara membuat dan mengelola kolam hangat menggunakan AWS CLI.

Daftar Isi

- [Contoh 1: Simpan instance di negara bagian Stopped](#)
- [Contoh 2: Simpan instance di negara bagian Running](#)
- [Contoh 3: Simpan instance di negara bagian Hibernated](#)
- [Contoh 4: Kembalikan instance ke kolam hangat saat menskalakan](#)
- [Contoh 5: Tentukan jumlah minimum instance di kolam hangat](#)
- [Contoh 6: Tentukan ukuran kolam hangat menggunakan spesifikasi khusus](#)

- [Contoh 7: Tentukan ukuran kolam hangat mutlak](#)
- [Contoh 8: Hapus kolam hangat](#)

Contoh 1: Simpan instance di negara bagian **Stopped**

[put-warm-pool](#) Contoh berikut membuat kolam hangat yang menjaga instance dalam Stopped keadaan.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /  
--pool-state Stopped
```

Contoh 2: Simpan instance di negara bagian **Running**

[put-warm-pool](#) Contoh berikut membuat kolam hangat yang menyimpan instance dalam Running keadaan, bukan Stopped status.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /  
--pool-state Running
```

Contoh 3: Simpan instance di negara bagian **Hibernated**

[put-warm-pool](#) Contoh berikut membuat kolam hangat yang menyimpan instance dalam Hibernated keadaan, bukan Stopped status. Ini memungkinkan Anda menghentikan instance tanpa menghapus konten memori (RAM) mereka.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /  
--pool-state Hibernated
```

Contoh 4: Kembalikan instance ke kolam hangat saat menskalakan

[put-warm-pool](#) Contoh berikut membuat kolam hangat yang menyimpan instance dalam Stopped keadaan dan menyertakan `--instance-reuse-policy` opsi. Nilai kebijakan penggunaan kembali instans `'{"ReuseOnScaleIn": true}'` memberi tahu Auto Scaling Amazon EC2 untuk mengembalikan instans ke kolam hangat saat grup Auto Scaling Anda masuk.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /  
--pool-state Stopped --instance-reuse-policy '{"ReuseOnScaleIn": true}'
```

Contoh 5: Tentukan jumlah minimum instance di kolam hangat

[put-warm-pool](#) Contoh berikut membuat kolam hangat yang mempertahankan minimal 4 instance, sehingga setidaknya ada 4 contoh yang tersedia untuk menangani lonjakan lalu lintas.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /  
  --pool-state Stopped --min-size 4
```

Contoh 6: Tentukan ukuran kolam hangat menggunakan spesifikasi khusus

Secara default, Amazon EC2 Auto Scaling mengelola ukuran kolam hangat Anda sebagai perbedaan antara kapasitas maksimum dan yang diinginkan dari grup Auto Scaling. Namun, Anda dapat mengatur ukuran kolam hangat secara independen dari kapasitas maksimum grup dengan menggunakan `--max-group-prepared-capacity` opsi.

[put-warm-pool](#) Contoh berikut membuat kolam hangat dan menetapkan jumlah maksimum instance yang dapat eksis secara bersamaan di kolam hangat dan grup Auto Scaling. Jika grup memiliki kapasitas 800 yang diinginkan, kolam hangat awalnya akan memiliki ukuran 100 saat diinisialisasi setelah menjalankan perintah ini.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /  
  --pool-state Stopped --max-group-prepared-capacity 900
```

Untuk mempertahankan jumlah minimum instance di kolam hangat, sertakan `--min-size` opsi dengan perintah, sebagai berikut.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /  
  --pool-state Stopped --max-group-prepared-capacity 900 --min-size 25
```

Contoh 7: Tentukan ukuran kolam hangat mutlak

Jika Anda menetapkan nilai yang sama untuk `--min-size` opsi `--max-group-prepared-capacity` dan, kolam hangat memiliki ukuran absolut. [put-warm-pool](#) Contoh berikut menciptakan kolam hangat yang mempertahankan ukuran kolam hangat konstan 10 kali.

```
aws autoscaling put-warm-pool --auto-scaling-group-name my-asg /  
  --pool-state Stopped --min-size 10 --max-group-prepared-capacity 10
```

Contoh 8: Hapus kolam hangat

Gunakan [delete-warm-pool](#) perintah berikut untuk menghapus kolam hangat.

```
aws autoscaling delete-warm-pool --auto-scaling-group-name my-asg
```

Jika ada instance di kolam hangat, atau jika aktivitas penskalaan sedang berlangsung, gunakan [delete-warm-pool](#) perintah dengan opsi. `--force-delete` Opsi ini juga menghentikan instans Amazon EC2 dan tindakan siklus hidup yang luar biasa.

```
aws autoscaling delete-warm-pool --auto-scaling-group-name my-asg --force-delete
```

Lepaskan atau lampirkan instance

Anda dapat melepaskan instance dari grup Auto Scaling Anda. Setelah sebuah instance terlepas, instance tersebut menjadi independen dan dapat dikelola sendiri atau dilampirkan ke grup Auto Scaling yang berbeda, terpisah dari grup asli miliknya. Ini dapat berguna, misalnya, ketika Anda ingin melakukan pengujian menggunakan instance yang ada yang sudah menjalankan aplikasi Anda.

Topik ini memberikan petunjuk tentang cara melepaskan dan melampirkan instance. Saat melampirkan instance, Anda juga dapat menggunakan instance yang ada daripada yang terpisah.

Alih-alih melepaskan dan melampirkan kembali instance ke grup yang sama, sebaiknya gunakan prosedur siaga untuk menghapus instance sementara dari grup. Untuk informasi selengkapnya, lihat [Hapus instans sementara dari grup Auto Scaling](#).

Daftar Isi

- [Pertimbangan untuk melepaskan contoh](#)
- [Pertimbangan untuk melampirkan instance](#)
- [Pindahkan instance ke grup yang berbeda menggunakan detach and attach](#)

Pertimbangan untuk melepaskan contoh

Saat Anda melepaskan instance, ingatlah poin-poin ini:

- Anda dapat melepaskan instance hanya ketika berada di InService negara bagian.

- Setelah Anda melepaskan sebuah instance, instance terus berjalan dan menimbulkan biaya. Untuk menghindari biaya yang tidak perlu, pastikan untuk memasang kembali atau menghentikan instance yang terlepas saat tidak lagi diperlukan.
- Anda dapat memilih untuk mengurangi kapasitas yang diinginkan dengan jumlah contoh yang Anda lepaskan. Jika Anda memilih untuk tidak mengurangi kapasitas, Amazon EC2 Auto Scaling meluncurkan instans baru untuk mengganti instans terpisah untuk mempertahankan kapasitas yang diinginkan.
- Jika jumlah instance yang Anda lepaskan akan membawa grup Auto Scaling di bawah kapasitas minimumnya, Anda harus mengurangi kapasitas minimum.
- Jika Anda melepaskan beberapa instance dari Availability Zone yang sama tanpa mengurangi kapasitas yang diinginkan, grup akan menyeimbangkan dirinya sendiri kecuali Anda menangguhkan prosesnya. AZRebalance Untuk informasi selengkapnya, lihat [Menangguhkan dan melanjutkan proses Auto Scaling Amazon EC2](#).
- Jika Anda melepaskan instance dari grup Auto Scaling yang memiliki grup target penyeimbang beban terlampir atau Classic Load Balancer, instance tersebut akan dideregistrasi dari penyeimbang beban. Jika pengurusan sambungan (penundaan deregistrasi) diaktifkan untuk penyeimbang beban Anda, Auto Scaling Amazon EC2 akan menunggu permintaan dalam penerbangan selesai.

Note

Jika Anda melepaskan contoh yang ada di Standby negara bagian, berhati-hatilah. Mencoba melepaskan instance setelah memasukkannya ke Standby status dapat menyebabkan instance lain berakhir secara tidak terduga.

Pertimbangan untuk melampirkan instance

Perhatikan hal berikut saat melampirkan instance:

- Auto Scaling Amazon EC2 memperlakukan instans terlampir sama seperti instans yang diluncurkan oleh grup itu sendiri. Ini berarti bahwa instance terlampir dapat dihentikan selama peristiwa scale-in jika dipilih.
- Ketika Anda melampirkan instance, kapasitas yang diinginkan grup akan meningkat berdasarkan jumlah instance yang dilampirkan. Jika kapasitas yang diinginkan setelah menambahkan instance baru melebihi ukuran maksimum grup, permintaan untuk melampirkan lebih banyak instance gagal.

- Jika Anda menambahkan instans ke grup yang menyebabkan distribusi tidak merata di seluruh Availability Zone, Auto Scaling Amazon EC2 menyeimbangkan kembali grup untuk membuat kembali distribusi yang merata kecuali Anda menanggukannya. AZRebalance Untuk informasi selengkapnya, lihat [Menanggukkan dan melanjutkan proses Auto Scaling Amazon EC2](#).
- Jika Anda melampirkan instance ke grup Auto Scaling yang memiliki grup target penyeimbang beban terlampir atau Classic Load Balancer, instance tersebut terdaftar dengan penyeimbang beban.

Untuk contoh yang akan dilampirkan, itu harus memenuhi kriteria berikut:

- Instans ada di running negara bagian dengan Amazon EC2.
- AMI yang digunakan untuk meluncurkan instance tetap harus ada.
- Instance ini bukanlah anggota dari grup Auto Scaling lainnya.
- Instance diluncurkan ke salah satu Availability Zone yang ditentukan dalam grup Auto Scaling.
- Jika grup Auto Scaling memiliki grup target penyeimbang beban terpasang atau Classic Load Balancer, instance dan penyeimbang beban keduanya harus berada dalam VPC yang sama.

Pindahkan instance ke grup yang berbeda menggunakan detach and attach

Gunakan salah satu prosedur berikut untuk melepaskan instance dari grup Auto Scaling Anda dan lampirkan ke grup Auto Scaling yang berbeda.

Untuk membuat grup Auto Scaling baru dari instance terpisah, lihat [Buat grup Auto Scaling menggunakan parameter dari instance yang ada](#) (tidak disarankan, membuat konfigurasi peluncuran).

Console

Untuk melepaskan instance dari grup Auto Scaling

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Manajemen Instance, dalam Instance, pilih satu instance dan pilih Tindakan, Pelepasan.

4. Dalam kotak dialog Lepaskan instance, simpan kotak centang Ganti instance yang dipilih untuk meluncurkan instance pengganti. Kosongkan kotak centang untuk mengurangi kapasitas yang diinginkan.
5. Saat diminta konfirmasi, ketik **detach** untuk mengonfirmasi penghapusan instance yang ditentukan dari grup Auto Scaling, lalu pilih Lepaskan instance.

Anda sekarang dapat melampirkan instance ke grup Auto Scaling yang berbeda.

Untuk melampirkan instance ke grup Auto Scaling

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. (Opsional) Pada panel navigasi, di bawah Auto Scaling, pilih Grup Auto Scaling. Pilih grup Auto Scaling dan pastikan ukuran maksimum grup Auto Scaling cukup besar sehingga Anda dapat menambahkan instance lain. Jika tidak, pada tab Rincian, tingkatkan kapasitas maksimum.
3. Pada panel navigasi, di bawah Instans, pilih Instans, lalu pilih instance.
4. Pilih Tindakan, Pengaturan instance, Lampirkan ke Grup Auto Scaling.
5. Pada halaman Melampirkan ke grup Auto Scaling, untuk Grup Auto Scaling, pilih grup Auto Scaling, lalu pilih Lampirkan.
6. Jika instance tidak memenuhi kriteria, Anda mendapatkan pesan kesalahan dengan detail. Misalnya, instance tersebut mungkin tidak berada dalam Zona Ketersediaan yang sama dengan grup Auto Scaling. Pilih Tutup dan coba lagi dengan grup Auto Scaling yang memenuhi kriteria.

AWS CLI

Untuk melepaskan dan melampirkan sebuah instance, gunakan perintah contoh berikut. Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri.

Untuk melepaskan instance dari grup Auto Scaling

1. Untuk menggambarkan contoh saat ini, gunakan [describe-auto-scaling-instances](#) perintah berikut.

```
aws autoscaling describe-auto-scaling-instances \  
  --query 'AutoScalingInstances[?AutoScalingGroupName==`my-asg`]'
```

Contoh berikut menunjukkan output yang dihasilkan ketika Anda menjalankan perintah ini.

Catat ID instance yang ingin Anda hapus dari grup. Anda memerlukan ID ini di langkah berikutnya.

```
{
  "AutoScalingInstances": [
    {
      "ProtectedFromScaleIn": false,
      "AvailabilityZone": "us-west-2a",
      "LaunchTemplate": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "1",
        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
      },
      "InstanceId": "i-05b4f7d5be44822a6",
      "InstanceType": "t3.micro",
      "AutoScalingGroupName": "my-asg",
      "HealthStatus": "HEALTHY",
      "LifecycleState": "InService"
    },
    {
      "ProtectedFromScaleIn": false,
      "AvailabilityZone": "us-west-2a",
      "LaunchTemplate": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "1",
        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
      },
      "InstanceId": "i-0c20ac468fa3049e8",
      "InstanceType": "t3.micro",
      "AutoScalingGroupName": "my-asg",
      "HealthStatus": "HEALTHY",
      "LifecycleState": "InService"
    },
    {
      "ProtectedFromScaleIn": false,
      "AvailabilityZone": "us-west-2a",
      "LaunchTemplate": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "1",
        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
      }
    }
  ]
}
```

```

    },
    "InstanceId": "i-0787762faf1c28619",
    "InstanceType": "t3.micro",
    "AutoScalingGroupName": "my-asg",
    "HealthStatus": "HEALTHY",
    "LifecycleState": "InService"
  },
  {
    "ProtectedFromScaleIn": false,
    "AvailabilityZone": "us-west-2a",
    "LaunchTemplate": {
      "LaunchTemplateName": "my-launch-template",
      "Version": "1",
      "LaunchTemplateId": "lt-050555ad16a3f9c7f"
    },
    "InstanceId": "i-0f280a4c58d319a8a",
    "InstanceType": "t3.micro",
    "AutoScalingGroupName": "my-asg",
    "HealthStatus": "HEALTHY",
    "LifecycleState": "InService"
  }
]
}

```

2. Untuk melepaskan instance tanpa mengurangi kapasitas yang diinginkan, gunakan perintah [detach-instance](#) berikut.

```
aws autoscaling detach-instances --instance-ids i-05b4f7d5be44822a6 \
  --auto-scaling-group-name my-asg
```

Untuk melepaskan instance dan mengurangi kapasitas yang diinginkan, sertakan opsi. `--should-decrement-desired-capacity`

```
aws autoscaling detach-instances --instance-ids i-05b4f7d5be44822a6 \
  --auto-scaling-group-name my-asg --should-decrement-desired-capacity
```

Anda sekarang dapat melampirkan instance ke grup Auto Scaling yang berbeda.

Untuk melampirkan instance ke grup Auto Scaling

1. Untuk melampirkan instance ke grup Auto Scaling yang berbeda, gunakan perintah [attach-instance](#) berikut.

```
aws autoscaling attach-instances --instance-ids i-05b4f7d5be44822a6 --auto-scaling-group-name my-asg-for-testing
```

2. Untuk memverifikasi ukuran grup Auto Scaling setelah melampirkan instance, gunakan perintah berikut. [describe-auto-scaling-groups](#)

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names my-asg-for-testing
```

Contoh respons berikut menunjukkan bahwa grup memiliki dua instance yang berjalan, salah satunya adalah instance yang Anda lampirkan.

```
{
  "AutoScalingGroups": [
    {
      "AutoScalingGroupName": "my-asg-for-testing",
      "AutoScalingGroupARN": "arn",
      "LaunchTemplate": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "2",
        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
      },
      "MinSize": 1,
      "MaxSize": 5,
      "DesiredCapacity": 2,
      ...
      "Instances": [
        {
          "ProtectedFromScaleIn": false,
          "AvailabilityZone": "us-west-2a",
          "LaunchTemplate": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "1",
            "LaunchTemplateId": "lt-050555ad16a3f9c7f"
          },
          "InstanceId": "i-05b4f7d5be44822a6",
          "InstanceType": "t3.micro",

```

```
        "HealthStatus": "Healthy",
        "LifecycleState": "InService"
    },
    {
        "ProtectedFromScaleIn": false,
        "AvailabilityZone": "us-west-2a",
        "LaunchTemplate": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "2",
            "LaunchTemplateId": "lt-050555ad16a3f9c7f"
        },
        "InstanceId": "i-00dcdfffd5175890",
        "InstanceType": "t3.micro",
        "HealthStatus": "Healthy",
        "LifecycleState": "InService"
    }
],
...
}
]
```

Hapus instans sementara dari grup Auto Scaling

Anda dapat menempatkan instance dalam status `InService` ke dalam status `Standby`, memperbarui, atau memecahkan masalah instance, kemudian mengembalikan permintaan ke layanan. Instans yang siaga masih merupakan bagian dari grup Auto Scaling, tetapi mereka tidak secara aktif menangani lalu lintas penyeimbang beban.

Fitur ini membantu Anda menghentikan dan memulai instans atau mem-boot ulang instans tanpa khawatir tentang Auto Scaling Amazon EC2 yang menghentikan instans sebagai bagian dari pemeriksaan kesehatannya atau selama peristiwa penskalaan.

Misalnya, Anda dapat mengubah Amazon Machine Image (AMI) untuk grup Auto Scaling kapan saja dengan mengubah template peluncuran atau konfigurasi peluncuran. Instance berikutnya yang diluncurkan grup Auto Scaling menggunakan AMI ini. Namun, grup Auto Scaling tidak memperbarui instance yang saat ini dalam layanan. Anda dapat menghentikan instans ini dan membiarkan Auto Scaling Amazon EC2 menggantinya, atau menggunakan fitur penyegaran instans untuk menghentikan dan mengganti instans. Atau, Anda dapat meletakkan instance di mode siaga, memperbarui perangkat lunak, lalu menjalankan instance kembali.

Melepaskan instance dari grup Auto Scaling mirip dengan menempatkan instance dalam keadaan siaga. Melepaskan instance mungkin berguna jika Anda ingin melampirkannya ke grup lain atau mengelola instance seperti instans EC2 mandiri dan mungkin menghentikannya. Untuk informasi selengkapnya, lihat [Lepaskan atau lampirkan instance](#).

Daftar Isi

- [Cara kerja status siaga](#)
- [Pertimbangan](#)
- [Status kesehatan suatu instance dalam status siaga](#)
- [Hapus sementara instance dengan menyetelnya ke standby](#)

Cara kerja status siaga

Status siaga berfungsi sebagai berikut untuk membantu Anda menghapus proses dari grup Auto Scaling Anda:

1. Anda menempatkan instance ke status siaga. Dalam status ini, instance tetap berlangsung hingga Anda keluar dari status siaga.
2. Jika ada grup target penyeimbang beban atau Classic Load Balancer yang terpasang pada grup Auto Scaling Anda, instance tersebut dideregistrasi dari penyeimbang beban. Jika pengurusan sambungan diaktifkan untuk penyeimbang beban, Elastic Load Balancing menunggu 300 detik secara default sebelum menyelesaikan proses deregistrasi, yang membantu menyelesaikan permintaan dalam penerbangan.
3. Anda dapat memperbarui atau memecahkan masalah instance.
4. Anda mengembalikan permintaan ke layanan dengan keluar dari status siaga.
5. Jika ada grup target penyeimbang beban atau Classic Load Balancer yang terpasang pada grup Auto Scaling Anda, instans akan terdaftar di load balancer.

Untuk informasi selengkapnya tentang siklus hidup instance dalam grup Auto Scaling, lihat [Siklus aktif instance Amazon EC2 Auto Scaling](#)

Pertimbangan

Berikut ini adalah pertimbangan saat memindahkan instance masuk dan keluar dari status siaga:

- Saat Anda meletakkan instance dalam keadaan siaga, Anda dapat mengurangi kapasitas yang diinginkan melalui operasi ini, atau menyimpannya dengan nilai yang sama.
- Jika Anda memilih untuk tidak mengurangi kapasitas yang diinginkan dari grup Auto Scaling, Amazon EC2 Auto Scaling meluncurkan instans untuk menggantikan yang dalam keadaan siaga. Tujuannya adalah membantu Anda menjaga kapasitas untuk aplikasi Anda saat satu atau beberapa instance berada dalam status siaga.
- Jika Anda memilih untuk mengurangi kapasitas yang diinginkan dari grup Auto Scaling, ini mencegah peluncuran instance untuk menggantikan yang dalam keadaan siaga.
- Setelah Anda mengembalikan instans dalam layanan, kapasitas yang diinginkan akan bertambah untuk mencerminkan berapa banyak instans dalam grup Auto Scaling.
- Untuk melakukan kenaikan (dan penurunan), kapasitas baru yang diinginkan harus antara ukuran kelompok minimum dan maksimum. Jika tidak, operasi gagal.
- Jika sewaktu-waktu setelah menempatkan instance dalam keadaan siaga, atau mengembalikan instans ke layanan dengan keluar dari status siaga, grup Auto Scaling Anda ditemukan tidak seimbang antara Availability Zone, Auto Scaling Amazon EC2 mengkompensasi dengan menyeimbangkan kembali Availability Zone kecuali Anda menanggung prosesnya. AZRebalance Untuk informasi selengkapnya, lihat [Menangguhkan dan melanjutkan proses Auto Scaling Amazon EC2](#).
- Anda ditagih untuk instance yang berada dalam status siaga.

Status kesehatan suatu instance dalam status siaga

Amazon EC2 Auto Scaling tidak melakukan pemeriksaan kesehatan pada instance yang berada dalam status siaga. Meskipun terdapat dalam status siaga, status kesehatan mencerminkan status yang dimiliki sebelum Anda meletakkannya dalam status siaga. Amazon EC2 Auto Scaling tidak melakukan pemeriksaan kesehatan pada instance hingga Anda memasangnya kembali sebagai layanan.

Misalnya, jika Anda meletakkan instance yang sehat di standby dan kemudian menghentikannya, Amazon EC2 Auto Scaling terus melaporkan instance tersebut sebagai sehat. Jika Anda mencoba untuk menempatkan penghentian yang sedang dalam status siaga kembali ke layanan, Amazon EC2 Auto Scaling melakukan pemeriksaan kesehatan pada instance tersebut, menentukan bahwa layanan ini akan berakhir dan tidak sehat, dan meluncurkan penggantian instance. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

Hapus sementara instance dengan menyetelnya ke standby

Gunakan salah satu prosedur berikut untuk mengambil instance di luar layanan sementara dengan menemukannya ke status siaga.

Console

Untuk menghapus temporer instance

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Manajemen instance, di Instance, pilih satu instance.
4. Pilih Tindakan, Atur ke Siaga.
5. Dalam kotak dialog Set to Standby, simpan kotak centang Ganti instance yang dipilih untuk meluncurkan instance pengganti. Kosongkan kotak centang untuk mengurangi kapasitas yang diinginkan.
6. Saat diminta konfirmasi, ketik **standby** untuk mengonfirmasi menempatkan instance yang ditentukan ke Standby status, lalu pilih Setel ke Siaga.
7. Anda dapat memperbarui atau memecahkan masalah instance Anda sesuai kebutuhan. Setelah selesai, lanjutkan dengan langkah berikutnya untuk mengembalikan permintaan ke layanan.
8. Pilih instance, pilih Tindakan, Setel ke InService. Dalam Setel ke kotak InService dialog, pilih Setel ke InService.

AWS CLI

Untuk menghapus instans sementara dari grup Auto Scaling Anda, gunakan perintah contoh berikut. Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri.

Untuk menghapus temporer instance

1. Gunakan [describe-auto-scaling-instances](#) perintah berikut untuk mengidentifikasi instance untuk memperbarui.


```
aws autoscaling describe-auto-scaling-instances \
  --query 'AutoScalingInstances[?AutoScalingGroupName==`my-asg`]'
```

Contoh berikut menunjukkan output yang dihasilkan ketika Anda menjalankan perintah ini.

Catat ID instance yang ingin Anda hapus dari grup. Anda memerlukan ID ini di langkah berikutnya.

```
{
  "AutoScalingInstances": [
    {
      "ProtectedFromScaleIn": false,
      "AvailabilityZone": "us-west-2a",
      "LaunchTemplate": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "1",
        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
      },
      "InstanceId": "i-05b4f7d5be44822a6",
      "InstanceType": "t3.micro",
      "AutoScalingGroupName": "my-asg",
      "HealthStatus": "HEALTHY",
      "LifecycleState": "InService"
    },
    ...
  ]
}
```

2. Pindahkan instance ke Standby status menggunakan perintah [enter-standby](#) berikut. Opsi `--should-decrement-desired-capacity` mengurangi kapasitas yang diinginkan sehingga grup Auto Scaling tidak meluncurkan instance penggantian.

```
aws autoscaling enter-standby --instance-ids i-05b4f7d5be44822a6 \
  --auto-scaling-group-name my-asg --should-decrement-desired-capacity
```

Berikut ini adalah contoh respons.

```
{
  "Activities": [
    {
```

```

    "ActivityId": "3b1839fe-24b0-40d9-80ae-bcd883c2be32",
    "AutoScalingGroupName": "my-asg",
    "Description": "Moving EC2 instance to Standby:
i-05b4f7d5be44822a6",
    "Cause": "At 2023-12-15T21:31:26Z instance i-05b4f7d5be44822a6 was
moved to standby
    in response to a user request, shrinking the capacity from 4 to
3.",
    "StartTime": "2023-12-15T21:31:26.150Z",
    "StatusCode": "InProgress",
    "Progress": 50,
    "Details": "{\"Subnet ID\": \"subnet-c934b782\", \"Availability Zone
\": \"us-west-2a\"}"
  }
]
}

```

3. (Opsional) Verifikasi bahwa instance Standby menggunakan [describe-auto-scaling-instances](#) perintah berikut.

```
aws autoscaling describe-auto-scaling-instances --instance-
ids i-05b4f7d5be44822a6
```

Berikut ini adalah contoh respons. Perhatikan bahwa status instance sekarang adalah Standby.

```

{
  "AutoScalingInstances": [
    {
      "ProtectedFromScaleIn": false,
      "AvailabilityZone": "us-west-2a",
      "LaunchTemplate": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "1",
        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
      },
      "InstanceId": "i-05b4f7d5be44822a6",
      "InstanceType": "t3.micro",
      "AutoScalingGroupName": "my-asg",
      "HealthStatus": "HEALTHY",
      "LifecycleState": "Standby"
    },
  ],
}

```

```
    ...
  ]
}
```

4. Anda dapat memperbarui atau memecahkan masalah instance Anda sesuai kebutuhan. Setelah selesai, lanjutkan dengan langkah berikutnya untuk mengembalikan permintaan ke layanan.
5. Masukkan kembali instance dalam layanan menggunakan perintah [exit-standby](#) berikut.

```
aws autoscaling exit-standby --instance-ids i-05b4f7d5be44822a6 --auto-scaling-
group-name my-asg
```

Berikut ini adalah contoh respons.

```
{
  "Activities": [
    {
      "ActivityId": "db12b166-cdcc-4c54-8aac-08c5935f8389",
      "AutoScalingGroupName": "my-asg",
      "Description": "Moving EC2 instance out of Standby:
i-05b4f7d5be44822a6",
      "Cause": "At 2023-12-15T21:46:14Z instance i-05b4f7d5be44822a6 was
moved out of standby in
      response to a user request, increasing the capacity from 3 to
4.",
      "StartTime": "2023-12-15T21:46:14.678Z",
      "StatusCode": "PreInService",
      "Progress": 30,
      "Details": "{\"Subnet ID\": \"subnet-c934b782\", \"Availability Zone
\": \"us-west-2a\"}"
    }
  ]
}
```

6. (Opsional) Verifikasi bahwa permintaan tersebut kembali dalam layanan menggunakan perintah `describe-auto-scaling-instances`.

```
aws autoscaling describe-auto-scaling-instances --instance-
ids i-05b4f7d5be44822a6
```

Berikut ini adalah contoh respons. Perhatikan bahwa status instance adalah InService.

```
{
  "AutoScalingInstances": [
    {
      "ProtectedFromScaleIn": false,
      "AvailabilityZone": "us-west-2a",
      "LaunchTemplate": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "1",
        "LaunchTemplateId": "lt-050555ad16a3f9c7f"
      },
      "InstanceId": "i-05b4f7d5be44822a6",
      "InstanceType": "t3.micro",
      "AutoScalingGroupName": "my-asg",
      "HealthStatus": "HEALTHY",
      "LifecycleState": "InService"
    },
    ...
  ]
}
```

Hapus infrastruktur Auto Scaling Anda

Untuk sepenuhnya menghapus infrastruktur penskalaan Anda, selesaikan tugas berikut.

Tugas

- [Hapus grup Auto Scaling Anda](#)
- [\(Opsional\) Hapus konfigurasi peluncuran](#)
- [\(Opsional\) Hapus templat peluncuran](#)
- [\(Opsional\) Hapus load balancer dan grup target](#)
- [\(Opsional\) Hapus CloudWatch alarm](#)

Hapus grup Auto Scaling Anda

Saat Anda menghapus grup Auto Scaling, nilai minimal dan maksimal yang diinginkan akan ditetapkan ke 0. Akibatnya, instans dihentikan. Menghapus suatu instans juga akan menghapus log atau data terkait, dan volume pada instans. Jika Anda tidak ingin menghentikan satu atau beberapa instance, Anda dapat melepaskannya sebelum menghapus grup Auto Scaling. Jika grup memiliki

kebijakan penskalaan, menghapus grup akan menghapus kebijakan, tindakan alarm yang mendasari, dan setiap alarm yang tidak lagi memiliki tindakan terkait.

Untuk menghapus grup Auto Scaling Anda (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda dan pilih Tindakan, Hapus.
3. Saat diminta konfirmasi, ketik **delete** untuk mengonfirmasi penghapusan grup Auto Scaling yang ditentukan, lalu pilih Hapus.

Ikon memuat di kolom Nama menunjukkan bahwa grup Auto Scaling dihapus. Kolom Diinginkan, Min, dan Maks menunjukkan instans 0 untuk grup Auto Scaling. Hanya perlu beberapa menit untuk menghentikan proses dan menghapus grup. Refresh daftar untuk melihat status saat ini.

Untuk menghapus grup Auto Scaling (AWS CLI)

Gunakan [delete-auto-scaling-group](#) perintah berikut untuk menghapus grup Auto Scaling. Operasi ini tidak berfungsi jika grup memiliki instans EC2; ini untuk grup dengan instance nol saja.

```
aws autoscaling delete-auto-scaling-group --auto-scaling-group-name my-asg
```

Jika grup memiliki instance atau aktivitas penskalaan yang sedang berlangsung, gunakan [delete-auto-scaling-group](#) perintah dengan opsi. `--force-delete` Ini juga akan menghentikan instans EC2. Saat Anda menghapus grup Auto Scaling dari konsol Auto Scaling Amazon EC2, konsol menggunakan operasi ini untuk menghentikan instans EC2 dan menghapus grup secara bersamaan.

```
aws autoscaling delete-auto-scaling-group --auto-scaling-group-name my-asg --force-delete
```

(Opsional) Hapus konfigurasi peluncuran

Anda dapat melewati langkah ini untuk mempertahankan konfigurasi peluncuran untuk penggunaan di masa mendatang.

Untuk menghapus konfigurasi peluncuran (konsol)

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

2. Di panel navigasi kiri, di bawah Auto Scaling, pilih Auto Scaling Groups.
3. Pilih Luncurkan konfigurasi di dekat bagian atas halaman. Saat diminta konfirmasi, pilih Lihat konfigurasi peluncuran untuk mengonfirmasi bahwa Anda ingin melihat halaman Konfigurasi peluncuran.
4. Pilih konfigurasi peluncuran Anda dan pilih Tindakan, Hapus konfigurasi peluncuran.
5. Saat diminta konfirmasi, pilih Hapus.

Untuk menghapus konfigurasi peluncuran (AWS CLI)

Gunakan perintah [delete-launch-configuration](#) berikut ini.

```
aws autoscaling delete-launch-configuration --launch-configuration-name my-launch-config
```

(Opsional) Hapus templat peluncuran

Anda dapat menghapus templat peluncuran atau hanya satu versi dari templat peluncuran Anda. Saat Anda menghapus templat peluncuran, semua versinya dihapus.

Anda dapat melewati langkah ini untuk mempertahankan templat peluncuran untuk penggunaan di masa mendatang.

Untuk menghapus templat peluncuran Anda (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pada panel navigasi, di bawah Instans, pilih Luncurkan Template.
3. Pilih templat peluncuran Anda dan lakukan salah satu hal berikut:
 - Pilih Tindakan, Hapus templat. Saat diminta konfirmasi, ketik **Delete** untuk mengonfirmasi penghapusan templat peluncuran yang ditentukan, lalu pilih Hapus.
 - Pilih Tindakan, Hapus versi templat. Pilih versi yang akan dihapus lalu pilih Hapus.

Untuk menghapus template peluncuran (AWS CLI)

Gunakan [delete-launch-template](#) perintah berikut untuk menghapus template Anda dan semua versinya.

```
aws ec2 delete-launch-template --launch-template-id lt-068f72b72934aff71
```

Atau, Anda dapat menggunakan [delete-launch-template-versions](#) perintah untuk menghapus versi tertentu dari template peluncuran.

```
aws ec2 delete-launch-template-versions --launch-template-id lt-068f72b72934aff71 --versions 1
```

(Opsional) Hapus load balancer dan grup target

Lewati langkah ini jika grup Auto Scaling Anda tidak terkait dengan load balancer Elastic Load Balancing, atau jika Anda ingin menyimpan load balancer untuk penggunaan di masa mendatang.

Untuk menghapus load balancer Anda (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pada panel navigasi, di bawah PENYEIMBANGAN BEBAN, pilih Penyeimbang beban.
3. Pilih load balancer dan pilih Tindakan, Hapus.
4. Ketika diminta konfirmasi, pilih Ya, Hapus.

Untuk menghapus grup target Anda (konsol)

1. Pada panel navigasi, di bawah Penyeimbangan Beban, pilih Grup Target.
2. Pilih grup target dan pilih Tindakan, Hapus.
3. Ketika diminta untuk mengonfirmasi, pilih Ya, Hapus.

Untuk menghapus penyeimbang beban yang terkait dengan grup Auto Scaling (AWS CLI)

Untuk Application Load Balancers dan Network Load Balancers, gunakan perintah berikut dan.

[delete-load-balancer](#)[delete-target-group](#)

```
aws elbv2 delete-load-balancer --load-balancer-arn my-load-balancer-arn  
aws elbv2 delete-target-group --target-group-arn my-target-group-arn
```

Untuk Classic Load Balancers, gunakan perintah berikut [delete-load-balancer](#).

```
aws elb delete-load-balancer --load-balancer-name my-load-balancer
```

(Opsional) Hapus CloudWatch alarm

Untuk menghapus CloudWatch alarm yang terkait dengan grup Auto Scaling Anda, selesaikan langkah-langkah berikut. Misalnya, Anda mungkin memiliki alarm yang terkait dengan penskalaan langkah atau kebijakan penskalaan sederhana.

Note

Menghapus grup Auto Scaling secara otomatis menghapus CloudWatch alarm yang dikelola Amazon EC2 Auto Scaling untuk kebijakan penskalaan pelacakan target.

Anda dapat melewati langkah ini jika grup Auto Scaling Anda tidak terkait dengan CloudWatch alarm apa pun, atau jika Anda ingin menyimpan alarm untuk digunakan di masa mendatang.

Untuk menghapus CloudWatch alarm (konsol)

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Pada panel navigasi, pilih Alarm.
3. Pilih alarm dan pilih Tindakan, Hapus.
4. Ketika diminta konfirmasi, pilih Hapus.

Untuk menghapus CloudWatch alarm (AWS CLI)

Gunakan perintah [hapus-alarm](#). Anda dapat menghapus satu atau beberapa alarm sekaligus. Misalnya, gunakan perintah berikut untuk menghapus alarm `Step-Scaling-AlarmHigh-AddCapacity` dan `Step-Scaling-AlarmLow-RemoveCapacity`.

```
aws cloudwatch delete-alarms --alarm-name Step-Scaling-AlarmHigh-AddCapacity Step-Scaling-AlarmLow-RemoveCapacity
```

Contoh untuk membuat dan mengelola grup Auto Scaling dengan SDK AWS

Anda dapat membuat grup Auto Scaling menggunakan AWS Management Console, AWS SDK AWS CLI, dan AWS CloudFormation

Contoh kode berikut menunjukkan cara membuat, memperbarui, mendeskripsikan, dan menghapus grup Auto Scaling dalam bahasa pemrograman favorit Anda yang didukung menggunakan SDK AWS .

Daftar Isi

- [Membuat grup Auto Scaling menggunakan SDK AWS](#)
- [Memperbarui grup Auto Scaling menggunakan SDK AWS](#)
- [Menjelaskan grup Auto Scaling menggunakan SDK AWS](#)
- [Menghapus grup Auto Scaling menggunakan SDK AWS](#)

Membuat grup Auto Scaling menggunakan SDK AWS

Contoh kode berikut menunjukkan cara menggunakan `CreateAutoScalingGroup`.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/// <summary>
/// Create a new Amazon EC2 Auto Scaling group.
/// </summary>
/// <param name="groupName">The name to use for the new Auto Scaling
/// group.</param>
/// <param name="launchTemplateName">The name of the Amazon EC2 Auto Scaling
/// launch template to use to create instances in the group.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> CreateAutoScalingGroupAsync(
    string groupName,
    string launchTemplateName,
    string availabilityZone)
{
    var templateSpecification = new LaunchTemplateSpecification
```

```
{
    LaunchTemplateName = launchTemplateName,
};

var zoneList = new List<string>
{
    availabilityZone,
};

var request = new CreateAutoScalingGroupRequest
{
    AutoScalingGroupName = groupName,
    AvailabilityZones = zoneList,
    LaunchTemplate = templateSpecification,
    MaxSize = 6,
    MinSize = 1
};

var response = await
_amazonAutoScaling.CreateAutoScalingGroupAsync(request);
Console.WriteLine($"{groupName} Auto Scaling Group created");
return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
}
```

- Untuk detail API, lihat [CreateAutoScalingGroup](#) di Referensi AWS SDK for .NET API.

C++

SDK for C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";
```

```

Aws::AutoScaling::AutoScalingClient autoScalingClient(clientConfig);

    Aws::AutoScaling::Model::CreateAutoScalingGroupRequest request;
    request.SetAutoScalingGroupName(groupName);
    Aws::Vector<Aws::String> availabilityGroupZones;
    availabilityGroupZones.push_back(
        availabilityZones[availabilityZoneChoice - 1].GetZoneName());
    request.SetAvailabilityZones(availabilityGroupZones);
    request.SetMaxSize(1);
    request.SetMinSize(1);

    Aws::AutoScaling::Model::LaunchTemplateSpecification
launchTemplateSpecification;
    launchTemplateSpecification.SetLaunchTemplateName(templateName);
    request.SetLaunchTemplate(launchTemplateSpecification);

    Aws::AutoScaling::Model::CreateAutoScalingGroupOutcome outcome =
        autoScalingClient.CreateAutoScalingGroup(request);

    if (outcome.IsSuccess()) {
        std::cout << "Created Auto Scaling group '" << groupName << "'..."
            << std::endl;
    }
    else if (outcome.GetError().GetErrorType() ==
        Aws::AutoScaling::AutoScalingErrors::ALREADY_EXISTS_FAULT) {
        std::cout << "Auto Scaling group '" << groupName << "' already
exists."
            << std::endl;
    }
    else {
        std::cerr << "Error with AutoScaling::CreateAutoScalingGroup. "
            << outcome.GetError().GetMessage()
            << std::endl;
    }
}

```

- Untuk detail API, lihat [CreateAutoScalingGroup](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk membuat grup Auto Scaling

`create-auto-scaling-group` Contoh berikut membuat grup Auto Scaling di subnet di beberapa Availability Zone dalam suatu Region. Instans diluncurkan dengan versi default dari template peluncuran yang ditentukan. Perhatikan bahwa default digunakan untuk sebagian besar pengaturan lain, seperti kebijakan penghentian dan konfigurasi pemeriksaan kesehatan.

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12 \  
  --min-size 1 \  
  --max-size 5 \  
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [grup Auto Scaling di Panduan Pengguna Auto Scaling Amazon EC2](#).

Contoh 2: Untuk melampirkan Application Load Balancer, Network Load Balancer, atau Gateway Load Balancer

Contoh ini menentukan ARN dari kelompok target untuk penyeimbang beban yang mendukung lalu lintas yang diharapkan. Jenis pemeriksaan kesehatan menentukan ELB sehingga ketika Elastic Load Balancing melaporkan sebuah instance sebagai tidak sehat, grup Auto Scaling menggantikannya. Perintah ini juga mendefinisikan masa tenggang pemeriksaan kesehatan 600 detik. Masa tenggang membantu mencegah penghentian prematur dari instans yang baru diluncurkan.

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12 \  
  --target-group-arns arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/my-targets/943f017f100becff \  
  --health-check-type ELB \  
  --health-check-grace-period 600 \  
  --min-size 1 \  
  --max-size 5 \  

```

```
--vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Elastic Load Balancing dan Amazon EC2 Auto Scaling](#) dalam Panduan Pengguna Amazon EC2 Auto Scaling.

Contoh 3: Untuk menentukan grup penempatan dan menggunakan versi terbaru dari template peluncuran

Contoh ini meluncurkan instance ke grup penempatan dalam Availability Zone tunggal. Ini dapat berguna untuk grup latensi rendah dengan beban kerja HPC. Contoh ini juga menentukan ukuran minimum, ukuran maksimum, dan kapasitas grup yang diinginkan.

```
aws autoscaling create-auto-scaling-group \
  --auto-scaling-group-name my-asg \
  --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest' \
  --min-size 1 \
  --max-size 5 \
  --desired-capacity 3 \
  --placement-group my-placement-group \
  --vpc-zone-identifier "subnet-6194ea3b"
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Grup penempatan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Contoh 4: Untuk menentukan satu contoh grup Auto Scaling dan menggunakan versi spesifik dari template peluncuran

Contoh ini membuat grup Auto Scaling dengan kapasitas minimum dan maksimum yang disetel 1 untuk menegaskan bahwa satu instance akan berjalan. Perintah ini juga menentukan v1 dari template peluncuran di mana ID ENI yang ada ditentukan. Bila Anda menggunakan template peluncuran yang menentukan ENI yang ada untuk eth0, Anda harus menentukan Availability Zone untuk grup Auto Scaling yang cocok dengan antarmuka jaringan, tanpa juga menentukan ID subnet dalam permintaan.

```
aws autoscaling create-auto-scaling-group \
  --auto-scaling-group-name my-asg-single-instance \
  --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='1' \
```

```
--min-size 1 \  
--max-size 1 \  
--availability-zones us-west-2a
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [grup Auto Scaling di Panduan Pengguna Auto Scaling Amazon EC2](#).

Contoh 5: Untuk menentukan kebijakan penghentian yang berbeda

Contoh ini membuat grup Auto Scaling menggunakan konfigurasi peluncuran dan menetapkan kebijakan penghentian untuk menghentikan instance tertua terlebih dahulu. Perintah ini juga menerapkan tag ke grup dan instance-nya, dengan kunci dari Role dan nilai WebServer

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-configuration-name my-lc \  
  --min-size 1 \  
  --max-size 5 \  
  --termination-policies "OldestInstance" \  
  --tags "ResourceId=my-asg,ResourceType=auto-scaling-  
group,Key=Role,Value=WebServer,PropagateAtLaunch=true" \  
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Bekerja dengan kebijakan penghentian Penskalaan Otomatis Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2](#).

Contoh 6: Untuk menentukan hook siklus hidup peluncuran

Contoh ini membuat grup Auto Scaling dengan pengait siklus hidup yang mendukung tindakan kustom saat peluncuran instance.

```
aws autoscaling create-auto-scaling-group \  
  --cli-input-json file://~/config.json
```

Isi config.json file:

```
{  
  "AutoScalingGroupName": "my-asg",
```

```
"LaunchTemplate": {
  "LaunchTemplateId": "lt-1234567890abcde12"
},
"LifecycleHookSpecificationList": [{
  "LifecycleHookName": "my-launch-hook",
  "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
  "NotificationTargetARN": "arn:aws:sqs:us-west-2:123456789012:my-sqs-queue",
  "RoleARN": "arn:aws:iam::123456789012:role/my-notification-role",
  "NotificationMetadata": "SQS message metadata",
  "HeartbeatTimeout": 4800,
  "DefaultResult": "ABANDON"
}],
"MinSize": 1,
"MaxSize": 5,
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
"Tags": [{
  "ResourceType": "auto-scaling-group",
  "ResourceId": "my-asg",
  "PropagateAtLaunch": true,
  "Value": "test",
  "Key": "environment"
}]
}
```

Perintah ini tidak menghasilkan output.

Untuk informasi lebih lanjut, lihat [Kait siklus hidup Amazon EC2 Auto Scaling](#) di Panduan Pengguna Amazon EC2 Auto Scaling.

Contoh 7: Untuk menentukan hook siklus hidup penghentian

Contoh ini membuat grup Auto Scaling dengan pengait siklus hidup yang mendukung tindakan kustom saat penghentian instance.

```
aws autoscaling create-auto-scaling-group \
  --cli-input-json file://~/config.json
```

Isi dari config.json:

```
{
  "AutoScalingGroupName": "my-asg",
  "LaunchTemplate": {
```

```

    "LaunchTemplateId": "lt-1234567890abcde12"
  },
  "LifecycleHookSpecificationList": [{
    "LifecycleHookName": "my-termination-hook",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING",
    "HeartbeatTimeout": 120,
    "DefaultResult": "CONTINUE"
  }],
  "MinSize": 1,
  "MaxSize": 5,
  "TargetGroupARNs": [
    "arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-
    targets/73e2d6bc24d8a067"
  ],
  "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
}

```

Perintah ini tidak menghasilkan output.

Untuk informasi lebih lanjut, lihat [Kait siklus hidup Amazon EC2 Auto Scaling](#) di Panduan Pengguna Amazon EC2 Auto Scaling.

Contoh 8: Untuk menentukan kebijakan penghentian kustom

Contoh ini membuat grup Auto Scaling yang menentukan kebijakan penghentian fungsi Lambda kustom yang memberi tahu Auto Scaling Amazon EC2 instans mana yang aman untuk dihentikan pada skala.

```

aws autoscaling create-auto-scaling-group \
  --auto-scaling-group-name my-asg-single-instance \
  --launch-template LaunchTemplateName=my-template-for-auto-scaling \
  --min-size 1 \
  --max-size 5 \
  --termination-policies "arn:aws:lambda:us-
  west-2:123456789012:function:HelloFunction:prod" \
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"

```


Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Membuat kebijakan penghentian kustom dengan Lambda di Panduan](#) Pengguna Auto Scaling Amazon EC2.

- Untuk detail API, lihat [CreateAutoScalingGroup](#) di Referensi AWS CLI Perintah.

Java

SDK for Java 2.x

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import software.amazon.awssdk.core.waiters.WaiterResponse;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.autoscaling.AutoScalingClient;
import software.amazon.awssdk.services.autoscaling.model.AutoScalingException;
import
    software.amazon.awssdk.services.autoscaling.model.CreateAutoScalingGroupRequest;
import
    software.amazon.awssdk.services.autoscaling.model.DescribeAutoScalingGroupsRequest;
import
    software.amazon.awssdk.services.autoscaling.model.DescribeAutoScalingGroupsResponse;
import
    software.amazon.awssdk.services.autoscaling.model.LaunchTemplateSpecification;
import software.amazon.awssdk.services.autoscaling.waiters.AutoScalingWaiter;

/**
 * Before running this SDK for Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class CreateAutoScalingGroup {
    public static void main(String[] args) {
        final String usage = ""

                Usage:
                <groupName> <launchTemplateName> <serviceLinkedRoleARN>
                <vpcZoneId>

                Where:
```

```
        groupName - The name of the Auto Scaling group.
        launchTemplateName - The name of the launch template.\s
        vpcZoneId - A subnet Id for a virtual private cloud (VPC)
where instances in the Auto Scaling group can be created.
        """;

    if (args.length != 3) {
        System.out.println(usage);
        System.exit(1);
    }

    String groupName = args[0];
    String launchTemplateName = args[1];
    String vpcZoneId = args[2];
    AutoScalingClient autoScalingClient = AutoScalingClient.builder()
        .region(Region.US_EAST_1)
        .build();

    createAutoScalingGroup(autoScalingClient, groupName, launchTemplateName,
vpcZoneId);
    autoScalingClient.close();
}

    public static void createAutoScalingGroup(AutoScalingClient
autoScalingClient,
        String groupName,
        String launchTemplateName,
        String vpcZoneId) {

    try {
        AutoScalingWaiter waiter = autoScalingClient.waiter();
        LaunchTemplateSpecification templateSpecification =
LaunchTemplateSpecification.builder()
            .launchTemplateName(launchTemplateName)
            .build();

        CreateAutoScalingGroupRequest request =
CreateAutoScalingGroupRequest.builder()
            .autoScalingGroupName(groupName)
            .availabilityZones("us-east-1a")
            .launchTemplate(templateSpecification)
            .maxSize(1)
            .minSize(1)
            .vpcZoneIdentifier(vpcZoneId)
```

```

        .build();

        autoScalingClient.createAutoScalingGroup(request);
        DescribeAutoScalingGroupsRequest groupsRequest =
DescribeAutoScalingGroupsRequest.builder()
        .autoScalingGroupNames(groupName)
        .build();

        WaiterResponse<DescribeAutoScalingGroupsResponse> waiterResponse =
waiter
        .waitUntilGroupExists(groupsRequest);
        waiterResponse.matched().response().ifPresent(System.out::println);
        System.out.println("Auto Scaling Group created");

    } catch (AutoScalingException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}
}

```

- Untuk detail API, lihat [CreateAutoScalingGroup](#) di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK for Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

suspend fun createAutoScalingGroup(groupName: String, launchTemplateNameVal:
String, serviceLinkedRoleARNVal: String, vpcZoneIdVal: String) {
    val templateSpecification = LaunchTemplateSpecification {
        launchTemplateName = launchTemplateNameVal
    }

    val request = CreateAutoScalingGroupRequest {
        autoScalingGroupName = groupName

```

```

        availabilityZones = listOf("us-east-1a")
        launchTemplate = templateSpecification
        maxSize = 1
        minSize = 1
        vpcZoneIdentifier = vpcZoneIdVal
        serviceLinkedRoleArn = serviceLinkedRoleARNVal
    }

    // This object is required for the waiter call.
    val groupsRequestWaiter = DescribeAutoScalingGroupsRequest {
        autoScalingGroupNames = listOf(groupName)
    }

    AutoScalingClient { region = "us-east-1" }.use { autoScalingClient ->
        autoScalingClient.createAutoScalingGroup(request)
        autoScalingClient.waitUntilGroupExists(groupsRequestWaiter)
        println("$groupName was created!")
    }
}

```

- Untuk detail API, lihat [CreateAutoScalingGroup](#) di AWS SDK untuk referensi API Kotlin.

PHP

SDK for PHP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

public function createAutoScalingGroup(
    $autoScalingGroupName,
    $availabilityZones,
    $minSize,
    $maxSize,
    $launchTemplateId
) {
    return $this->autoScalingClient->createAutoScalingGroup([
        'AutoScalingGroupName' => $autoScalingGroupName,

```

```

        'AvailabilityZones' => $availabilityZones,
        'MinSize' => $minSize,
        'MaxSize' => $maxSize,
        'LaunchTemplate' => [
            'LaunchTemplateId' => $launchTemplateId,
        ],
    ]);
}

```

- Untuk detail API, lihat [CreateAutoScalingGroup](#) di Referensi AWS SDK for PHP API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini membuat grup Auto Scaling dengan nama dan atribut yang ditentukan. Kapasitas default yang diinginkan adalah ukuran minimum. Oleh karena itu, grup Auto Scaling ini meluncurkan dua instance, satu di masing-masing dari dua Availability Zone yang ditentukan.

```

New-ASAutoScalingGroup -AutoScalingGroupName my-asg -LaunchConfigurationName my-
lc -MinSize 2 -MaxSize 6 -AvailabilityZone @("us-west-2a", "us-west-2b")

```

- Untuk detail API, lihat [CreateAutoScalingGroup](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK for Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

class AutoScalingWrapper:
    """Encapsulates Amazon EC2 Auto Scaling actions."""

```

```

def __init__(self, autoscaling_client):
    """
    :param autoscaling_client: A Boto3 Amazon EC2 Auto Scaling client.
    """
    self.autoscaling_client = autoscaling_client

def create_group(
    self, group_name, group_zones, launch_template_name, min_size, max_size
):
    """
    Creates an Auto Scaling group.

    :param group_name: The name to give to the group.
    :param group_zones: The Availability Zones in which instances can be
    created.
    :param launch_template_name: The name of an existing Amazon EC2 launch
    template.
                                The launch template specifies the
    configuration of
                                instances that are created by auto scaling
    activities.
    :param min_size: The minimum number of active instances in the group.
    :param max_size: The maximum number of active instances in the group.
    """
    try:
        self.autoscaling_client.create_auto_scaling_group(
            AutoScalingGroupName=group_name,
            AvailabilityZones=group_zones,
            LaunchTemplate={
                "LaunchTemplateName": launch_template_name,
                "Version": "$Default",
            },
            MinSize=min_size,
            MaxSize=max_size,
        )
    except ClientError as err:
        logger.error(
            "Couldn't create group %s. Here's why: %s: %s",
            group_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
    raise

```

- Untuk detail API, lihat [CreateAutoScalingGroup](#) di AWS SDK for Python (Boto3) Referensi API.

Rust

SDK for Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
async fn create_group(client: &Client, name: &str, id: &str) -> Result<(), Error>
{
    client
        .create_auto_scaling_group()
        .auto_scaling_group_name(name)
        .instance_id(id)
        .min_size(1)
        .max_size(5)
        .send()
        .await?;

    println!("Created AutoScaling group");

    Ok(())
}
```

- Untuk detail API, lihat [CreateAutoScalingGroup](#) referensi AWS SDK for Rust API.

Untuk contoh yang dapat Anda gunakan saat membuat [grup instance campuran](#), lihat sumber daya berikut.

- [AWS SDK for .NET](#)
- [AWS SDK for Go](#)

- [AWS SDK untuk JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK untuk Python](#)
- [AWS SDK for Ruby V3](#)

Memperbarui grup Auto Scaling menggunakan SDK AWS

Contoh kode berikut menunjukkan cara menggunakan `UpdateAutoScalingGroup`.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/// <summary>
/// Update the capacity of an Auto Scaling group.
/// </summary>
/// <param name="groupName">The name of the Auto Scaling group.</param>
/// <param name="launchTemplateName">The name of the EC2 launch template.</
param>
/// <param name="maxSize">The maximum number of instances that can be
/// created for the Auto Scaling group.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> UpdateAutoScalingGroupAsync(
    string groupName,
    string launchTemplateName,
    int maxSize)
{
    var templateSpecification = new LaunchTemplateSpecification
    {
        LaunchTemplateName = launchTemplateName,
    };

    var groupRequest = new UpdateAutoScalingGroupRequest
    {
```



```

        MaxSize = maxSize,
        AutoScalingGroupName = groupName,
        LaunchTemplate = templateSpecification,
    };

    var response = await
    _amazonAutoScaling.UpdateAutoScalingGroupAsync(groupRequest);
    if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
    {
        Console.WriteLine($"You successfully updated the Auto Scaling group
        {groupName}.");
        return true;
    }
    else
    {
        return false;
    }
}

```

- Untuk detail API, lihat [UpdateAutoScalingGroup](#) di Referensi AWS SDK for .NET API.

C++

SDK for C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::AutoScaling::AutoScalingClient autoScalingClient(clientConfig);

Aws::AutoScaling::Model::UpdateAutoScalingGroupRequest request;
request.SetAutoScalingGroupName(groupName);
request.SetMaxSize(3);

```

```
Aws::AutoScaling::Model::UpdateAutoScalingGroupOutcome outcome =
    autoScalingClient.UpdateAutoScalingGroup(request);

if (!outcome.IsSuccess()) {
    std::cerr << "Error with AutoScaling::UpdateAutoScalingGroup. "
                << outcome.GetError().GetMessage()
                << std::endl;
}
}
```

- Untuk detail API, lihat [UpdateAutoScalingGroup](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk memperbarui batas ukuran grup Auto Scaling

Contoh ini memperbarui grup Auto Scaling yang ditentukan dengan ukuran minimum 2 dan ukuran maksimum 10.

```
aws autoscaling update-auto-scaling-group \
  --auto-scaling-group-name my-asg \
  --min-size 2 \
  --max-size 10
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Menyetel batas kapasitas untuk grup Auto Scaling di Panduan Pengguna Auto Scaling Amazon EC2](#).

Contoh 2: Untuk menambahkan pemeriksaan kesehatan Elastic Load Balancing dan menentukan Availability Zone dan subnet mana yang akan digunakan

Contoh ini memperbarui grup Auto Scaling yang ditentukan untuk menambahkan pemeriksaan kesehatan Elastic Load Balancing. Perintah ini juga memperbarui nilai `--vpc-zone-identifier` dengan daftar ID subnet di beberapa Availability Zone.

```
aws autoscaling update-auto-scaling-group \
```

```
--auto-scaling-group-name my-asg \  
--health-check-type ELB \  
--health-check-grace-period 600 \  
--vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Elastic Load Balancing dan Amazon EC2 Auto Scaling](#) dalam Panduan Pengguna Amazon EC2 Auto Scaling.

Contoh 3: Untuk memperbarui grup penempatan dan kebijakan penghentian

Contoh ini memperbarui grup penempatan dan kebijakan penghentian yang akan digunakan.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --placement-group my-placement-group \  
  --termination-policies "OldestInstance"
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [grup Auto Scaling di Panduan Pengguna Auto Scaling Amazon EC2](#).

Contoh 4: Untuk menggunakan versi terbaru dari template peluncuran

Contoh ini memperbarui grup Auto Scaling yang ditentukan untuk menggunakan versi terbaru dari template peluncuran yang ditentukan.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest'
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Templat peluncuran](#) di Panduan Pengguna Amazon EC2 Auto Scaling.

Contoh 5: Untuk menggunakan versi tertentu dari template peluncuran

Contoh ini memperbarui grup Auto Scaling yang ditentukan untuk menggunakan versi tertentu dari template peluncuran, bukan versi terbaru atau default.

```
aws autoscaling update-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Templat peluncuran](#) di Panduan Pengguna Amazon EC2 Auto Scaling.

Contoh 6: Untuk menentukan kebijakan instans campuran dan mengaktifkan penyeimbangan kembali kapasitas

Contoh ini memperbarui grup Auto Scaling yang ditentukan untuk menggunakan kebijakan instance campuran dan memungkinkan penyeimbangan kembali kapasitas. Struktur ini memungkinkan Anda menentukan grup dengan kapasitas Spot dan Sesuai Permintaan dan menggunakan templat peluncuran yang berbeda untuk arsitektur yang berbeda.

```
aws autoscaling update-auto-scaling-group \  
  --cli-input-json file://~/config.json
```

Isi dari config.json:

```
{  
  "AutoScalingGroupName": "my-asg",  
  "CapacityRebalance": true,  
  "MixedInstancesPolicy": {  
    "LaunchTemplate": {  
      "LaunchTemplateSpecification": {  
        "LaunchTemplateName": "my-launch-template-for-x86",  
        "Version": "$Latest"  
      },  
      "Overrides": [  
        {  
          "InstanceType": "c6g.large",  
          "LaunchTemplateSpecification": {  
            "LaunchTemplateName": "my-launch-template-for-arm",  
            "Version": "$Latest"  
          }  
        },  
        {  
          "InstanceType": "c5.large"        }  
      ]  
    }  
  }  
}
```

```

        },
        {
            "InstanceType": "c5a.large"
        }
    ]
},
"InstancesDistribution": {
    "OnDemandPercentageAboveBaseCapacity": 50,
    "SpotAllocationStrategy": "capacity-optimized"
}
}
}

```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa tipe instans dan opsi pembelian](#) dalam Panduan Pengguna Amazon EC2 Auto Scaling.

- Untuk detail API, lihat [UpdateAutoScalingGroup](#) di Referensi AWS CLI Perintah.

Java

SDK for Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

public static void updateAutoScalingGroup(AutoScalingClient
autoScalingClient, String groupName,
    String launchTemplateName) {
    try {
        AutoScalingWaiter waiter = autoScalingClient.waiter();
        LaunchTemplateSpecification templateSpecification =
LaunchTemplateSpecification.builder()
            .launchTemplateName(launchTemplateName)
            .build();

        UpdateAutoScalingGroupRequest groupRequest =
UpdateAutoScalingGroupRequest.builder()

```

```

        .maxSize(3)
        .autoScalingGroupName(groupName)
        .launchTemplate(templateSpecification)
        .build();

        autoScalingClient.updateAutoScalingGroup(groupRequest);
        DescribeAutoScalingGroupsRequest groupsRequest =
DescribeAutoScalingGroupsRequest.builder()
        .autoScalingGroupNames(groupName)
        .build();

        WaiterResponse<DescribeAutoScalingGroupsResponse> waiterResponse =
waiter
        .waitUntilGroupInService(groupsRequest);
        waiterResponse.matched().response().ifPresent(System.out::println);
        System.out.println("You successfully updated the auto scaling group
" + groupName);

    } catch (AutoScalingException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}

```

- Untuk detail API, lihat [UpdateAutoScalingGroup](#) di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK for Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

suspend fun updateAutoScalingGroup(groupName: String, launchTemplateNameVal:
String, serviceLinkedRoleARNVal: String) {
    val templateSpecification = LaunchTemplateSpecification {
        launchTemplateName = launchTemplateNameVal
    }
}

```

```

val groupRequest = UpdateAutoScalingGroupRequest {
    maxSize = 3
    serviceLinkedRoleArn = serviceLinkedRoleARNVal
    autoScalingGroupName = groupName
    launchTemplate = templateSpecification
}

val groupsRequestWaiter = DescribeAutoScalingGroupsRequest {
    autoScalingGroupNames = listOf(groupName)
}

AutoScalingClient { region = "us-east-1" }.use { autoScalingClient ->
    autoScalingClient.updateAutoScalingGroup(groupRequest)
    autoScalingClient.waitUntilGroupExists(groupsRequestWaiter)
    println("You successfully updated the Auto Scaling group $groupName")
}
}

```

- Untuk detail API, lihat [UpdateAutoScalingGroup](#) di AWS SDK untuk referensi API Kotlin.

PHP

SDK for PHP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

public function updateAutoScalingGroup($autoScalingGroupName, $args)
{
    if (array_key_exists('MaxSize', $args)) {
        $maxSize = ['MaxSize' => $args['MaxSize']];
    } else {
        $maxSize = [];
    }
    if (array_key_exists('MinSize', $args)) {
        $minSize = ['MinSize' => $args['MinSize']];
    } else {

```

```
        $minSize = [];
    }
    $parameters = ['AutoScalingGroupName' => $autoScalingGroupName];
    $parameters = array_merge($parameters, $minSize, $maxSize);
    return $this->autoScalingClient->updateAutoScalingGroup($parameters);
}
```

- Untuk detail API, lihat [UpdateAutoScalingGroup](#) di Referensi AWS SDK for PHP API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini memperbarui ukuran minimum dan maksimum grup Auto Scaling yang ditentukan.

```
Update-ASAutoScalingGroup -AutoScalingGroupName my-asg -MaxSize 5 -MinSize 1
```

Contoh 2: Contoh ini memperbarui periode cooldown default dari grup Auto Scaling yang ditentukan.

```
Update-ASAutoScalingGroup -AutoScalingGroupName my-asg -DefaultCooldown 10
```

Contoh 3: Contoh ini memperbarui Availability Zones dari grup Auto Scaling yang ditentukan.

```
Update-ASAutoScalingGroup -AutoScalingGroupName my-asg -AvailabilityZone @("us-west-2a", "us-west-2b")
```

Contoh 4: Contoh ini memperbarui grup Auto Scaling yang ditentukan untuk menggunakan pemeriksaan kesehatan Elastic Load Balancing.

```
Update-ASAutoScalingGroup -AutoScalingGroupName my-asg -HealthCheckType ELB -HealthCheckGracePeriod 60
```

- Untuk detail API, lihat [UpdateAutoScalingGroup](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK for Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class AutoScalingWrapper:
    """Encapsulates Amazon EC2 Auto Scaling actions."""

    def __init__(self, autoscaling_client):
        """
        :param autoscaling_client: A Boto3 Amazon EC2 Auto Scaling client.
        """
        self.autoscaling_client = autoscaling_client

    def update_group(self, group_name, **kwargs):
        """
        Updates an Auto Scaling group.

        :param group_name: The name of the group to update.
        :param kwargs: Keyword arguments to pass through to the service.
        """
        try:
            self.autoscaling_client.update_auto_scaling_group(
                AutoScalingGroupName=group_name, **kwargs
            )
        except ClientError as err:
            logger.error(
                "Couldn't update group %s. Here's why: %s: %s",
                group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Untuk detail API, lihat [UpdateAutoScalingGroup](#) di AWS SDK for Python (Boto3) Referensi API.

Rust

SDK for Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
async fn update_group(client: &Client, name: &str, size: i32) -> Result<(),
Error> {
    client
        .update_auto_scaling_group()
        .auto_scaling_group_name(name)
        .max_size(size)
        .send()
        .await?;

    println!("Updated AutoScaling group");

    Ok(())
}
```

- Untuk detail API, lihat [UpdateAutoScalingGroup](#) referensi AWS SDK for Rust API.

Menjelaskan grup Auto Scaling menggunakan SDK AWS

Contoh kode berikut menunjukkan cara menggunakan `DescribeAutoScalingGroups`.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
/// <summary>
/// Get data about the instances in an Amazon EC2 Auto Scaling group.
/// </summary>
/// <param name="groupName">The name of the Amazon EC2 Auto Scaling group.</
param>
/// <returns>A list of Amazon EC2 Auto Scaling details.</returns>
public async Task<List<AutoScalingInstanceDetails>>
DescribeAutoScalingInstancesAsync(
    string groupName)
{
    var groups = await DescribeAutoScalingGroupsAsync(groupName);
    var instanceIds = new List<string>();
    groups!.ForEach(group =>
    {
        if (group.AutoScalingGroupName == groupName)
        {
            group.Instances.ForEach(instance =>
            {
                instanceIds.Add(instance.InstanceId);
            });
        }
    });

    var scalingGroupsRequest = new DescribeAutoScalingInstancesRequest
    {
        MaxRecords = 10,
        InstanceIds = instanceIds,
    };

    var response = await
_amazonAutoScaling.DescribeAutoScalingInstancesAsync(scalingGroupsRequest);
```

```

    var instanceDetails = response.AutoScalingInstances;

    return instanceDetails;
}

```

- Untuk detail API, lihat [DescribeAutoScalingGroups](#) di Referensi AWS SDK for .NET API.

C++

SDK for C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

    Aws::AutoScaling::AutoScalingClient autoScalingClient(clientConfig);

    Aws::AutoScaling::Model::DescribeAutoScalingGroupsRequest request;
    Aws::Vector<Aws::String> groupNames;
    groupNames.push_back(groupName);
    request.SetAutoScalingGroupNames(groupNames);

    Aws::AutoScaling::Model::DescribeAutoScalingGroupsOutcome outcome =
        client.DescribeAutoScalingGroups(request);

    if (outcome.IsSuccess()) {
        autoScalingGroup = outcome.GetResult().GetAutoScalingGroups();
    }
    else {
        std::cerr << "Error with AutoScaling::DescribeAutoScalingGroups. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
}

```

- Untuk detail API, lihat [DescribeAutoScalingGroups](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk menggambarkan grup Auto Scaling yang ditentukan

Contoh ini menjelaskan grup Auto Scaling yang ditentukan.

```
aws autoscaling describe-auto-scaling-groups \  
  --auto-scaling-group-name my-asg
```

Output:

```
{  
  "AutoScalingGroups": [  
    {  
      "AutoScalingGroupName": "my-asg",  
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:930d940e-891e-4781-a11a-7b0acd480f03:autoScalingGroupName/my-asg",  
      "LaunchTemplate": {  
        "LaunchTemplateName": "my-launch-template",  
        "Version": "1",  
        "LaunchTemplateId": "lt-1234567890abcde12"  
      },  
      "MinSize": 0,  
      "MaxSize": 1,  
      "DesiredCapacity": 1,  
      "DefaultCooldown": 300,  
      "AvailabilityZones": [  
        "us-west-2a",  
        "us-west-2b",  
        "us-west-2c"  
      ],  
      "LoadBalancerNames": [],  
      "TargetGroupARNs": [],  
      "HealthCheckType": "EC2",  
      "HealthCheckGracePeriod": 0,  
    }  
  ]  
}
```

```

    "Instances": [
      {
        "InstanceId": "i-06905f55584de02da",
        "InstanceType": "t2.micro",
        "AvailabilityZone": "us-west-2a",
        "HealthStatus": "Healthy",
        "LifecycleState": "InService",
        "ProtectedFromScaleIn": false,
        "LaunchTemplate": {
          "LaunchTemplateName": "my-launch-template",
          "Version": "1",
          "LaunchTemplateId": "lt-1234567890abcde12"
        }
      }
    ],
    "CreatedTime": "2023-10-28T02:39:22.152Z",
    "SuspendedProcesses": [],
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-
c934b782",
    "EnabledMetrics": [],
    "Tags": [],
    "TerminationPolicies": [
      "Default"
    ],
    "NewInstancesProtectedFromScaleIn": false,
    "ServiceLinkedRoleARN": "arn",
    "TrafficSources": []
  }
]
}

```

Contoh 2: Untuk mendeskripsikan 100 grup Auto Scaling pertama yang ditentukan

Contoh ini menjelaskan grup Auto Scaling yang ditentukan. Ini memungkinkan Anda untuk menentukan hingga 100 nama grup.

```

aws autoscaling describe-auto-scaling-groups \
  --max-items 100 \
  --auto-scaling-group-name "group1" "group2" "group3" "group4"

```

Lihat contoh 1 untuk output sampel.

Contoh 3: Untuk mendeskripsikan grup Auto Scaling di wilayah tertentu

Contoh ini menjelaskan grup Auto Scaling di wilayah tertentu, hingga maksimal 75 grup.

```
aws autoscaling describe-auto-scaling-groups \  
  --max-items 75 \  
  --region us-east-1
```

Lihat contoh 1 untuk output sampel.

Contoh 4: Untuk menjelaskan jumlah grup Auto Scaling yang ditentukan

Untuk mengembalikan sejumlah grup Auto Scaling tertentu, gunakan opsi. `--max-items`

```
aws autoscaling describe-auto-scaling-groups \  
  --max-items 1
```

Lihat contoh 1 untuk output sampel.

Jika output termasuk `NextToken` bidang, ada lebih banyak grup. Untuk mendapatkan grup tambahan, gunakan nilai bidang ini dengan `--starting-token` opsi dalam panggilan berikutnya sebagai berikut.

```
aws autoscaling describe-auto-scaling-groups \  
  --starting-token Z3M3LMPEXAMPLE
```

Lihat contoh 1 untuk output sampel.

Contoh 5: Untuk menjelaskan grup Auto Scaling yang menggunakan konfigurasi peluncuran

Contoh ini menggunakan `--query` opsi untuk mendeskripsikan grup Auto Scaling yang menggunakan konfigurasi peluncuran.

```
aws autoscaling describe-auto-scaling-groups \  
  --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`]'
```

Output:

```
[  
  {  
    "AutoScalingGroupName": "my-asg",  
    "AutoScalingGroupARN": "arn:aws:autoscaling:us-  
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-  
a11a-7b0acd480f03:autoScalingGroupName/my-asg",
```

```
"LaunchConfigurationName": "my-lc",
"MinSize": 0,
"MaxSize": 1,
"DesiredCapacity": 1,
"DefaultCooldown": 300,
"AvailabilityZones": [
  "us-west-2a",
  "us-west-2b",
  "us-west-2c"
],
"LoadBalancerNames": [],
"TargetGroupARNs": [],
"HealthCheckType": "EC2",
"HealthCheckGracePeriod": 0,
"Instances": [
  {
    "InstanceId": "i-088c57934a6449037",
    "InstanceType": "t2.micro",
    "AvailabilityZone": "us-west-2c",
    "HealthStatus": "Healthy",
    "LifecycleState": "InService",
    "LaunchConfigurationName": "my-lc",
    "ProtectedFromScaleIn": false
  }
],
"CreatedTime": "2023-10-28T02:39:22.152Z",
"SuspendedProcesses": [],
"VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782",
"EnabledMetrics": [],
"Tags": [],
"TerminationPolicies": [
  "Default"
],
"NewInstancesProtectedFromScaleIn": false,
"ServiceLinkedRoleARN": "arn",
"TrafficSources": []
}
]
```

Untuk informasi selengkapnya, lihat [Memfilter output AWS CLI](#) di Panduan Pengguna Antarmuka Baris AWS Perintah.

- Untuk detail API, lihat [DescribeAutoScalingGroups](#) di Referensi AWS CLI Perintah.

Java

SDK for Java 2.x

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.autoscaling.AutoScalingClient;
import software.amazon.awssdk.services.autoscaling.model.AutoScalingException;
import software.amazon.awssdk.services.autoscaling.model.AutoScalingGroup;
import
    software.amazon.awssdk.services.autoscaling.model.DescribeAutoScalingGroupsResponse;
import
    software.amazon.awssdk.services.autoscaling.model.DescribeAutoScalingGroupsRequest;
import software.amazon.awssdk.services.autoscaling.model.Instance;
import java.util.List;

/**
 * Before running this SDK for Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class DescribeAutoScalingInstances {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <groupName>

            Where:
                groupName - The name of the Auto Scaling group.
            """;

        if (args.length != 1) {
```

```
        System.out.println(usage);
        System.exit(1);
    }

    String groupName = args[0];
    AutoScalingClient autoScalingClient = AutoScalingClient.builder()
        .region(Region.US_EAST_1)
        .build();

    String instanceId = getAutoScaling(autoScalingClient, groupName);
    System.out.println(instanceId);
    autoScalingClient.close();
}

public static String getAutoScaling(AutoScalingClient autoScalingClient,
String groupName) {
    try {
        String instanceId = "";
        DescribeAutoScalingGroupsRequest scalingGroupsRequest =
DescribeAutoScalingGroupsRequest.builder()
            .autoScalingGroupNames(groupName)
            .build();

        DescribeAutoScalingGroupsResponse response = autoScalingClient
            .describeAutoScalingGroups(scalingGroupsRequest);
        List<AutoScalingGroup> groups = response.autoScalingGroups();
        for (AutoScalingGroup group : groups) {
            System.out.println("The group name is " +
group.autoScalingGroupName());
            System.out.println("The group ARN is " +
group.autoScalingGroupARN());

            List<Instance> instances = group.instances();
            for (Instance instance : instances) {
                instanceId = instance.instanceId();
            }
        }
        return instanceId;
    } catch (AutoScalingException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
    return "";
}
```

```
}

```

- Untuk detail API, lihat [DescribeAutoScalingGroups](#) di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK for Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
suspend fun getAutoScalingGroups(groupName: String) {
    val scalingGroupsRequest = DescribeAutoScalingGroupsRequest {
        autoScalingGroupNames = listOf(groupName)
    }

    AutoScalingClient { region = "us-east-1" }.use { autoScalingClient ->
        val response =
            autoScalingClient.describeAutoScalingGroups(scalingGroupsRequest)
        response.autoScalingGroups?.forEach { group ->
            println("The group name is ${group.autoScalingGroupName}")
            println("The group ARN is ${group.autoScalingGroupArn}")
            group.instances?.forEach { instance ->
                println("The instance id is ${instance.instanceId}")
                println("The lifecycle state is " + instance.lifecycleState)
            }
        }
    }
}

```

- Untuk detail API, lihat [DescribeAutoScalingGroups](#) di AWS SDK untuk referensi API Kotlin.

PHP

SDK for PHP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
public function describeAutoScalingGroups($autoScalingGroupNames)
{
    return $this->autoScalingClient->describeAutoScalingGroups([
        'AutoScalingGroupNames' => $autoScalingGroupNames
    ]);
}
```

- Untuk detail API, lihat [DescribeAutoScalingGroups](#) di Referensi AWS SDK for PHP API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini mencantumkan nama grup Auto Scaling Anda.

```
Get-ASAutoScalingGroup | format-table -property AutoScalingGroupName
```

Output:

```
AutoScalingGroupName
-----
my-asg-1
my-asg-2
my-asg-3
my-asg-4
my-asg-5
my-asg-6
```

Contoh 2: Contoh ini menjelaskan grup Auto Scaling yang ditentukan.

```
Get-ASAutoScalingGroup -AutoScalingGroupName my-asg-1
```

Output:

```
AutoScalingGroupARN      : arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:930d940e-891e-4781-a11a-7b0acd480
                           f03:autoScalingGroupName/my-asg-1
AutoScalingGroupName     : my-asg-1
AvailabilityZones        : {us-west-2b, us-west-2a}
CreatedTime              : 3/1/2015 9:05:31 AM
DefaultCooldown          : 300
DesiredCapacity          : 2
EnabledMetrics            : {}
HealthCheckGracePeriod   : 300
HealthCheckType          : EC2
Instances                : {my-lc}
LaunchConfigurationName  : my-lc
LoadBalancerNames        : {}
MaxSize                  : 0
MinSize                  : 0
PlacementGroup           :
Status                   :
SuspendedProcesses       : {}
Tags                     : {}
TerminationPolicies      : {Default}
VPCZoneIdentifier        : subnet-e4f33493,subnet-5264e837
```

Contoh 3: Contoh ini menjelaskan dua grup Auto Scaling yang ditentukan.

```
Get-ASAutoScalingGroup -AutoScalingGroupName @"my-asg-1", "my-asg-2"
```

Contoh 4: Contoh ini menjelaskan instance Auto Scaling untuk grup Auto Scaling yang ditentukan.

```
(Get-ASAutoScalingGroup -AutoScalingGroupName my-asg-1).Instances
```

Contoh 5: Contoh ini menjelaskan semua grup Auto Scaling Anda.

```
Get-ASAutoScalingGroup
```

Contoh 6: Contoh ini menjelaskan semua grup Auto Scaling Anda, dalam batch 10.

```
$nextToken = $null
do {
    Get-ASAutoScalingGroup -NextToken $nextToken -MaxRecord 10
    $nextToken = $AWSHistory.LastServiceResponse.NextToken
} while ($nextToken -ne $null)
```

Contoh 7: Contoh ini menjelaskan LaunchTemplate untuk grup Auto Scaling yang ditentukan. Contoh ini mengasumsikan bahwa “Opsi pembelian instans” diatur ke “Patuhi templat peluncuran”. Jika opsi ini diatur ke “Gabungkan opsi pembelian dan jenis instance”, LaunchTemplate dapat diakses menggunakan "MixedInstancesPolicy.LaunchTemplate" properti.

```
(Get-ASAutoScalingGroup -AutoScalingGroupName my-ag-1).LaunchTemplate
```

Output:

LaunchTemplateId	LaunchTemplateName	Version
lt-06095fd619cb40371	test-launch-template	\$Default

- Untuk detail API, lihat [DescribeAutoScalingGroups](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK for Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class AutoScalingWrapper:
    """Encapsulates Amazon EC2 Auto Scaling actions."""

    def __init__(self, autoscaling_client):
        """
```

```
        :param autoscaling_client: A Boto3 Amazon EC2 Auto Scaling client.
        """
        self.autoscaling_client = autoscaling_client

def describe_group(self, group_name):
    """
    Gets information about an Auto Scaling group.

    :param group_name: The name of the group to look up.
    :return: Information about the group, if found.
    """
    try:
        response = self.autoscaling_client.describe_auto_scaling_groups(
            AutoScalingGroupNames=[group_name]
        )
    except ClientError as err:
        logger.error(
            "Couldn't describe group %s. Here's why: %s: %s",
            group_name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        groups = response.get("AutoScalingGroups", [])
        return groups[0] if len(groups) > 0 else None
```

- Untuk detail API, lihat [DescribeAutoScalingGroups](#) di AWS SDK for Python (Boto3) Referensi API.

Rust

SDK for Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
async fn list_groups(client: &Client) -> Result<(), Error> {
    let resp = client.describe_auto_scaling_groups().send().await?;

    println!("Groups:");

    let groups = resp.auto_scaling_groups();

    for group in groups {
        println!(
            "Name: {}",
            group.auto_scaling_group_name().unwrap_or("Unknown")
        );
        println!(
            "Arn: {}",
            group.auto_scaling_group_arn().unwrap_or("unknown"),
        );
        println!("Zones: {:?}", group.availability_zones(),);
        println!();
    }

    println!("Found {} group(s)", groups.len());

    Ok(())
}
```

- Untuk detail API, lihat [DescribeAutoScalingGroups](#) referensi AWS SDK for Rust API.

Menghapus grup Auto Scaling menggunakan SDK AWS

Contoh kode berikut menunjukkan cara menggunakan `DeleteAutoScalingGroup`.

.NET

AWS SDK for .NET

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Perbarui ukuran minimum grup Auto Scaling ke nol, hentikan semua instance dalam grup, dan hapus grup.

```

/// <summary>
/// Try to terminate an instance by its Id.
/// </summary>
/// <param name="instanceId">The Id of the instance to terminate.</param>
/// <returns>Async task.</returns>
public async Task TryTerminateInstanceById(string instanceId)
{
    var stopping = false;
    Console.WriteLine($"Stopping {instanceId}...");
    while (!stopping)
    {
        try
        {
            await
            _amazonAutoScaling.TerminateInstanceInAutoScalingGroupAsync(
                new TerminateInstanceInAutoScalingGroupRequest()
                {
                    InstanceId = instanceId,
                    ShouldDecrementDesiredCapacity = false
                });
            stopping = true;
        }
        catch (ScalingActivityInProgressException)
        {
            Console.WriteLine($"Scaling activity in progress for
{instanceId}. Waiting...");
            Thread.Sleep(10000);
        }
    }
}

/// <summary>
/// Tries to delete the EC2 Auto Scaling group. If the group is in use or in
progress,
/// waits and retries until the group is successfully deleted.
/// </summary>
/// <param name="groupName">The name of the group to try to delete.</param>
/// <returns>Async task.</returns>
public async Task TryDeleteGroupByName(string groupName)
{

```

```

    var stopped = false;
    while (!stopped)
    {
        try
        {
            await _amazonAutoScaling.DeleteAutoScalingGroupAsync(
                new DeleteAutoScalingGroupRequest()
                {
                    AutoScalingGroupName = groupName
                });
            stopped = true;
        }
        catch (Exception e)
            when ((e is ScalingActivityInProgressException)
                || (e is Amazon.AutoScaling.Model.ResourceInUseException))
        {
            Console.WriteLine($"Some instances are still running.
Waiting...");
            Thread.Sleep(10000);
        }
    }
}

/// <summary>
/// Terminate instances and delete the Auto Scaling group by name.
/// </summary>
/// <param name="groupName">The name of the group to delete.</param>
/// <returns>Async task.</returns>
public async Task TerminateAndDeleteAutoScalingGroupWithName(string
groupName)
{
    var describeGroupsResponse = await
_amazonAutoScaling.DescribeAutoScalingGroupsAsync(
    new DescribeAutoScalingGroupsRequest()
    {
        AutoScalingGroupNames = new List<string>() { groupName }
    });
    if (describeGroupsResponse.AutoScalingGroups.Any())
    {
        // Update the size to 0.
        await _amazonAutoScaling.UpdateAutoScalingGroupAsync(
            new UpdateAutoScalingGroupRequest()
            {
                AutoScalingGroupName = groupName,

```

```

        MinSize = 0
    });
    var group = describeGroupsResponse.AutoScalingGroups[0];
    foreach (var instance in group.Instances)
    {
        await TryTerminateInstanceById(instance.InstanceId);
    }

    await TryDeleteGroupByName(groupName);
}
else
{
    Console.WriteLine($"No groups found with name {groupName}.");
}
}

```

```

/// <summary>
/// Delete an Auto Scaling group.
/// </summary>
/// <param name="groupName">The name of the Amazon EC2 Auto Scaling group.</
param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteAutoScalingGroupAsync(
    string groupName)
{
    var deleteAutoScalingGroupRequest = new DeleteAutoScalingGroupRequest
    {
        AutoScalingGroupName = groupName,
        ForceDelete = true,
    };

    var response = await
_amazonAutoScaling.DeleteAutoScalingGroupAsync(deleteAutoScalingGroupRequest);
    if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
    {
        Console.WriteLine($"You successfully deleted {groupName}");
        return true;
    }

    Console.WriteLine($"Couldn't delete {groupName}.");
    return false;
}

```

```
}

```

- Untuk detail API, lihat [DeleteAutoScalingGroup](#) di Referensi AWS SDK for .NET API.

C++

SDK for C++

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
Aws::Client::ClientConfiguration clientConfig;
// Optional: Set to the AWS Region (overrides config file).
// clientConfig.region = "us-east-1";

Aws::AutoScaling::AutoScalingClient autoScalingClient(clientConfig);

Aws::AutoScaling::Model::DeleteAutoScalingGroupRequest request;
request.SetAutoScalingGroupName(groupName);

Aws::AutoScaling::Model::DeleteAutoScalingGroupOutcome outcome =
    autoScalingClient.DeleteAutoScalingGroup(request);

if (outcome.IsSuccess()) {
    std::cout << "Auto Scaling group '" << groupName << "' was
deleted."
                << std::endl;
}
else {
    std::cerr << "Error with AutoScaling::DeleteAutoScalingGroup. "
              << outcome.GetError().GetMessage()
              << std::endl;
    result = false;
}
}
```

- Untuk detail API, lihat [DeleteAutoScalingGroup](#) di Referensi AWS SDK for C++ API.

CLI

AWS CLI

Contoh 1: Untuk menghapus grup Auto Scaling yang ditentukan

Contoh ini menghapus grup Auto Scaling yang ditentukan.

```
aws autoscaling delete-auto-scaling-group \  
  --auto-scaling-group-name my-asg
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Menghapus infrastruktur Auto Scaling](#) di Panduan Pengguna Auto Scaling Amazon EC2.

Contoh 2: Untuk menghapus paksa grup Auto Scaling yang ditentukan

Untuk menghapus grup Auto Scaling tanpa menunggu instance dalam grup dihentikan, gunakan opsi. `--force-delete`

```
aws autoscaling delete-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --force-delete
```

Perintah ini tidak menghasilkan output.

Untuk informasi selengkapnya, lihat [Menghapus infrastruktur Auto Scaling](#) di Panduan Pengguna Auto Scaling Amazon EC2.

- Untuk detail API, lihat [DeleteAutoScalingGroup](#) di Referensi AWS CLI Perintah.

Java

SDK for Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.autoscaling.AutoScalingClient;
import software.amazon.awssdk.services.autoscaling.model.AutoScalingException;
import
    software.amazon.awssdk.services.autoscaling.model.DeleteAutoScalingGroupRequest;

/**
 * Before running this SDK for Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class DeleteAutoScalingGroup {
    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <groupName>

            Where:
                groupName - The name of the Auto Scaling group.
            """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String groupName = args[0];
```

```

        AutoScalingClient autoScalingClient = AutoScalingClient.builder()
            .region(Region.US_EAST_1)
            .build();

        deleteAutoScalingGroup(autoScalingClient, groupName);
        autoScalingClient.close();
    }

    public static void deleteAutoScalingGroup(AutoScalingClient
autoScalingClient, String groupName) {
        try {
            DeleteAutoScalingGroupRequest deleteAutoScalingGroupRequest =
DeleteAutoScalingGroupRequest.builder()
                .autoScalingGroupName(groupName)
                .forceDelete(true)
                .build();

            autoScalingClient.deleteAutoScalingGroup(deleteAutoScalingGroupRequest);
            System.out.println("You successfully deleted " + groupName);

        } catch (AutoScalingException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}

```

- Untuk detail API, lihat [DeleteAutoScalingGroup](#) di Referensi AWS SDK for Java 2.x API.

Kotlin

SDK for Kotlin

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
suspend fun deleteSpecificAutoScalingGroup(groupName: String) {
```

```
val deleteAutoScalingGroupRequest = DeleteAutoScalingGroupRequest {
    autoScalingGroupName = groupName
    forceDelete = true
}

AutoScalingClient { region = "us-east-1" }.use { autoScalingClient ->
    autoScalingClient.deleteAutoScalingGroup(deleteAutoScalingGroupRequest)
    println("You successfully deleted $groupName")
}
}
```

- Untuk detail API, lihat [DeleteAutoScalingGroup](#) di AWS SDK untuk referensi API Kotlin.

PHP

SDK for PHP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
public function deleteAutoScalingGroup($autoScalingGroupName)
{
    return $this->autoScalingClient->deleteAutoScalingGroup([
        'AutoScalingGroupName' => $autoScalingGroupName,
        'ForceDelete' => true,
    ]);
}
```

- Untuk detail API, lihat [DeleteAutoScalingGroup](#) di Referensi AWS SDK for PHP API.

PowerShell

Alat untuk PowerShell

Contoh 1: Contoh ini menghapus grup Auto Scaling yang ditentukan jika tidak memiliki instance yang berjalan. Anda diminta untuk konfirmasi sebelum operasi berlangsung.

```
Remove-ASAutoScalingGroup -AutoScalingGroupName my-asg
```

Output:

```
Confirm
Are you sure you want to perform this action?
Performing operation "Remove-ASAutoScalingGroup (DeleteAutoScalingGroup)" on
Target "my-asg".
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is
"Y"):
```

Contoh 2: Jika Anda menentukan parameter Force, Anda tidak diminta untuk konfirmasi sebelum operasi berlangsung.

```
Remove-ASAutoScalingGroup -AutoScalingGroupName my-asg -Force
```

Contoh 3: Contoh ini menghapus grup Auto Scaling yang ditentukan dan mengakhiri semua instance yang sedang berjalan yang dikandungnya.

```
Remove-ASAutoScalingGroup -AutoScalingGroupName my-asg -ForceDelete $true -Force
```

- Untuk detail API, lihat [DeleteAutoScalingGroup](#) di Referensi AWS Tools for PowerShell Cmdlet.

Python

SDK for Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Perbarui ukuran minimum grup Auto Scaling ke nol, hentikan semua instance dalam grup, dan hapus grup.

```
class AutoScaler:
    """
    Encapsulates Amazon EC2 Auto Scaling and EC2 management actions.
    """

    def __init__(
        self,
        resource_prefix,
        inst_type,
        ami_param,
        autoscaling_client,
        ec2_client,
        ssm_client,
        iam_client,
    ):
        """
        :param resource_prefix: The prefix for naming AWS resources that are
        created by this class.
        :param inst_type: The type of EC2 instance to create, such as t3.micro.
        :param ami_param: The Systems Manager parameter used to look up the AMI
        that is
                created.
        :param autoscaling_client: A Boto3 EC2 Auto Scaling client.
        :param ec2_client: A Boto3 EC2 client.
        :param ssm_client: A Boto3 Systems Manager client.
        :param iam_client: A Boto3 IAM client.
        """
        self.inst_type = inst_type
        self.ami_param = ami_param
        self.autoscaling_client = autoscaling_client
        self.ec2_client = ec2_client
        self.ssm_client = ssm_client
        self.iam_client = iam_client
        self.launch_template_name = f"{resource_prefix}-template"
        self.group_name = f"{resource_prefix}-group"
        self.instance_policy_name = f"{resource_prefix}-pol"
        self.instance_role_name = f"{resource_prefix}-role"
        self.instance_profile_name = f"{resource_prefix}-prof"
        self.bad_creds_policy_name = f"{resource_prefix}-bc-pol"
        self.bad_creds_role_name = f"{resource_prefix}-bc-role"
```

```

self.bad_creds_profile_name = f"{resource_prefix}-bc-prof"
self.key_pair_name = f"{resource_prefix}-key-pair"

def _try_terminate_instance(self, inst_id):
    stopping = False
    log.info(f"Stopping {inst_id}.")
    while not stopping:
        try:
            self.autoscaling_client.terminate_instance_in_auto_scaling_group(
                InstanceId=inst_id, ShouldDecrementDesiredCapacity=True
            )
            stopping = True
        except ClientError as err:
            if err.response["Error"]["Code"] == "ScalingActivityInProgress":
                log.info("Scaling activity in progress for %s. Waiting...",
inst_id)
                time.sleep(10)
            else:
                raise AutoScalerError(f"Couldn't stop instance {inst_id}:
{err}.")

def _try_delete_group(self):
    """
    Tries to delete the EC2 Auto Scaling group. If the group is in use or in
progress,
the function waits and retries until the group is successfully deleted.
    """
    stopped = False
    while not stopped:
        try:
            self.autoscaling_client.delete_auto_scaling_group(
                AutoScalingGroupName=self.group_name
            )
            stopped = True
            log.info("Deleted EC2 Auto Scaling group %s.", self.group_name)
        except ClientError as err:
            if (
                err.response["Error"]["Code"] == "ResourceInUse"
                or err.response["Error"]["Code"] ==
"ScalingActivityInProgress"
            ):
                log.info(

```

```

        "Some instances are still running. Waiting for them to
stop..."
    )
    time.sleep(10)
    else:
        raise AutoScalerError(
            f"Couldn't delete group {self.group_name}: {err}."
        )

def delete_group(self):
    """
    Terminates all instances in the group, deletes the EC2 Auto Scaling
group.
    """
    try:
        response = self.autoscaling_client.describe_auto_scaling_groups(
            AutoScalingGroupNames=[self.group_name]
        )
        groups = response.get("AutoScalingGroups", [])
        if len(groups) > 0:
            self.autoscaling_client.update_auto_scaling_group(
                AutoScalingGroupName=self.group_name, MinSize=0
            )
            instance_ids = [inst["InstanceId"] for inst in groups[0]
["Instances"]]
            for inst_id in instance_ids:
                self._try_terminate_instance(inst_id)
                self._try_delete_group()
            else:
                log.info("No groups found named %s, nothing to do.",
self.group_name)
        except ClientError as err:
            raise AutoScalerError(f"Couldn't delete group {self.group_name}:
{err}.")

```

- Untuk detail API, lihat [DeleteAutoScalingGroup](#) di AWS SDK for Python (Boto3) Referensi API.

Rust

SDK for Rust

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
async fn delete_group(client: &Client, name: &str, force: bool) -> Result<(),
Error> {
    client
        .delete_auto_scaling_group()
        .auto_scaling_group_name(name)
        .set_force_delete(if force { Some(true) } else { None })
        .send()
        .await?;

    println!("Deleted Auto Scaling group");

    Ok(())
}
```

- Untuk detail API, lihat [DeleteAutoScalingGroup](#) referensi AWS SDK for Rust API.

Daur ulang instans di grup Auto Scaling Anda

Auto Scaling Amazon EC2 menawarkan kemampuan yang memungkinkan Anda mengganti instans Amazon EC2 di grup Auto Scaling setelah melakukan pembaruan, seperti menambahkan templat peluncuran baru dengan Amazon Machine Image (AMI) baru atau menambahkan jenis instans baru. Ini juga membantu Anda merampingkan pembaruan dengan memberi Anda opsi untuk memasukkannya ke dalam operasi yang sama yang menggantikan instance.

Bagian ini mencakup informasi untuk membantu Anda melakukan hal berikut:

- Mulai penyegaran instans untuk mengganti instance di grup Auto Scaling Anda.
- Deklarasikan pembaruan spesifik yang menjelaskan konfigurasi yang diinginkan dan perbarui grup Auto Scaling ke konfigurasi yang diinginkan.
- Lewati mengganti instance yang sudah diperbarui.
- Gunakan pos pemeriksaan untuk memperbarui instance secara bertahap dan melakukan verifikasi pada instans Anda di titik-titik tertentu.
- Terima pemberitahuan melalui email saat pos pemeriksaan tercapai.
- Gunakan rollback untuk mengembalikan grup Auto Scaling ke konfigurasi yang sebelumnya digunakan.
- Secara otomatis memutar kembali jika penyegaran instance gagal karena alasan tertentu atau jika ada CloudWatch alarm Amazon yang Anda tentukan masuk ke ALARM status.
- Batasi masa pakai instans untuk menyediakan versi perangkat lunak dan konfigurasi instans yang konsisten di seluruh grup Auto Scaling.

Daftar Isi

- [Menggunakan penyegaran instans untuk memperbarui instance dalam grup Auto Scaling](#)
- [Ganti instans Auto Scaling berdasarkan masa pakai instans maksimum](#)

Menggunakan penyegaran instans untuk memperbarui instance dalam grup Auto Scaling

Anda dapat menggunakan penyegaran instans untuk memperbarui instans di grup Auto Scaling Anda. Fitur ini dapat berguna ketika perubahan konfigurasi mengharuskan Anda untuk mengganti instance, terutama jika grup Auto Scaling Anda berisi sejumlah besar instance.

Beberapa situasi di mana penyegaran instance dapat membantu meliputi:

- Menerapkan Amazon Machine Image (AMI) atau skrip data pengguna baru di seluruh grup Auto Scaling Anda. Anda dapat membuat template peluncuran baru dengan perubahan dan kemudian menggunakan penyegaran instance untuk segera meluncurkan pembaruan.
- Memigrasikan instans Anda ke jenis instans baru untuk memanfaatkan peningkatan dan pengoptimalan terbaru.
- Mengalihkan grup Auto Scaling Anda dari menggunakan konfigurasi peluncuran menjadi menggunakan template peluncuran. Anda dapat menyalin konfigurasi peluncuran untuk meluncurkan templat dan kemudian menggunakan penyegaran instans untuk memperbarui instance Anda ke templat baru. Untuk informasi selengkapnya tentang migrasi ke templat peluncuran, lihat [Migrasi grup Auto Scaling Anda untuk meluncurkan template](#).

Daftar Isi

- [Cara kerja penyegaran instance](#)
- [Memahami nilai default untuk penyegaran instance](#)
- [Mulai penyegaran instance](#)
- [Pantau penyegaran instance](#)
- [Batalkan penyegaran instance](#)
- [Batalkan perubahan dengan rollback](#)
- [Gunakan penyegaran instance dengan melewati pencocokan](#)
- [Tambahkan pos pemeriksaan ke penyegaran instance](#)

Cara kerja penyegaran instance

Topik ini menjelaskan cara kerja penyegaran instance dan memperkenalkan konsep kunci yang perlu Anda pahami untuk menggunakannya secara efektif.

Daftar Isi

- [Cara kerjanya](#)
- [Konsep inti](#)
- [Masa tenggang pemeriksaan kesehatan](#)
- [Kompatibilitas tipe instans](#)
- [Batasan](#)

Cara kerjanya

Untuk menyegarkan instance dalam grup Auto Scaling, Anda dapat menentukan konfigurasi baru yang berisi versi terbaru aplikasi Anda dan pembaruan lain yang ingin Anda buat. Kemudian, mulai penyegaran instance untuk mengganti instance yang ada dengan yang baru berdasarkan konfigurasi itu.

Untuk melakukan penyegaran instance:

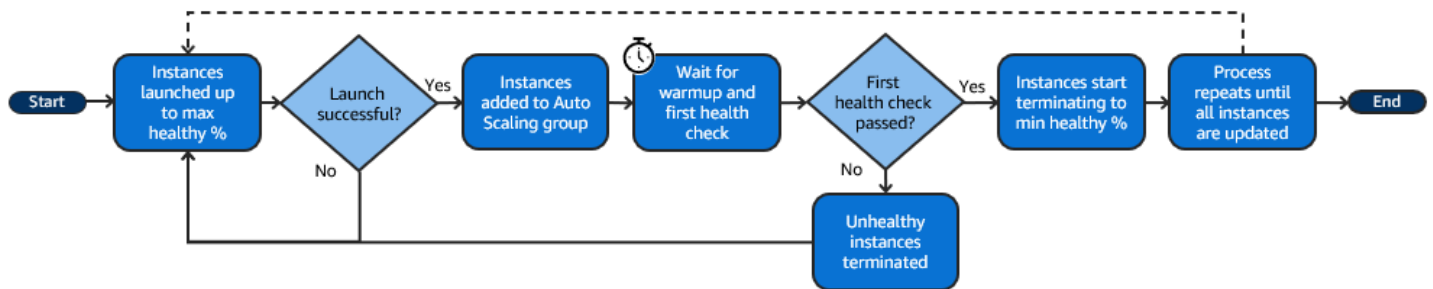
1. Buat template peluncuran baru atau perbarui template yang ada dengan perubahan konfigurasi yang diinginkan, seperti Amazon Machine Image (AMI) baru. Untuk informasi selengkapnya, lihat [Membuat template peluncuran untuk grup Auto Scaling](#).
2. Mulai penyegaran instans menggunakan konsol Auto Scaling Amazon EC2 AWS CLI, atau SDK:
 - Tentukan template peluncuran baru atau luncurkan versi template yang Anda buat. Ini akan digunakan untuk meluncurkan instance baru.
 - Tetapkan persentase sehat minimum dan maksimum yang disukai. Ini mengontrol berapa banyak instance yang diganti secara bersamaan dan apakah instance baru diluncurkan sebelum menghentikan yang lama.
 - Konfigurasi pengaturan opsional apa pun, seperti:
 - Checkpoints — Jeda penyegaran instans setelah persentase penggantian tertentu untuk memverifikasi kemajuan.
 - Lewati pencocokan — Bandingkan instance lama dengan konfigurasi baru dan hanya ganti yang tidak cocok. Saat Anda memulai penyegaran instans dari konsol, lewati pencocokan aktif secara default.
 - Beberapa jenis instans — Menerapkan [kebijakan instance campuran](#) baru atau yang diperbarui sebagai bagian dari konfigurasi yang diinginkan.

Saat penyegaran instans dimulai, Auto Scaling Amazon EC2 akan:

- Ganti instance dalam batch berdasarkan persentase sehat minimum dan maksimum.
- Luncurkan instance baru terlebih dahulu sebelum menghentikan yang lama jika persentase sehat minimum ditetapkan ke 100 persen. Ini memastikan bahwa kapasitas yang Anda inginkan dipertahankan setiap saat.
- Periksa contoh untuk status kesehatan dan beri mereka waktu untuk melakukan pemanasan sebelum lebih banyak contoh diganti.
- Mengakhiri dan mengganti contoh yang ditemukan tidak sehat.

- Secara otomatis memperbarui pengaturan grup Auto Scaling dengan perubahan konfigurasi baru setelah penyegaran instans berhasil.
- Jika grup Anda memiliki kolam hangat, Auto Scaling Amazon EC2 menggantikan instans terlebih dahulu. InService Kemudian menggantikan contoh di kolam hangat.

Diagram alir berikut menggambarkan perilaku peluncuran sebelum menghentikan saat Anda menetapkan persentase sehat minimum menjadi 100 persen.



Note

Persentase sehat minimum dan maksimum untuk penyegaran instans hanya perlu ditentukan jika Anda belum menetapkan kebijakan pemeliharaan instans, atau jika Anda perlu mengganti kebijakan yang ada. Untuk informasi selengkapnya, lihat [Kebijakan pemeliharaan instans](#).

Demikian pula, Anda hanya perlu menentukan periode pemanasan instance untuk penyegaran instance jika Anda belum mengaktifkan pemanasan default, atau jika Anda perlu mengganti default. Untuk informasi selengkapnya, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#).

Konsep inti

Sebelum memulai, biasakan diri Anda dengan konsep inti penyegaran contoh berikut:

Persentase sehat minimum

Persentase sehat minimum adalah persentase kapasitas yang diinginkan untuk tetap dalam layanan, sehat, dan siap digunakan selama penyegaran instance sehingga penyegaran dapat berlanjut. Misalnya, jika persentase sehat minimum adalah 90 persen dan persentase sehat maksimum adalah 100 persen, maka 10 persen kapasitas akan diganti sekaligus. Jika instans baru tidak lulus pemeriksaan kesehatan, Auto Scaling Amazon EC2 akan menghentikan dan

menggantikannya. Jika penyegaran instans tidak dapat meluncurkan instance sehat apa pun, pada akhirnya akan gagal, membuat 90 persen grup lainnya tidak tersentuh. Jika instans baru tetap sehat dan menyelesaikan periode pemanasan, Auto Scaling Amazon EC2 dapat terus menggantikan instans lain.

Penyegaran instance dapat menggantikan instance satu per satu, beberapa sekaligus, atau sekaligus. Untuk mengganti satu contoh pada satu waktu, tetapkan persentase sehat minimum dan maksimum menjadi 100 persen. Ini mengubah perilaku penyegaran instans untuk diluncurkan sebelum dihentikan, yang mencegah kapasitas grup turun di bawah 100 persen dari kapasitas yang diinginkan. Untuk mengganti semua instance sekaligus, tetapkan persentase sehat minimum 0 persen.

Persentase sehat maksimum

Persentase sehat maksimum adalah persentase kapasitas yang diinginkan yang dapat ditingkatkan oleh grup Auto Scaling Anda saat mengganti instance. Perbedaan antara minimum dan maksimum tidak boleh lebih besar dari 100. Rentang yang lebih besar meningkatkan jumlah instance yang dapat diganti pada saat yang bersamaan.

Contoh pemanasan

Pemanasan instance adalah periode waktu dari saat status instance baru berubah menjadi InService saat dianggap telah selesai diinisialisasi. Selama penyegaran instans, jika instans lulus pemeriksaan kesehatannya, Auto Scaling Amazon EC2 tidak segera beralih ke penggantian instance berikutnya setelah menentukan bahwa instans yang baru diluncurkan sehat. Itu menunggu periode pemanasan sebelum beralih untuk mengganti instance berikutnya. Ini dapat membantu ketika aplikasi Anda masih memerlukan beberapa waktu inisialisasi sebelum menanggapi permintaan.

Instance warmup bekerja dengan cara yang sama seperti pemanasan instance default. Oleh karena itu, pertimbangan penskalaan yang sama berlaku. Untuk informasi selengkapnya, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#).

Konfigurasi yang diinginkan

Konfigurasi yang diinginkan adalah konfigurasi baru yang Anda inginkan untuk diterapkan Auto Scaling Amazon EC2 di seluruh grup Auto Scaling Anda. Misalnya, Anda dapat menentukan template peluncuran baru dan jenis instans baru untuk instance Anda. Selama penyegaran instans, Auto Scaling Amazon EC2 memperbarui grup Auto Scaling ke konfigurasi yang diinginkan. Jika peristiwa penskalaan terjadi selama penyegaran instans, Auto Scaling Amazon

EC2 meluncurkan instans baru dengan konfigurasi yang diinginkan, bukan pengaturan grup saat ini. Setelah penyegaran instans berhasil, Auto Scaling Amazon EC2 memperbarui setelan grup Auto Scaling untuk mencerminkan konfigurasi baru yang diinginkan yang Anda tentukan sebagai bagian dari penyegaran instans.

Lewati pencocokan

Lewati pencocokan memberi tahu Amazon EC2 Auto Scaling untuk mengabaikan instans yang sudah memiliki pembaruan terbaru Anda. Dengan cara ini, Anda tidak mengganti lebih banyak instance daripada yang Anda butuhkan. Ini sangat membantu ketika Anda ingin memastikan grup Auto Scaling Anda menggunakan versi tertentu dari template peluncuran Anda dan hanya menggantikan instance yang menggunakan versi berbeda.

Pos pemeriksaan

Pos pemeriksaan adalah titik waktu di mana penyegaran instance berhenti sejenak untuk jumlah waktu tertentu. Penyegaran instance dapat berisi beberapa pos pemeriksaan. Auto Scaling Amazon EC2 memancarkan peristiwa untuk setiap pos pemeriksaan. Oleh karena itu, Anda dapat menambahkan EventBridge aturan untuk mengirim acara ke target, seperti Amazon SNS, untuk diberi tahu saat pos pemeriksaan tercapai. Setelah pos pemeriksaan tercapai, Anda memiliki kesempatan untuk memverifikasi penyebaran Anda. Jika ada masalah yang teridentifikasi, Anda dapat membatalkan penyegaran instance atau memutarkannya kembali. Kemampuan untuk menyebarkan pembaruan secara bertahap adalah manfaat utama dari pos pemeriksaan. Jika Anda tidak menggunakan pos pemeriksaan, penggantian bergulir dilakukan terus menerus.

Untuk mempelajari lebih lanjut tentang semua setelan default yang dapat Anda konfigurasi saat memulai penyegaran instans, lihat [Memahami nilai default untuk penyegaran instance](#).

Masa tenggang pemeriksaan kesehatan

Auto Scaling Amazon EC2 menentukan apakah instans sehat berdasarkan status pemeriksaan kesehatan yang digunakan grup Auto Scaling Anda. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

Untuk memastikan bahwa pemeriksaan kesehatan ini dimulai sesegera mungkin, jangan atur tenggang pemeriksaan kesehatan kelompok terlalu tinggi, tetapi cukup tinggi untuk pemeriksaan kesehatan Elastic Load Balancing Anda untuk menentukan apakah target tersedia untuk menangani permintaan. Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).

Kompatibilitas tipe instans

Sebelum Anda mengubah jenis instans Anda, ada baiknya untuk memverifikasi bahwa itu berfungsi dengan template peluncuran Anda. Ini mengonfirmasi kompatibilitas dengan AMI yang Anda tentukan. Misalnya, katakanlah Anda meluncurkan instans asli Anda dari AMI paravirtual (PV), tetapi Anda ingin mengubah ke jenis instans generasi saat ini yang hanya didukung oleh AMI mesin virtual perangkat keras (HVM). Dalam hal ini, Anda harus menggunakan AMI HVM di template peluncuran Anda.

Untuk mengonfirmasi kompatibilitas jenis instans tanpa meluncurkan instance, gunakan perintah [run-instance](#) dengan `--dry-run` opsi, seperti yang ditunjukkan pada contoh berikut.

```
aws ec2 run-instances --launch-template LaunchTemplateName=my-template,Version='1' --dry-run
```

Untuk informasi tentang cara kompatibilitas ditentukan, lihat [Kompatibilitas untuk mengubah jenis instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Batasan

- Total durasi: Jumlah waktu maksimum penyegaran instans dapat terus mengganti instans secara aktif adalah 14 hari.
- Perbedaan perilaku khusus untuk grup tertimbang: Jika grup instans campuran dikonfigurasi dengan bobot instans yang lebih besar dari atau sama dengan kapasitas yang diinginkan grup, Auto Scaling Amazon EC2 dapat InService menggantikan semua instans sekaligus. Untuk menghindari situasi ini, ikuti rekomendasi dalam [Konfigurasi grup Auto Scaling untuk menggunakan bobot instance](#) topik. Tentukan kapasitas yang diinginkan yang lebih besar dari bobot terbesar Anda saat Anda menggunakan bobot dengan grup Auto Scaling Anda.
- Batas waktu satu jam: Ketika penyegaran instans tidak dapat melanjutkan penggantian karena menunggu untuk mengganti instans saat siaga atau dilindungi dari skala masuk, atau instans baru tidak lulus pemeriksaan kesehatannya, Auto Scaling Amazon EC2 terus mencoba lagi selama satu jam. Ini juga menyediakan pesan status untuk membantu Anda menyelesaikan masalah. Jika masalah berlanjut setelah satu jam, operasi gagal. Tujuannya adalah untuk memberikan waktu untuk pulih jika ada masalah sementara.
- Menerapkan kode melalui data pengguna: Lewati pencocokan tidak memeriksa perubahan kode yang diterapkan dari skrip data pengguna. Jika Anda menggunakan data pengguna untuk menarik kode baru dan menginstal pembaruan ini pada instance baru, sebaiknya Anda menonaktifkan

pencocokan lewati untuk memastikan bahwa semua instance menerima kode terbaru Anda, bahkan tanpa pembaruan versi templat peluncuran.

Memahami nilai default untuk penyegaran instance

Sebelum memulai penyegaran instans, Anda dapat menyesuaikan berbagai preferensi yang memengaruhi penyegaran instans. Beberapa default preferensi berbeda tergantung pada apakah Anda menggunakan konsol atau baris perintah (AWS CLI atau AWS SDK).

Tabel berikut mencantumkan nilai default untuk pengaturan penyegaran instance.

Pengaturan	AWS CLI atau AWS SDK	Konsol Auto Scaling Amazon EC2
CloudWatch alarm	Dinonaktifkan (null)	Nonaktif
Rollback otomatis	Dinonaktifkan (false)	Nonaktif
Pos pemeriksaan	Dinonaktifkan (false)	Nonaktif
Penundaan pos pemeriksaan	1 jam (3600detik)	1 jam
Contoh pemanasan	Instans default pemanasan , jika didefinisikan, atau masa tenggang pemeriksaan kesehatan sebaliknya.	Instans default pemanasan , jika didefinisikan, atau masa tenggang pemeriksaan kesehatan sebaliknya.
Persentase sehat maksimum	Bervariasi berdasarkan kebijakan pemeliharaan instans Anda. Jika tidak ada kebijakan pemeliharaan instance, defaultnya 100 persen (null).	Bervariasi berdasarkan kebijakan pemeliharaan instans Anda. Jika tidak ada kebijakan pemeliharaan instance, defaultnya 100 persen (null).
Persentase sehat minimum	Bervariasi berdasarkan kebijakan pemeliharaan instans Anda. Jika tidak ada kebijakan pemeliharaan	Bervariasi berdasarkan kebijakan pemeliharaan instans Anda. Jika tidak ada kebijakan pemeliharaan

Pengaturan	AWS CLI atau AWS SDK	Konsol Auto Scaling Amazon EC2
	instance, defaultnya adalah 90 persen.	instance, defaultnya adalah 90 persen.
Instans yang dilindungi skala dalam	Tunggu	Abaikan
Lewati pencocokan	Dinonaktifkan (false)	Diaktifkan
Contoh siaga	Tunggu	Abaikan

Deskripsi setiap pengaturan berikut:

CloudWatch alarm (**AlarmSpecification**)

Spesifikasi CloudWatch alarm. CloudWatch alarm dapat digunakan untuk mengidentifikasi masalah apa pun dan gagal operasi jika alarm masuk ke ALARM keadaan. Untuk informasi selengkapnya, lihat [Mulai penyegaran instance dengan auto rollback](#).

Rollback otomatis () **AutoRollback**

Mengontrol apakah Auto Scaling Amazon EC2 memutar kembali grup Auto Scaling ke konfigurasi sebelumnya jika penyegaran instans gagal. Untuk informasi selengkapnya, lihat [Batalkan perubahan dengan rollback](#).

Pos pemeriksaan () **CheckpointPercentages**

Mengontrol apakah Auto Scaling Amazon EC2 menggantikan instans secara bertahap. Ini berguna jika Anda perlu melakukan verifikasi pada instans Anda sebelum mengganti semua instance. Untuk informasi selengkapnya, lihat [Tambahkan pos pemeriksaan ke penyegaran instance](#).

Penundaan pos pemeriksaan () **CheckpointDelay**

Jumlah waktu, dalam hitungan detik, untuk menunggu setelah pos pemeriksaan sebelum melanjutkan. Untuk informasi selengkapnya, lihat [Tambahkan pos pemeriksaan ke penyegaran instance](#).

Contoh pemanasan () **InstanceWarmup**

Periode waktu, dalam hitungan detik, selama Auto Scaling Amazon EC2 menunggu hingga instans baru dianggap telah selesai diinisialisasi sebelum melanjutkan untuk mengganti instance berikutnya. Jika Anda telah mendefinisikan pemanasan instans default dengan benar untuk grup Auto Scaling, maka Anda tidak perlu mengubah pemanasan instance (kecuali jika Anda ingin mengganti default). Untuk informasi selengkapnya, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#).

Persentase sehat maksimum (**MaxHealthyPercentage**)

Persentase kapasitas yang diinginkan dari grup Auto Scaling yang dapat ditingkatkan oleh grup Anda saat mengganti instance.

Persentase sehat minimum (**MinHealthyPercentage**)

Persentase kapasitas yang diinginkan dari grup Auto Scaling yang harus dalam pelayanan, sehat, dan siap digunakan sebelum operasi dapat dilanjutkan.

Instans yang dilindungi skala dalam () **ScaleInProtectedInstances**

Mengontrol apa yang dilakukan Auto Scaling Amazon EC2 jika instans yang dilindungi dari penskalaan ditemukan. Untuk informasi lebih lanjut tentang contoh ini, lihat [Gunakan perlindungan skala dalam instance](#).

Auto Scaling Amazon EC2 menyediakan opsi berikut:

- Replace (**Refresh**) - Menggantikan instance yang dilindungi dari skala masuk.
- Ignore (**Ignore**) — Mengabaikan instance yang dilindungi dari skala masuk dan terus mengganti instance yang tidak dilindungi.
- Tunggu (**Wait**) — Tunggu satu jam sampai Anda menghapus perlindungan skala. Jika Anda tidak melakukannya, penyegaran instance gagal.

Lewati pencocokan (**SkipMatching**)

Mengontrol apakah Auto Scaling Amazon EC2 melewatkan penggantian instans yang sesuai dengan konfigurasi yang diinginkan. Jika tidak ada konfigurasi yang diinginkan yang ditentukan, maka konfigurasi tersebut akan melewatkan penggantian instance yang memiliki template peluncuran dan tipe instance yang sama dengan yang digunakan grup Auto Scaling sebelum penyegaran instance dimulai. Untuk informasi selengkapnya, lihat [Gunakan penyegaran instance dengan melewatkan pencocokan](#).

Contoh siaga () **StandbyInstances**

Mengontrol apa yang dilakukan Auto Scaling Amazon EC2 jika instance ditemukan dalam status. Standby Untuk informasi lebih lanjut tentang contoh ini, lihat [Hapus instans sementara dari grup Auto Scaling](#).

Auto Scaling Amazon EC2 menyediakan opsi berikut:

- Terminate (Terminate) - Mengakhiri instance yang ada di. Standby
- Ignore (Ignore) — Mengabaikan instance yang ada di Standby dan terus menggantikan instance yang berada dalam status. InService
- Tunggu (Wait) — Menunggu satu jam bagi Anda untuk mengembalikan instans ke layanan. Jika Anda tidak melakukannya, penyegaran instance gagal.

Mulai penyegaran instance

Important

Anda dapat memutar kembali penyegaran instance yang sedang berlangsung untuk membatalkan perubahan apa pun. Agar ini berfungsi, grup Auto Scaling harus memenuhi prasyarat untuk menggunakan rollback sebelum memulai penyegaran instance. Untuk informasi selengkapnya, lihat [Batalan perubahan dengan rollback](#).

Prosedur berikut membantu Anda memulai penyegaran instans menggunakan AWS Management Console atau AWS CLI.

Mulai penyegaran instance (konsol)

Jika ini adalah pertama kalinya Anda memulai penyegaran instans, menggunakan konsol akan membantu Anda memahami fitur dan opsi yang tersedia.

Mulai penyegaran instance di konsol (prosedur dasar)

Gunakan prosedur berikut jika sebelumnya Anda belum menetapkan [kebijakan instance campuran](#) untuk grup Auto Scaling Anda. Jika sebelumnya Anda telah menetapkan kebijakan instance campuran, lihat [Mulai penyegaran instance di konsol \(grup instance campuran\)](#) untuk memulai penyegaran instans.

Untuk memulai penyegaran instance

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Pada tab Penyegaran instans, di Penyegaran instans aktif, pilih Mulai penyegaran instans.
4. Untuk pengaturan Ketersediaan, lakukan hal berikut:
 - a. Misalnya metode penggantian:
 - Jika Anda belum menetapkan kebijakan pemeliharaan instans pada grup Auto Scaling, setelan default untuk metode penggantian Instance adalah Terminate dan launch. Ini adalah perilaku default lama untuk penyegaran instance.
 - Jika Anda menetapkan kebijakan pemeliharaan instans pada grup Auto Scaling, ini akan memberikan nilai default untuk metode penggantian Instance. Untuk mengganti kebijakan pemeliharaan instans, pilih Override. Penggantian hanya berlaku untuk penyegaran instance saat ini. Lain kali Anda memulai penyegaran instance, nilai-nilai ini disetel ulang ke default kebijakan pemeliharaan instans.

Prosedur berikut menjelaskan cara memperbarui metode penggantian instance.

- i. Pilih salah satu metode penggantian contoh berikut:
 - Luncurkan sebelum penghentian: Instance baru harus disediakan terlebih dahulu sebelum instance yang ada dapat dihentikan. Ini adalah pilihan yang baik untuk aplikasi yang mendukung ketersediaan daripada penghematan biaya.
 - Hentikan dan luncurkan: Instans baru disediakan pada saat yang sama instans Anda yang ada dihentikan. Ini adalah pilihan yang baik untuk aplikasi yang mendukung penghematan biaya daripada ketersediaan. Ini juga merupakan pilihan yang baik untuk aplikasi yang seharusnya tidak meluncurkan kapasitas lebih dari yang tersedia saat ini.
 - Perilaku kustom: Opsi ini memungkinkan Anda mengatur rentang minimum dan maksimum kustom untuk jumlah kapasitas yang Anda inginkan tersedia saat mengganti instance. Ini dapat membantu Anda mencapai keseimbangan yang tepat antara biaya dan ketersediaan.

- ii. Untuk Tetapkan persentase sehat, masukkan nilai untuk satu atau kedua bidang berikut. Bidang aktifkan bervariasi tergantung pada opsi yang Anda pilih untuk metode penggantian Instance.
 - Min: Menetapkan persentase sehat minimum yang diperlukan untuk melanjutkan penyegaran instance.
 - Max: Menetapkan persentase sehat maksimum yang mungkin selama penyegaran instans.
- iii. Perluas bagian Lihat perkiraan kapasitas sementara selama penggantian berdasarkan ukuran grup saat ini untuk mengonfirmasi bagaimana nilai untuk Min dan Maks berlaku untuk grup Anda. Nilai pasti yang digunakan tergantung pada nilai kapasitas yang diinginkan, yang akan berubah jika skala grup.
- iv. Perluas bagian Setel perilaku fallback untuk ukuran penggantian yang tidak valid, lalu pilih apakah akan Melanggar persentase sehat maksimal untuk memprioritaskan ketersediaan, atau Melanggar persentase sehat min.

Menjaga opsi persentase Melanggar min sehat default tidak disarankan untuk grup yang sangat kecil. Jika hanya ada satu instance dalam grup Auto Scaling, memulai penyegaran instance dapat mengakibatkan pemadaman.

Langkah ini mengonfigurasi perilaku fallback jika Anda menggunakan grup Auto Scaling yang belum memiliki kebijakan pemeliharaan instance. Opsi ini tidak tersedia dan tidak muncul ketika grup Anda memiliki kebijakan pemeliharaan instans. Opsi ini juga hanya tersedia untuk metode penggantian Terminate dan luncurkan. Metode penggantian lainnya akan melanggar persentase sehat maksimum untuk memprioritaskan ketersediaan.

- b. Untuk pemanasan Instance, masukkan jumlah detik dari saat status instance baru berubah menjadi saat selesai InService diinisialisasi. Auto Scaling Amazon EC2 menunggu jumlah waktu ini sebelum melanjutkan untuk mengganti instance berikutnya.

Saat melakukan pemanasan, instance yang baru diluncurkan juga tidak dihitung dalam metrik instans agregat dari grup Auto Scaling (seperti `CPUUtilization`, dan). `NetworkIn` `NetworkOut` Jika Anda menambahkan kebijakan penskalaan ke grup Auto Scaling, aktivitas penskalaan berjalan secara paralel. Jika Anda menyetel interval panjang untuk periode pemanasan penyegaran instans, dibutuhkan lebih banyak waktu agar instans yang baru diluncurkan muncul dalam metrik. Oleh karena itu, periode pemanasan yang memadai membuat Amazon EC2 Auto Scaling tidak melakukan penskalaan pada data metrik basi.

Jika Anda telah mendefinisikan pemanasan instance default dengan benar untuk grup Auto Scaling, maka Anda tidak perlu mengubah pemanasan instance. Namun, jika Anda ingin mengganti default, Anda dapat menetapkan nilai untuk opsi ini. Untuk informasi selengkapnya tentang menyetel pemanasan instance default, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#)

5. Untuk pengaturan Refresh, lakukan hal berikut:


- a. (Opsional) Untuk Pos Pemeriksaan, pilih Aktifkan pos pemeriksaan untuk mengganti instance menggunakan pendekatan bertahap atau bertahap untuk penyegaran instans. Ini memberikan waktu tambahan untuk verifikasi antara set penggantian. Jika Anda memilih untuk tidak mengaktifkan pos pemeriksaan, instance diganti dalam satu operasi yang hampir terus menerus.

Jika Anda mengaktifkan pos pemeriksaan, lihat [Aktifkan pos pemeriksaan \(konsol\)](#) langkah-langkah tambahan.

b. Mengaktifkan atau menonaktifkan Lewati pencocokan:

- Untuk melewati penggantian instance yang sudah cocok dengan template peluncuran Anda, pilih kotak centang Aktifkan lompat pencocokan.
- Jika Anda menonaktifkan pencocokan lewati dengan membersihkan kotak centang ini, semua instance dapat diganti.

Saat Anda mengaktifkan pencocokan lewati, Anda dapat mengatur templat peluncuran baru atau versi baru dari templat peluncuran alih-alih menggunakan yang sudah ada. Lakukan ini di bagian konfigurasi yang diinginkan pada halaman penyegaran instans Mulai.

 Note

Untuk menggunakan fitur skip matching untuk memperbarui grup Auto Scaling yang saat ini menggunakan konfigurasi peluncuran, Anda harus memilih template peluncuran dalam konfigurasi yang diinginkan. Lewati pencocokan dengan konfigurasi peluncuran tidak didukung.

- c. Untuk instance Siaga, pilih Abaikan, Hentikan, atau Tunggu. Ini menentukan apa yang terjadi jika instance ditemukan dalam Standby keadaan. Untuk informasi selengkapnya, lihat [Hapus instans sementara dari grup Auto Scaling](#).

Jika Anda memilih Tunggu, Anda harus mengambil langkah-langkah tambahan untuk mengembalikan instans ini ke layanan. Jika tidak, penyegaran instance menggantikan semua InService instance dan menunggu selama satu jam. Kemudian, jika ada Standby instance yang tersisa, penyegaran instance gagal. Untuk mencegah situasi ini, pilih untuk Abaikan atau Hentikan instance ini sebagai gantinya.

- d. Untuk instance yang dilindungi Scale-in, pilih Abaikan, Ganti, atau Tunggu. Ini menentukan apa yang terjadi jika instance yang dilindungi scale-in ditemukan. Untuk informasi selengkapnya, lihat [Gunakan perlindungan skala dalam instance](#).

Jika Anda memilih Tunggu, Anda harus mengambil langkah tambahan untuk menghapus perlindungan skala dalam dari instance ini. Jika tidak, penyegaran instance menggantikan semua instance yang tidak dilindungi dan menunggu satu jam. Kemudian, jika ada instance yang dilindungi skala dalam tetap ada, penyegaran instance gagal. Untuk mencegah situasi ini, pilih untuk Abaikan atau Ganti instance ini sebagai gantinya.

6. (Opsional) Untuk CloudWatch alarm, pilih Aktifkan CloudWatch alarm, lalu pilih satu atau beberapa alarm. CloudWatch alarm dapat digunakan untuk mengidentifikasi masalah apa pun dan gagal operasi jika alarm masuk ke ALARM keadaan. Untuk informasi selengkapnya, lihat [Mulai penyegaran instance dengan auto rollback](#).
7. (Opsional) Perluas bagian konfigurasi yang diinginkan untuk menentukan pembaruan yang ingin Anda buat ke grup Auto Scaling Anda.


Untuk langkah ini, Anda dapat memilih untuk menggunakan sintaks JSON atau YAMAL untuk mengedit nilai parameter daripada membuat pilihan di antarmuka konsol. Untuk melakukannya, pilih Gunakan editor kode alih-alih Gunakan antarmuka konsol. Prosedur berikut menjelaskan cara membuat pilihan menggunakan antarmuka konsol.

- a. Untuk template peluncuran Pembaruan:

- Jika Anda belum membuat template peluncuran baru atau versi template peluncuran baru untuk grup Auto Scaling Anda, jangan pilih kotak centang ini.
- Jika Anda membuat template peluncuran baru atau versi template peluncuran baru, pilih kotak centang ini. Ketika Anda memilih opsi ini, Amazon EC2 Auto Scaling menunjukkan template peluncuran saat ini dan versi template peluncuran saat ini. Ini juga mencantumkan versi lain yang tersedia. Pilih template peluncuran dan kemudian pilih versinya.


Setelah Anda memilih versi, Anda dapat melihat informasi versi. Ini adalah versi template peluncuran yang akan digunakan saat mengganti instance sebagai bagian dari penyegaran instance. Jika penyegaran instans berhasil, versi template peluncuran ini juga akan digunakan setiap kali instance baru diluncurkan, seperti saat grup menskalakan.

- b. Untuk Pilih satu set jenis instans dan opsi pembelian untuk mengganti jenis instance di template peluncuran:
- Jangan pilih kotak centang ini jika Anda ingin menggunakan jenis instans dan opsi pembelian yang Anda tentukan dalam template peluncuran Anda.
 - Pilih kotak centang ini jika Anda ingin mengganti jenis instance di template peluncuran atau menjalankan Instans Spot. Anda dapat menambahkan setiap jenis instans secara manual atau memilih jenis instans utama dan opsi rekomendasi yang mengambil jenis instans tambahan yang cocok untuk Anda. Jika Anda berencana untuk meluncurkan Instans Spot, sebaiknya tambahkan beberapa jenis instans yang berbeda. Dengan cara ini, Auto Scaling Amazon EC2 dapat meluncurkan jenis instans lain jika kapasitas instans tidak mencukupi di Availability Zone yang Anda pilih. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).

 Warning

Jangan gunakan Instans Spot dengan aplikasi yang tidak dapat menangani interupsi Instans Spot. Gangguan dapat terjadi jika layanan Amazon EC2 Spot perlu merebut kembali kapasitas.

Jika Anda memilih kotak centang ini, pastikan bahwa template peluncuran belum meminta Instans Spot. Anda tidak dapat menggunakan template peluncuran yang meminta Instans Spot untuk membuat grup Auto Scaling yang menggunakan beberapa jenis instans dan meluncurkan Instans Spot dan Sesuai Permintaan.

 Note

Untuk mengonfigurasi opsi ini pada grup Auto Scaling yang saat ini menggunakan konfigurasi peluncuran, Anda harus memilih templat peluncuran di Template peluncuran Perbarui. Mengganti jenis instans dalam konfigurasi peluncuran Anda tidak didukung.

8. (Opsional) Untuk pengaturan Rollback, pilih Aktifkan rollback otomatis untuk secara otomatis memutar kembali penyegaran instance jika gagal.

Pengaturan ini hanya dapat diaktifkan jika grup Auto Scaling memenuhi prasyarat untuk menggunakan rollback.

Untuk informasi selengkapnya, lihat [Batalan perubahan dengan rollback](#).

9. Tinjau semua pilihan Anda untuk mengonfirmasi bahwa semuanya sudah diatur dengan benar.

Pada titik ini, ada baiknya untuk memverifikasi bahwa perbedaan antara perubahan saat ini dan yang diusulkan tidak akan memengaruhi aplikasi Anda dengan cara yang tidak terduga atau tidak diinginkan. Untuk mengonfirmasi bahwa jenis instans Anda kompatibel dengan template peluncuran, lihat [Kompatibilitas tipe instans](#).

10. Jika Anda puas dengan pilihan penyegaran instans, pilih Mulai penyegaran instans.

Mulai penyegaran instance di konsol (grup instance campuran)

Gunakan prosedur berikut jika Anda telah membuat grup Auto Scaling dengan kebijakan [instance campuran](#). Jika Anda belum menetapkan kebijakan instance campuran untuk grup Anda, lihat [Mulai penyegaran instance di konsol \(prosedur dasar\)](#) untuk memulai penyegaran instance.

Untuk memulai penyegaran instance

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Pada tab Penyegaran instans, di Penyegaran instans aktif, pilih Mulai penyegaran instans.

4. Untuk pengaturan Ketersediaan, lakukan hal berikut:

- a. Misalnya metode penggantian:

- Jika Anda belum menetapkan kebijakan pemeliharaan instans pada grup Auto Scaling, setelah default untuk metode penggantian Instance adalah Terminate dan launch. Ini adalah perilaku default lama untuk penyegaran instance.
- Jika Anda menetapkan kebijakan pemeliharaan instans pada grup Auto Scaling, ini akan memberikan nilai default untuk metode penggantian Instance. Untuk mengganti kebijakan

pemeliharaan instans, pilih Override. Penggantian hanya berlaku untuk penyegaran instance saat ini. Lain kali Anda memulai penyegaran instance, nilai-nilai ini disetel ulang ke default kebijakan pemeliharaan instans.

Prosedur berikut menjelaskan cara memperbarui metode penggantian instance.

- i. Pilih salah satu metode penggantian contoh berikut:
 - Luncurkan sebelum penghentian: Instance baru harus disediakan terlebih dahulu sebelum instance yang ada dapat dihentikan. Ini adalah pilihan yang baik untuk aplikasi yang mendukung ketersediaan daripada penghematan biaya.
 - Hentikan dan luncurkan: Instans baru disediakan pada saat yang sama instans Anda yang ada dihentikan. Ini adalah pilihan yang baik untuk aplikasi yang mendukung penghematan biaya daripada ketersediaan. Ini juga merupakan pilihan yang baik untuk aplikasi yang seharusnya tidak meluncurkan kapasitas lebih dari yang tersedia saat ini.
 - Perilaku kustom: Opsi ini memungkinkan Anda mengatur rentang minimum dan maksimum kustom untuk jumlah kapasitas yang Anda inginkan tersedia saat mengganti instance. Ini dapat membantu Anda mencapai keseimbangan yang tepat antara biaya dan ketersediaan.
- ii. Untuk Tetapkan persentase sehat, masukkan nilai untuk satu atau kedua bidang berikut. Bidang aktifkan bervariasi tergantung pada opsi yang Anda pilih untuk metode penggantian Instance.
 - Min: Menetapkan persentase sehat minimum yang diperlukan untuk melanjutkan penyegaran instance.
 - Max: Menetapkan persentase sehat maksimum yang mungkin selama penyegaran instans.
- iii. Perluas bagian Lihat perkiraan kapasitas sementara selama penggantian berdasarkan ukuran grup saat ini untuk mengonfirmasi bagaimana nilai untuk Min dan Maks berlaku untuk grup Anda. Nilai pasti yang digunakan tergantung pada nilai kapasitas yang diinginkan, yang akan berubah jika skala grup.
- iv. Perluas bagian Setel perilaku fallback untuk ukuran penggantian yang tidak valid, lalu pilih apakah akan Melanggar persentase sehat maksimal untuk memprioritaskan ketersediaan, atau Melanggar persentase sehat min.

Menjaga opsi persentase Melanggar min sehat default tidak disarankan untuk grup yang sangat kecil. Jika hanya ada satu instance dalam grup Auto Scaling, memulai penyegaran instance dapat mengakibatkan pemadaman.

Langkah ini mengonfigurasi perilaku fallback jika Anda menggunakan grup Auto Scaling yang belum memiliki kebijakan pemeliharaan instance. Opsi ini tidak tersedia dan tidak muncul ketika grup Anda memiliki kebijakan pemeliharaan instans. Opsi ini juga hanya tersedia untuk metode penggantian Terminate dan luncurkan. Metode penggantian lainnya akan melanggar persentase sehat maksimum untuk memprioritaskan ketersediaan.

- b. Untuk pemanasan Instance, masukkan jumlah detik dari saat status instance baru berubah menjadi saat selesai InService diinisialisasi. Auto Scaling Amazon EC2 menunggu jumlah waktu ini sebelum melanjutkan untuk mengganti instance berikutnya.

Saat melakukan pemanasan, instance yang baru diluncurkan juga tidak dihitung dalam metrik instans agregat dari grup Auto Scaling (sepertiCPUUtilization,, dan). NetworkIn NetworkOut Jika Anda menambahkan kebijakan penskalaan ke grup Auto Scaling, aktivitas penskalaan berjalan secara paralel. Jika Anda menyetel interval panjang untuk periode pemanasan penyegaran instans, dibutuhkan lebih banyak waktu agar instans yang baru diluncurkan muncul dalam metrik. Oleh karena itu, periode pemanasan yang memadai membuat Amazon EC2 Auto Scaling tidak melakukan penskalaan pada data metrik basi.

Jika Anda telah mendefinisikan pemanasan instance default dengan benar untuk grup Auto Scaling, maka Anda tidak perlu mengubah pemanasan instance. Namun, jika Anda ingin mengganti default, Anda dapat menetapkan nilai untuk opsi ini. Untuk informasi selengkapnya tentang menyetel pemanasan instance default, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#)

5. Untuk pengaturan Refresh, lakukan hal berikut:

- a. (Opsional) Untuk Pos Pemeriksaan, pilih Aktifkan pos pemeriksaan untuk mengganti instance menggunakan pendekatan bertahap atau bertahap untuk penyegaran instans. Ini memberikan waktu tambahan untuk verifikasi antara set penggantian. Jika Anda memilih untuk tidak mengaktifkan pos pemeriksaan, instance diganti dalam satu operasi yang hampir terus menerus.

Jika Anda mengaktifkan pos pemeriksaan, lihat [Aktifkan pos pemeriksaan \(konsol\)](#) langkah-langkah tambahan.

b. Mengaktifkan atau menonaktifkan Lewati pencocokan:

- Untuk melewati penggantian instance yang sudah cocok dengan template peluncuran Anda dan penggantian jenis instance apa pun, pilih kotak centang Aktifkan lompatan pencocokan.
- Jika Anda memilih untuk menonaktifkan pencocokan lewati dengan membersihkan kotak centang ini, semua instance dapat diganti.

Saat Anda mengaktifkan pencocokan lewati, Anda dapat mengatur templat peluncuran baru atau versi baru dari templat peluncuran alih-alih menggunakan yang sudah ada. Lakukan ini di bagian konfigurasi yang diinginkan pada halaman penyegaran instans Mulai. Anda juga dapat memperbarui penggantian jenis instans Anda dalam konfigurasi yang Diinginkan.

c. Untuk instance Siaga, pilih Abaikan, Hentikan, atau Tunggu. Ini menentukan apa yang terjadi jika instance ditemukan dalam Standby keadaan. Untuk informasi selengkapnya, lihat [Hapus instans sementara dari grup Auto Scaling](#).

Jika Anda memilih Tunggu, Anda harus mengambil langkah-langkah tambahan untuk mengembalikan instans ini ke layanan. Jika tidak, penyegaran instance menggantikan semua InService instance dan menunggu satu jam. Kemudian, jika ada Standby instance yang tersisa, penyegaran instance gagal. Untuk mencegah situasi ini, pilih untuk Abaikan atau Hentikan instance ini sebagai gantinya.

d. Untuk instance yang dilindungi Scale-in, pilih Abaikan, Ganti, atau Tunggu. Ini menentukan apa yang terjadi jika instance yang dilindungi scale-in ditemukan. Untuk informasi selengkapnya, lihat [Gunakan perlindungan skala dalam instance](#).

Jika Anda memilih Tunggu, Anda harus mengambil langkah tambahan untuk menghapus perlindungan skala dalam dari instance ini. Jika tidak, penyegaran instance menggantikan semua instance yang tidak dilindungi dan menunggu satu jam. Kemudian, jika ada instance yang dilindungi skala dalam tetap ada, penyegaran instance gagal. Untuk mencegah situasi ini, pilih untuk Abaikan atau Ganti instance ini sebagai gantinya.

6. (Opsional) Untuk CloudWatch alarm, pilih Aktifkan CloudWatch alarm, lalu pilih satu atau beberapa alarm. CloudWatch alarm dapat digunakan untuk mengidentifikasi masalah apa pun dan gagal operasi jika alarm masuk ke ALARM keadaan. Untuk informasi selengkapnya, lihat [Mulai penyegaran instance dengan auto rollback](#).

7. Di bagian konfigurasi yang diinginkan, lakukan hal berikut.

Untuk langkah ini, Anda dapat memilih untuk menggunakan sintaks JSON atau YAMAL untuk mengedit nilai parameter daripada membuat pilihan di antarmuka konsol. Untuk melakukannya, pilih Gunakan editor kode alih-alih Gunakan antarmuka konsol. Prosedur berikut menjelaskan cara membuat pilihan menggunakan antarmuka konsol.

a. Untuk template peluncuran Pembaruan:

- Jika Anda belum membuat template peluncuran baru atau versi template peluncuran baru untuk grup Auto Scaling Anda, jangan pilih kotak centang ini.
- Jika Anda membuat template peluncuran baru atau versi template peluncuran baru, pilih kotak centang ini. Ketika Anda memilih opsi ini, Amazon EC2 Auto Scaling menunjukkan template peluncuran saat ini dan versi template peluncuran saat ini. Ini juga mencantumkan versi lain yang tersedia. Pilih template peluncuran dan kemudian pilih versinya.

Setelah Anda memilih versi, Anda dapat melihat informasi versi. Ini adalah versi template peluncuran yang akan digunakan saat mengganti instance sebagai bagian dari penyegaran instance. Jika penyegaran instans berhasil, versi template peluncuran ini juga akan digunakan setiap kali instance baru diluncurkan, seperti saat grup menskalakan.

b. Untuk Gunakan pengaturan ini untuk mengganti jenis instance dan opsi pembelian yang ditentukan dalam template peluncuran:

Secara default, kotak centang ini dipilih. Auto Scaling Amazon EC2 mengisi setiap parameter dengan nilai yang saat ini ditetapkan dalam kebijakan instans campuran untuk grup Auto Scaling. Hanya perbarui nilai untuk parameter yang ingin Anda ubah. Untuk panduan tentang pengaturan ini, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).

 Warning

Kami menyarankan Anda untuk tidak menghapus kotak centang ini. Hapus hanya jika Anda ingin berhenti menggunakan kebijakan instance campuran. Setelah penyegaran instans berhasil, Auto Scaling Amazon EC2 memperbarui grup Anda agar sesuai dengan konfigurasi yang diinginkan. Jika tidak lagi menyertakan kebijakan instans campuran, Auto Scaling Amazon EC2 secara bertahap menghentikan Instans Spot yang sedang berjalan dan menggantinya dengan Instans Sesuai Permintaan. Atau, jika template peluncuran Anda meminta

Instans Spot, Auto Scaling Amazon EC2 secara bertahap menghentikan Instans Sesuai Permintaan yang sedang berjalan dan menggantinya dengan Instans Spot.

8. (Opsional) Untuk pengaturan Rollback, pilih Aktifkan rollback otomatis untuk secara otomatis memutar kembali penyegaran instance jika gagal karena alasan apa pun.

Pengaturan ini hanya dapat diaktifkan jika grup Auto Scaling memenuhi prasyarat untuk menggunakan rollback.

Untuk informasi selengkapnya, lihat [Batalan perubahan dengan rollback](#).

9. Tinjau semua pilihan Anda untuk mengonfirmasi bahwa semuanya sudah diatur dengan benar.

Pada titik ini, ada baiknya untuk memverifikasi bahwa perbedaan antara perubahan saat ini dan yang diusulkan tidak akan memengaruhi aplikasi Anda dengan cara yang tidak terduga atau tidak diinginkan. Untuk mengonfirmasi bahwa jenis instans Anda kompatibel dengan template peluncuran, lihat [Kompatibilitas tipe instans](#).

Jika Anda puas dengan pilihan penyegaran instans, pilih Mulai penyegaran instans.

Mulai refresh instance (AWS CLI)

Untuk memulai penyegaran instance

Gunakan [start-instance-refresh](#) perintah berikut untuk memulai penyegaran instance dari AWS CLI. Anda dapat menentukan preferensi apa pun yang ingin Anda ubah dalam file konfigurasi JSON. Saat Anda mereferensikan file konfigurasi, berikan path file dan nama seperti yang ditunjukkan pada contoh berikut.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Isi dari config.json:

```
{
  "AutoScalingGroupName": "my-asg",
  "Preferences": {
    "InstanceWarmup": 60,
    "MinHealthyPercentage": 50,
    "AutoRollback": true,
    "ScaleInProtectedInstances": Ignore,
    "StandbyInstances": Terminate
  }
}
```

```
}  
}
```

Jika preferensi tidak disediakan, nilai default digunakan. Untuk informasi selengkapnya, lihat [Memahami nilai default untuk penyegaran instance](#).

Contoh output:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Pantau penyegaran instance

Anda dapat memantau penyegaran instans yang sedang berlangsung atau mencari status penyegaran instans sebelumnya dari enam minggu terakhir menggunakan atau. AWS Management Console AWS CLI

Pantau dan periksa status penyegaran instans

Untuk memantau dan memeriksa status penyegaran instans, gunakan salah satu metode berikut:

Console

Tip

Dalam prosedur ini, kolom bernama seharusnya sudah ditampilkan. Untuk menampilkan kolom tersembunyi atau mengubah jumlah baris yang ditampilkan, pilih ikon roda gigi di sudut kanan atas bagian untuk membuka modal preferensi. Perbarui pengaturan sesuai kebutuhan dan pilih Konfirmasi.

Untuk memantau dan memeriksa status penyegaran instans (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Penyegaran instans, di bawah Riwayat penyegaran instans, Anda dapat menentukan status permintaan dengan melihat kolom Status. Operasi masuk ke Pending status saat sedang diinisialisasi. Status kemudian harus dengan cepat berubah menjadiInProgress. Ketika semua instance diperbarui, status berubah menjadiSuccessful.
4. Anda dapat memantau lebih lanjut keberhasilan atau kegagalan aktivitas yang sedang berlangsung dengan melihat aktivitas penskalaan grup. Di tab Aktivitas, di bawah Riwayat aktivitas, saat refresh instans dimulai, Anda melihat entri saat instans dihentikan dan set entri lain saat instans diluncurkan. Jika Anda memiliki banyak aktivitas penskalaan, Anda dapat melihat lebih banyak aktivitas dengan memilih ikon > di bagian atas riwayat aktivitas. Untuk informasi tentang masalah pemecahan masalah yang mungkin menyebabkan aktivitas gagal, lihat [Memecahkan Masalah Auto Scaling Amazon EC2](#)
5. (Opsional) Pada tab Manajemen instans, di bawah Instans, Anda dapat meninjau kemajuan instance tertentu sesuai kebutuhan.

AWS CLI

Untuk memantau dan memeriksa status instans refresh (AWS CLI)

Gunakan perintah [describe-instance-refreshes](#) berikut ini.

```
aws autoscaling describe-instance-refreshes --auto-scaling-group-name my-asg
```

Berikut ini adalah output contoh.

Penyegaran instance diurutkan berdasarkan waktu mulai. Penyegaran instance yang masih berlangsung dijelaskan terlebih dahulu.

```
{
  "InstanceRefreshes": [
    {
      "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b",
      "AutoScalingGroupName": "my-asg",
      "Status": "InProgress",
      "StatusReason": "Waiting for instances to warm up before continuing. For example: i-0645704820a8e83ff is warming up.",
      "StartTime": "2023-11-24T16:46:52+00:00",
      "PercentageComplete": 50,
      "InstancesToUpdate": 0,
    }
  ]
}
```

```
"Preferences":{
  "MaxHealthyPercentage":120,
  "MinHealthyPercentage":90,
  "InstanceWarmup":60,
  "SkipMatching":false,
  "AutoRollback":true,
  "ScaleInProtectedInstances":"Ignore",
  "StandbyInstances":"Ignore"
},
{
  "InstanceRefreshId":"0e151305-1e57-4a32-a256-1fd14157c5ec",
  "AutoScalingGroupName":"my-asg",
  "Status":"Successful",
  "StartTime":"2023-11-22T13:53:37+00:00",
  "EndTime":"2023-11-22T13:59:45+00:00",
  "PercentageComplete":100,
  "InstancesToUpdate":0,
  "Preferences":{
    "MaxHealthyPercentage":120,
    "MinHealthyPercentage":90,
    "InstanceWarmup":60,
    "SkipMatching":false,
    "AutoRollback":true,
    "ScaleInProtectedInstances":"Ignore",
    "StandbyInstances":"Ignore"
  }
}
]
```

Anda dapat memantau lebih lanjut keberhasilan atau kegagalan aktivitas yang sedang berlangsung dengan melihat aktivitas penskalaan grup. Aktivitas penskalaan juga membantu Anda menelusuri detail selengkapnya untuk membantu Anda memecahkan masalah dengan penyegaran instans. Untuk informasi selengkapnya, lihat [Memecahkan Masalah Auto Scaling Amazon EC2](#).

Status penyegaran instance

Ketika Anda memulai penyegaran instans, itu memasuki status Pending. Ini berpindah dari Tertunda InProgress hingga mencapai Sukses, Gagal, Dibatalkan RollbackSuccessful, atau RollbackFailed.

Penyegaran instance dapat memiliki status berikut:

Status	Deskripsi
Tertunda	Permintaan dibuat, tetapi penyegaran instance belum dimulai.
InProgress	Penyegaran instance sedang berlangsung.
Sukses	Penyegaran instance berhasil diselesaikan.
Failed	Penyegaran instance gagal diselesaikan. Anda dapat memecahkan masalah menggunakan alasan status dan aktivitas penskalaan.
Membatalkan	Penyegaran instance yang sedang berlangsung dibatalkan.
Dibatalkan	Penyegaran instance dibatalkan.
RollbackInProgress	Penyegaran instance sedang diputar kembali.
RollbackFailed	Rollback gagal selesai. Anda dapat memecahkan masalah menggunakan alasan status dan aktivitas penskalaan.
RollbackSuccessful	Rollback selesai dengan sukses.

Batalkan penyegaran instance

Anda dapat membatalkan penyegaran instance yang masih berlangsung. Anda tidak dapat membatalkannya setelah selesai.

Membatalkan penyegaran instance tidak mengembalikan instance apa pun yang sudah diganti. Untuk memutar kembali perubahan ke instance Anda, lakukan rollback sebagai gantinya. Untuk informasi selengkapnya, lihat [Batalkan perubahan dengan rollback](#).

Topik

- [Batalkan penyegaran instance \(konsol\)](#)
- [Batalkan penyegaran instance \(AWS CLI\)](#)

Batalan penyegaran instance (konsol)

Untuk membatalkan penyegaran instans

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.
3. Pada tab Penyegaran instans, di Penyegaran instans aktif, pilih Tindakan, Batal.
4. Ketika diminta untuk mengonfirmasi, pilih Konfirmasi.

Status penyegaran instans diatur ke Membatalkan. Setelah pembatalan selesai, status penyegaran instans diatur ke Dibatalkan.

Batalan penyegaran instance (AWS CLI)

Untuk membatalkan penyegaran instans

Gunakan [cancel-instance-refresh](#) perintah dari AWS CLI dan berikan nama grup Auto Scaling.

```
aws autoscaling cancel-instance-refresh --auto-scaling-group-name my-asg
```

Contoh output:

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Batalan perubahan dengan rollback

Anda dapat memutar kembali penyegaran instance yang masih berlangsung. Anda tidak dapat menggulungnya kembali setelah selesai. Namun, Anda dapat memperbarui grup Auto Scaling lagi dengan memulai penyegaran instans baru.

Saat memutar kembali, Amazon EC2 Auto Scaling menggantikan instans yang telah digunakan sejauh ini. Instance baru cocok dengan konfigurasi yang terakhir Anda simpan di grup Auto Scaling sebelum memulai penyegaran instans.

Auto Scaling Amazon EC2 menyediakan cara-cara berikut untuk memutar kembali:

- Rollback manual: Anda memulai rollback secara manual untuk membalikkan apa yang digunakan hingga titik rollback.
- Rollback otomatis: Auto Scaling Amazon EC2 secara otomatis membalikkan apa yang diterapkan jika penyegaran instans gagal karena alasan tertentu atau CloudWatch jika ada alarm yang Anda tentukan masuk ke status. ALARM

Daftar Isi

- [Pertimbangan](#)
- [Mulai rollback secara manual](#)
- [Mulai penyegaran instance dengan auto rollback](#)

Pertimbangan

Pertimbangan berikut berlaku saat menggunakan rollback:

- Opsi rollback hanya tersedia jika Anda menentukan konfigurasi yang diinginkan sebagai bagian dari memulai penyegaran instance.
- Anda hanya dapat memutar kembali ke versi sebelumnya dari template peluncuran jika versi tersebut adalah versi bernomor tertentu. Opsi rollback tidak tersedia jika grup Auto Scaling dikonfigurasi untuk menggunakan `$Default` atau meluncurkan versi `$Latest` template.
- Anda juga tidak dapat memutar kembali ke template peluncuran yang dikonfigurasi untuk menggunakan alias AMI dari AWS Systems Manager Parameter Store.
- Konfigurasi yang terakhir Anda simpan di grup Auto Scaling harus dalam keadaan stabil. Jika tidak dalam keadaan stabil, alur kerja rollback akan tetap terjadi, tetapi pada akhirnya akan gagal. Sampai Anda menyelesaikan masalah, grup Auto Scaling mungkin berada dalam status gagal di mana grup Auto Scaling tidak dapat lagi berhasil meluncurkan instans. Hal ini dapat mempengaruhi ketersediaan layanan atau aplikasi.

Mulai rollback secara manual

Console

Untuk memulai rollback penyegaran instance (konsol) secara manual

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.

2. Pilih kotak centang di samping grup Auto Scaling.
3. Pada tab Penyegaran instans, di Penyegaran instans aktif, pilih Tindakan, Mulai kembali.
4. Ketika diminta untuk mengonfirmasi, pilih Konfirmasi.

AWS CLI

Untuk memulai rollback instance refresh () secara manual AWS CLI

Gunakan [rollback-instance-refresh](#) perintah dari AWS CLI dan berikan nama grup Auto Scaling.

```
aws autoscaling rollback-instance-refresh --auto-scaling-group-name my-asg
```

Contoh output:

```
{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

Tip

Jika perintah ini menimbulkan kesalahan, pastikan Anda telah memperbarui AWS CLI secara lokal ke versi terbaru.

Mulai penyegaran instance dengan auto rollback

Menggunakan fitur auto rollback, Anda dapat secara otomatis memutar kembali penyegaran instans ketika gagal, seperti ketika ada kesalahan atau CloudWatch alarm Amazon tertentu masuk ke status.

ALARM

Jika Anda mengaktifkan auto rollback, dan ada kesalahan saat mengganti instance, penyegaran instans mencoba menyelesaikan semua penggantian selama satu jam sebelum gagal dan berputar kembali. Kesalahan ini biasanya disebabkan oleh hal-hal seperti kegagalan peluncuran EC2, pemeriksaan kesehatan yang salah dikonfigurasi, atau tidak mengabaikan atau mengizinkan penghentian instance yang dalam Standby status atau dilindungi dari skala masuk.

Menentukan CloudWatch alarm adalah opsional. Untuk menentukan alarm, Anda harus membuatnya terlebih dahulu. Anda dapat menentukan alarm metrik dan alarm komposit. Untuk informasi tentang membuat alarm, lihat [Panduan CloudWatch Pengguna Amazon](#). Menggunakan metrik Elastic

Load Balancing sebagai contoh, jika Anda menggunakan Application Load Balancer, Anda dapat menggunakan metrik dan. `HTTPCode_ELB_5XX_Count` `HTTPCode_ELB_4XX_Count`

Pertimbangan

- Jika Anda menentukan CloudWatch alarm tetapi tidak mengaktifkan auto rollback, dan status alarm masuk keALARM, penyegaran instance gagal tanpa memutar kembali.
- Anda dapat memilih maksimal 10 alarm saat memulai penyegaran instans.
- Saat memilih CloudWatch alarm, alarm harus dalam keadaan yang kompatibel. Jika status alarm `INSUFFICIENT_DATA` atauALARM, Anda menerima kesalahan saat mencoba memulai penyegaran instans.
- Saat membuat alarm untuk Amazon EC2 Auto Scaling untuk digunakan, alarm harus menyertakan cara menangani titik data yang hilang. Jika metrik memang didesain agar sering kehilangan titik data, status alarm adalah `INSUFFICIENT_DATA` selama periode tersebut. Ketika ini terjadi, Auto Scaling Amazon EC2 tidak dapat menggantikan instans hingga titik data baru ditemukan. Untuk memaksa alarm mempertahankan OK status sebelumnyaALARM, Anda dapat memilih untuk mengabaikan data yang hilang. Untuk informasi selengkapnya, lihat [Mengonfigurasi cara alarm menangani data yang hilang](#) di CloudWatch Panduan Pengguna Amazon.

Console

Untuk memulai penyegaran instance dengan auto rollback (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.
3. Pada tab Penyegaran instans, di Penyegaran instans aktif, pilih Mulai penyegaran instans.
4. Ikuti [Mulai penyegaran instance \(konsol\)](#) prosedur dan konfigurasi setelan penyegaran instans sesuai kebutuhan.
5. (Opsional) Di bawah Segarkan pengaturan, untuk CloudWatch alarm, pilih Aktifkan CloudWatch alarm, lalu pilih satu atau beberapa alarm untuk mengidentifikasi masalah apa pun dan gagal pengoperasian jika alarm masuk ke status. ALARM
6. Di bawah Pengaturan Rollback, pilih Aktifkan rollback otomatis untuk secara otomatis memutar kembali penyegaran instans yang gagal ke konfigurasi yang terakhir Anda simpan di grup Auto Scaling sebelum memulai penyegaran instans.
7. Tinjau pilihan Anda, lalu pilih Mulai penyegaran instance.

AWS CLI

Untuk memulai penyegaran instance dengan auto rollback ()AWS CLI

Gunakan [start-instance-refresh](#) perintah dan true tentukan AutoRollback opsi diPreferences.

Contoh berikut menunjukkan cara memulai penyegaran instance yang secara otomatis akan memutar kembali jika sesuatu gagal. Ganti nilai *italicized* parameter dengan nilai Anda sendiri.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Isi dari config.json.

```
{
  "AutoScalingGroupName": "my-asg",
  "DesiredConfiguration": {
    "LaunchTemplate": {
      "LaunchTemplateName": "my-launch-template",
      "Version": "1"
    }
  },
  "Preferences": {
    "AutoRollback": true
  }
}
```

Atau, untuk memutar kembali secara otomatis ketika penyegaran instance gagal atau ketika CloudWatch alarm tertentu dalam ALARM status, tentukan AlarmSpecification opsi di Preferences dan berikan nama alarm, seperti pada contoh berikut. Ganti nilai *italicized* parameter dengan nilai Anda sendiri.

```
{
  "AutoScalingGroupName": "my-asg",
  "DesiredConfiguration": {
    "LaunchTemplate": {
      "LaunchTemplateName": "my-launch-template",
      "Version": "1"
    }
  },
  "Preferences": {
```

```
"AutoRollback": true,  
  "AlarmSpecification": { "Alarms": [ "my-alarm" ] }  
}
```

Jika berhasil, perintah mengembalikan output yang serupa dengan berikut.

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

 Tip

Jika perintah ini menimbulkan kesalahan, pastikan Anda telah memperbarui AWS CLI secara lokal ke versi terbaru.

Gunakan penyegaran instance dengan melewati pencocokan

Lewati pencocokan memberi tahu Amazon EC2 Auto Scaling untuk mengabaikan instans yang sudah memiliki pembaruan terbaru Anda. Dengan cara ini, Anda tidak mengganti lebih banyak instance daripada yang Anda butuhkan. Ini sangat membantu ketika Anda ingin memastikan bahwa grup Auto Scaling Anda menggunakan versi tertentu dari template peluncuran Anda dan hanya menggantikan instance yang menggunakan versi berbeda.

Pertimbangan berikut berlaku untuk melewati pencocokan:

- Jika Anda memulai penyegaran instans dengan pencocokan lewati dan konfigurasi yang diinginkan, Auto Scaling Amazon EC2 akan memeriksa untuk melihat apakah ada instans yang cocok dengan konfigurasi yang Anda inginkan. Kemudian, itu hanya menggantikan instance yang tidak sesuai dengan konfigurasi yang Anda inginkan. Setelah penyegaran instans berhasil, Auto Scaling Amazon EC2 memperbarui grup untuk mencerminkan konfigurasi yang Anda inginkan.
- Jika Anda memulai penyegaran instans dengan pencocokan lewati, tetapi Anda tidak menentukan konfigurasi yang diinginkan, Auto Scaling Amazon EC2 akan memeriksa apakah ada instance yang cocok dengan konfigurasi yang terakhir Anda simpan di grup Auto Scaling. Kemudian, itu hanya menggantikan instance yang tidak cocok dengan konfigurasi terakhir Anda yang disimpan.
- Anda dapat menggunakan skip matching dengan template peluncuran baru, versi baru dari template peluncuran, atau serangkaian jenis instance. Jika Anda mengaktifkan pencocokan

lewati, tetapi tidak ada yang berubah, penyegaran instance akan segera berhasil tanpa mengganti instance apa pun. Jika Anda membuat perubahan lain dalam konfigurasi yang diinginkan (seperti mengubah strategi alokasi Spot), Auto Scaling Amazon EC2 menunggu penyegaran instans berhasil. Kemudian, ini memperbarui pengaturan grup Auto Scaling untuk mencerminkan konfigurasi baru yang diinginkan.

- Anda tidak dapat menggunakan pencocokan lewat dengan konfigurasi peluncuran baru.
- Saat memulai penyegaran instans dan menyediakan konfigurasi yang diinginkan, Auto Scaling Amazon EC2 memastikan bahwa semua instans menggunakan konfigurasi yang Anda inginkan. Oleh karena itu, ketika Anda menentukan salah satu `$Default` atau `$Latest` sebagai versi yang diinginkan untuk template peluncuran Anda dan kemudian membuat versi baru dari template peluncuran saat penyegaran instance sedang berlangsung, setiap instance yang sudah diganti akan diganti lagi.
- Lewati pencocokan tidak tahu apakah skrip data pengguna di template peluncuran akan menarik kode yang diperbarui dan menginstalnya pada instance baru. Akibatnya, lewat pencocokan mungkin melewatkan penggantian instance yang memiliki kode usang diinstal. Dalam hal ini, Anda harus menonaktifkan pencocokan lewat untuk memastikan bahwa semua instance menerima kode terbaru Anda, bahkan tanpa pembaruan versi templat peluncuran.

Bagian ini mencakup AWS CLI petunjuk untuk memulai penyegaran instans dengan pencocokan lewat diaktifkan. Untuk petunjuk tentang penggunaan konsol, lihat [Mulai penyegaran instance \(konsol\)](#).

Lewati pencocokan (prosedur dasar)

Ikuti langkah-langkah di bagian ini untuk menggunakan AWS CLI untuk melakukan hal berikut:

- Buat template peluncuran yang ingin Anda terapkan ke instance Anda.
- Mulai penyegaran instans untuk menerapkan template peluncuran Anda ke grup Auto Scaling Anda. Jika Anda tidak mengaktifkan pencocokan lewat, semua instance akan diganti. Ini benar bahkan jika template peluncuran yang digunakan untuk menyediakan instance sama dengan yang Anda tentukan untuk konfigurasi yang Anda inginkan.

Untuk menggunakan skip matching dengan template peluncuran baru

1. Gunakan [create-launch-template](#) perintah untuk membuat template peluncuran baru untuk grup Auto Scaling Anda. Sertakan `--launch-template-data` opsi dan masukan JSON yang menentukan detail instance yang dibuat untuk grup Auto Scaling Anda.

Misalnya, gunakan perintah berikut untuk membuat template peluncuran dasar dengan ID AMI `ami-0123456789abcdef0` dan tipe `t2.micro` instans.

```
aws ec2 create-launch-template --launch-template-name my-template-for-auto-scaling
--version-description version1 \
--launch-template-data
'{"ImageId": "ami-0123456789abcdef0", "InstanceType": "t2.micro"}'
```

Jika berhasil, perintah mengembalikan output yang serupa dengan berikut.

```
{
  "LaunchTemplate": {
    "LaunchTemplateId": "lt-068f72b729example",
    "LaunchTemplateName": "my-template-for-auto-scaling",
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
    "CreateTime": "2023-01-30T18:16:06.000Z",
    "DefaultVersionNumber": 1,
    "LatestVersionNumber": 1
  }
}
```

Untuk informasi selengkapnya, lihat [Contoh untuk membuat dan mengelola template peluncuran dengan AWS Command Line Interface \(AWS CLI\)](#).

- Gunakan `start-instance-refresh` perintah untuk memulai alur kerja penggantian instance dan menerapkan template peluncuran baru Anda dengan ID. `lt-068f72b729example` Karena template peluncuran baru, ia hanya memiliki satu versi. Ini berarti bahwa 1 versi template peluncuran adalah target penyegaran instance ini. Jika peristiwa penskalaan terjadi selama penyegaran instans, dan Auto Scaling Amazon EC2 menyediakan instans baru menggunakan versi templat peluncuran ini, instans 1 tersebut tidak akan diganti. Setelah berhasil menyelesaikan operasi, template peluncuran baru berhasil diterapkan ke grup Auto Scaling Anda.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Isi dari `config.json`.

```
{
  "AutoScalingGroupName": "my-asg",
```

```

"DesiredConfiguration": {
  "LaunchTemplate": {
    "LaunchTemplateId": "lt-068f72b729example",
    "Version": "$Default"
  }
},
"Preferences": {
  "SkipMatching": true
}
}

```

Jika berhasil, perintah mengembalikan output yang serupa dengan berikut.

```

{
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}

```

Lewati pencocokan (grup instance campuran)

Jika Anda memiliki grup Auto Scaling dengan [kebijakan instance campuran](#), ikuti langkah-langkah di bagian ini untuk menggunakan grup Auto Scaling dengan skip matching. AWS CLI Anda memiliki opsi berikut:

- Berikan template peluncuran baru untuk diterapkan ke semua jenis instance yang ditentukan dalam kebijakan.
- Berikan kumpulan tipe instans yang diperbarui dengan atau tanpa mengubah templat peluncuran dalam kebijakan. Misalnya, Anda mungkin ingin bermigrasi dari jenis instance yang tidak diinginkan. Anda akan menggunakan template peluncuran apa adanya, tanpa mengubah AMI, grup keamanan, atau spesifikasi lain dari instance yang diganti.

Ikuti langkah-langkah di salah satu bagian berikut, tergantung pada opsi mana yang sesuai dengan kebutuhan Anda.

Untuk menggunakan skip matching dengan template peluncuran baru

1. Gunakan [create-launch-template](#) perintah untuk membuat template peluncuran baru untuk grup Auto Scaling Anda. Sertakan `--launch-template-data` opsi dan masukan JSON yang menentukan detail instance yang dibuat untuk grup Auto Scaling Anda.

Misalnya, gunakan perintah berikut untuk membuat template peluncuran dengan ID AMI *ami-0123456789abcdef0*.

```
aws ec2 create-launch-template --launch-template-name my-new-template --version-  
description version1 \  
--launch-template-data '{"ImageId":"ami-0123456789abcdef0"}'
```

Jika berhasil, perintah mengembalikan output yang serupa dengan berikut.

```
{  
  "LaunchTemplate": {  
    "LaunchTemplateId": "lt-04d5cc9b88example",  
    "LaunchTemplateName": "my-new-template",  
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",  
    "CreateTime": "2023-01-31T15:56:02.000Z",  
    "DefaultVersionNumber": 1,  
    "LatestVersionNumber": 1  
  }  
}
```

Untuk informasi selengkapnya, lihat [Contoh untuk membuat dan mengelola template peluncuran dengan AWS Command Line Interface \(AWS CLI\)](#).

2. Untuk melihat kebijakan instance campuran yang ada untuk grup Auto Scaling Anda, jalankan [describe-auto-scaling-groups](#) perintah. Anda akan memerlukan informasi ini di langkah berikutnya, ketika Anda memulai penyegaran instance.

Perintah contoh berikut mengembalikan kebijakan instance campuran yang dikonfigurasi untuk grup Auto Scaling bernama *my-asg*

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Jika berhasil, perintah mengembalikan output yang serupa dengan berikut.

```
{  
  "AutoScalingGroups": [  
    {  
      "AutoScalingGroupName": "my-asg",  
      "AutoScalingGroupARN": "arn",  
      "MixedInstancesPolicy": {
```

```

    "LaunchTemplate":{
      "LaunchTemplateSpecification":{
        "LaunchTemplateId":"lt-073693ed27example",
        "LaunchTemplateName":"my-old-template",
        "Version":"$Default"
      },
      "Overrides":[
        {
          "InstanceType":"c5.large"
        },
        {
          "InstanceType":"c5a.large"
        },
        {
          "InstanceType":"m5.large"
        },
        {
          "InstanceType":"m5a.large"
        }
      ]
    },
    "InstancesDistribution":{
      "OnDemandAllocationStrategy":"prioritized",
      "OnDemandBaseCapacity":1,
      "OnDemandPercentageAboveBaseCapacity":50,
      "SpotAllocationStrategy":"price-capacity-optimized"
    }
  },
  "MinSize":1,
  "MaxSize":5,
  "DesiredCapacity":4,
  ...
}
]
}

```

- Gunakan [start-instance-refresh](#) perintah untuk memulai alur kerja penggantian instance dan menerapkan template peluncuran baru Anda dengan ID. *lt-04d5cc9b88example* Karena template peluncuran baru, ia hanya memiliki satu versi. Ini berarti bahwa 1 versi template peluncuran adalah target penyegaran instance ini. Jika peristiwa penskalaan terjadi selama penyegaran instans, dan Auto Scaling Amazon EC2 menyediakan instans baru menggunakan versi templat peluncuran ini, instans 1 tersebut tidak akan diganti. Setelah

berhasil menyelesaikan operasi, kebijakan instans campuran yang diperbarui berhasil diterapkan ke grup Auto Scaling Anda.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Isi dari config.json.

```
{
  "AutoScalingGroupName": "my-asg",
  "DesiredConfiguration": {
    "MixedInstancesPolicy": {
      "LaunchTemplate": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-04d5cc9b88example",
          "Version": "$Default"
        },
        "Overrides": [
          {
            "InstanceType": "c5.large"
          },
          {
            "InstanceType": "c5a.large"
          },
          {
            "InstanceType": "m5.large"
          },
          {
            "InstanceType": "m5a.large"
          }
        ]
      },
      "InstancesDistribution": {
        "OnDemandAllocationStrategy": "prioritized",
        "OnDemandBaseCapacity": 1,
        "OnDemandPercentageAboveBaseCapacity": 50,
        "SpotAllocationStrategy": "price-capacity-optimized"
      }
    }
  },
  "Preferences": {
    "SkipMatching": true
  }
}
```

```
}  
}
```

Jika berhasil, perintah mengembalikan output yang serupa dengan berikut.

```
{  
  "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"  
}
```

Dalam prosedur berikutnya ini, Anda menyediakan serangkaian jenis instans yang diperbarui tanpa mengubah template peluncuran.

Untuk menggunakan pencocokan lewati dengan kumpulan tipe instans yang diperbarui

1. Untuk melihat kebijakan instance campuran yang ada untuk grup Auto Scaling Anda, jalankan [describe-auto-scaling-groups](#) perintah. Anda akan memerlukan informasi ini di langkah berikutnya, ketika Anda memulai penyegaran instance.

Perintah contoh berikut mengembalikan kebijakan instance campuran yang dikonfigurasi untuk grup Auto Scaling bernama. *my-asg*

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Jika berhasil, perintah mengembalikan output yang serupa dengan berikut.

```
{  
  "AutoScalingGroups": [  
    {  
      "AutoScalingGroupName": "my-asg",  
      "AutoScalingGroupARN": "arn",  
      "MixedInstancesPolicy": {  
        "LaunchTemplate": {  
          "LaunchTemplateSpecification": {  
            "LaunchTemplateId": "lt-073693ed27example",  
            "LaunchTemplateName": "my-template-for-auto-scaling",  
            "Version": "$Default"  
          },  
          "Overrides": [  
            {  
              "InstanceType": "c5.large"  
            }  
          ]  
        }  
      }  
    }  
  ]  
}
```

```
    },
    {
      "InstanceType":"c5a.large"
    },
    {
      "InstanceType":"m5.large"
    },
    {
      "InstanceType":"m5a.large"
    }
  ]
},
"InstancesDistribution":{
  "OnDemandAllocationStrategy":"prioritized",
  "OnDemandBaseCapacity":1,
  "OnDemandPercentageAboveBaseCapacity":50,
  "SpotAllocationStrategy":"price-capacity-optimized"
}
},
"MinSize":1,
"MaxSize":5,
"DesiredCapacity":4,
...
}
]
}
```

2. Gunakan [start-instance-refresh](#) perintah untuk memulai alur kerja penggantian instance dan menerapkan pembaruan Anda. Jika Anda ingin mengganti instance yang menggunakan tipe instans tertentu, konfigurasi yang Anda inginkan harus menentukan kebijakan instance campuran dengan hanya tipe instance yang Anda inginkan. Anda dapat memilih apakah akan menambahkan jenis instance baru di tempatnya.

Contoh perintah berikut memulai penyegaran instance tanpa jenis instance yang tidak diinginkan *m5a.large*. Ketika jenis instans dalam grup Anda tidak cocok dengan salah satu dari tiga tipe instans yang tersisa, instance akan diganti. (Perhatikan bahwa penyegaran instans tidak memilih jenis instans untuk menyediakan instance baru; sebaliknya, [strategi alokasi](#) melakukannya.) Setelah berhasil menyelesaikan operasi, kebijakan instans campuran yang diperbarui berhasil diterapkan ke grup Auto Scaling Anda.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Isi dari config.json

```
{
  "AutoScalingGroupName": "my-asg",
  "DesiredConfiguration": {
    "MixedInstancesPolicy": {
      "LaunchTemplate": {
        "LaunchTemplateSpecification": {
          "LaunchTemplateId": "lt-073693ed27example",
          "Version": "$Default"
        },
        "Overrides": [
          {
            "InstanceType": "c5.large"
          },
          {
            "InstanceType": "c5a.large"
          },
          {
            "InstanceType": "m5.large"
          }
        ]
      },
      "InstancesDistribution": {
        "OnDemandAllocationStrategy": "prioritized",
        "OnDemandBaseCapacity": 1,
        "OnDemandPercentageAboveBaseCapacity": 50,
        "SpotAllocationStrategy": "price-capacity-optimized"
      }
    }
  },
  "Preferences": {
    "SkipMatching": true
  }
}
```

Tambahkan pos pemeriksaan ke penyegaran instance

Saat menggunakan penyegaran instans, Anda dapat memilih untuk mengganti instance secara bertahap, sehingga Anda dapat melakukan verifikasi pada instans saat Anda pergi. Untuk melakukan

penggantian bertahap, Anda menambahkan pos pemeriksaan, yang merupakan titik waktu di mana penyegaran instance berhenti. Menggunakan pos pemeriksaan memberi Anda kontrol yang lebih besar atas cara Anda memilih untuk memperbarui grup Auto Scaling Anda. Ini membantu Anda mengonfirmasi bahwa aplikasi Anda akan berfungsi dengan cara yang andal dan dapat diprediksi.

Daftar Isi

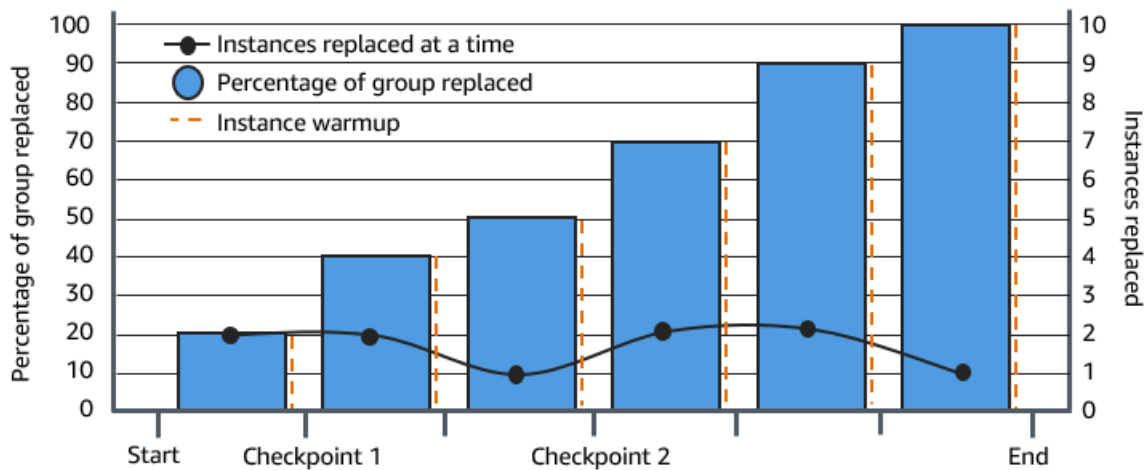
- [Cara kerjanya](#)
- [Pertimbangan](#)
- [Aktifkan pos pemeriksaan \(konsol\)](#)
- [Aktifkan pos pemeriksaan \(\)AWS CLI](#)

Cara kerjanya

Saat memulai penyegaran instans, Anda menentukan pos pemeriksaan sebagai persentase dari jumlah total instance dalam grup Auto Scaling. Pos pemeriksaan ini menunjukkan persentase minimum instance dalam grup Auto Scaling yang harus berupa instance baru sebelum pos pemeriksaan dianggap tercapai. Misalnya, jika pos pemeriksaan Anda [20, 50, 100], pos pemeriksaan pertama tercapai ketika 20 persen instance baru, yang kedua ketika 50 persen baru, dan pos pemeriksaan terakhir ketika semua instance baru.

Auto Scaling Amazon EC2 mempercepat penggantian instans untuk menghormati persentase pos pemeriksaan yang ditentukan sambil mempertahankan persentase kesehatan minimum grup. Untuk mencapai persentase pos pemeriksaan, Auto Scaling Amazon EC2 terkadang akan menggantikan lebih sedikit tetapi tidak pernah lebih dari yang diizinkan oleh persentase sehat minimum.

Pertimbangkan grup Auto Scaling berikut yang memiliki 10 instance. Persentase pos pemeriksaan adalah [20, 50, 100], persentase sehat minimum adalah 80 persen, dan persentase sehat maksimum adalah 100 persen. Untuk mempertahankan persentase sehat minimum, hanya dua contoh yang dapat diganti sekaligus. Diagram berikut merangkum proses untuk mengganti instance sebelum pos pemeriksaan tercapai.



Dalam contoh di atas, ada periode pemanasan instance untuk setiap instance baru yang dimulai. Anda mungkin juga memiliki kait siklus hidup yang menempatkan instance ke status tunggu dan kemudian melakukan tindakan khusus saat diluncurkan atau dihentikan.

Auto Scaling Amazon EC2 memancarkan peristiwa untuk setiap pos pemeriksaan kecuali untuk pos pemeriksaan lengkap 100 persen. Anda dapat menambahkan EventBridge aturan untuk mengirim acara ke target seperti Amazon SNS. Dengan cara ini, Anda akan diberi tahu ketika Anda dapat menjalankan verifikasi yang diperlukan. Untuk informasi selengkapnya, lihat [Buat EventBridge aturan untuk acara penyegaran misalnya](#).

Pertimbangan

Ingatlah pertimbangan berikut saat menggunakan pos pemeriksaan:

- Karena pos pemeriksaan didasarkan pada persentase, jumlah instance untuk mengganti perubahan dengan ukuran grup. Ketika aktivitas scale-out terjadi dan ukuran grup meningkat, operasi yang sedang berlangsung dapat mencapai pos pemeriksaan lagi. Jika itu terjadi, Amazon EC2 Auto Scaling mengirimkan pemberitahuan lain dan mengulangi waktu tunggu di antara pos pemeriksaan sebelum melanjutkan.
- Dimungkinkan untuk melewati pos pemeriksaan dalam keadaan tertentu. Misalnya, anggaplah grup Auto Scaling Anda memiliki dua instance dan persentase pos pemeriksaan Anda. [10, 40, 100] Setelah instance pertama diganti, Amazon EC2 Auto Scaling menghitung bahwa 50 persen dari grup diganti. Karena 50 persen lebih tinggi dari dua pos pemeriksaan pertama, ia melewati pos pemeriksaan pertama (10) dan mengirimkan pemberitahuan untuk pos pemeriksaan kedua (40).
- Membatalkan operasi menghentikan penggantian lebih lanjut dilakukan. Jika Anda membatalkan operasi atau gagal sebelum mencapai pos pemeriksaan terakhir, instance apa pun yang sudah diganti tidak akan dikembalikan ke konfigurasi sebelumnya.

- Untuk penyegaran sebagian, saat Anda menjalankan kembali operasi, Auto Scaling Amazon EC2 tidak dimulai ulang dari titik pos pemeriksaan terakhir, juga tidak berhenti ketika hanya instance sebelumnya yang diganti. Namun, ini menargetkan instance sebelumnya untuk penggantian terlebih dahulu, sebelum menargetkan instance baru.
- Persentase penyelesaian aktual mungkin lebih tinggi daripada persentase pos pemeriksaan itu ketika persentase pos pemeriksaan terlalu rendah relatif terhadap jumlah instance dalam grup. Misalnya, persentase pos pemeriksaan adalah 20 persen dan grup memiliki empat contoh. Jika Auto Scaling Amazon EC2 menggantikan salah satu dari empat instans, persentase sebenarnya yang diganti (25 persen) akan lebih tinggi daripada persentase pos pemeriksaan (20 persen).
- Setelah pos pemeriksaan tercapai, persentase keseluruhan yang ditampilkan selesai tidak diperbarui sampai setelah instance selesai pemanasan. Misalnya, persentase pos pemeriksaan Anda [20, 50] dengan penundaan pos pemeriksaan 15 menit dan persentase sehat minimum 80 persen. Grup Auto Scaling Anda memiliki 10 instans dan membuat penggantian berikut:
 - 0:00: Dua contoh sebelumnya diganti dengan yang baru.
 - 0:10: Dua contoh baru selesai pemanasan.
 - 0:25: Dua contoh sebelumnya diganti dengan yang baru. (Untuk mempertahankan persentase sehat minimum, hanya dua contoh yang diganti.)
 - 0:35: Dua contoh baru selesai pemanasan.
 - 0:35: Satu contoh sebelumnya diganti dengan yang baru.
 - 0:45: Satu contoh baru selesai pemanasan.

Pada 0:35, operasi berhenti meluncurkan instance baru. Persentase lengkap belum secara akurat mencerminkan jumlah penggantian yang diselesaikan (50 persen), karena instance baru belum selesai pemanasan. Setelah instance baru menyelesaikan periode pemanasannya pada 0:45, persentase lengkap menunjukkan 50 persen.

Aktifkan pos pemeriksaan (konsol)

Anda dapat mengaktifkan pos pemeriksaan sebelum memulai penyegaran instans untuk mengganti instance menggunakan pendekatan bertahap atau bertahap. Ini memberikan waktu tambahan untuk verifikasi.

Untuk memulai penyegaran instance yang menggunakan pos pemeriksaan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.

2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Pada tab Penyegaran instans, di Penyegaran instans aktif, pilih Mulai penyegaran instans.
4. Pada halaman Mulai penyegaran instance, masukkan nilai untuk Persentase sehat minimum dan pemanasan Instance.
5. Pilih kotak centang Aktifkan pos pemeriksaan.

Ini menampilkan kotak di mana Anda dapat menentukan ambang persentase untuk pos pemeriksaan pertama.

6. Untuk Lanjutkan hingga ____% dari grup di-refresh, masukkan angka (1-100). Ini menetapkan persentase untuk pos pemeriksaan pertama.
7. Untuk menambahkan pos pemeriksaan lain, pilih Tambahkan pos pemeriksaan dan kemudian tentukan persentase untuk pos pemeriksaan berikutnya.
8. Untuk menentukan berapa lama Auto Scaling Amazon EC2 menunggu setelah pos pemeriksaan tercapai, perbarui bidang di Tunggu di antara pos pemeriksaan. **1 hour** Satuan waktu dapat berupa jam, menit, atau detik.
9. Jika Anda sudah selesai dengan pilihan penyegaran instans, pilih Mulai penyegaran instans.

Aktifkan pos pemeriksaan ()AWS CLI

Untuk memulai penyegaran instance dengan pos pemeriksaan diaktifkan menggunakan AWS CLI, Anda memerlukan file konfigurasi yang mendefinisikan parameter berikut:

- `CheckpointPercentages`: Menentukan nilai ambang batas untuk persentase contoh yang akan diganti. Nilai ambang ini menyediakan pos pemeriksaan. Ketika persentase instance yang diganti dan dipanaskan mencapai salah satu ambang batas yang ditentukan, operasi menunggu periode waktu tertentu. Anda menentukan jumlah detik untuk menunggu `CheckpointDelay`. Ketika periode waktu yang ditentukan telah berlalu, penyegaran instance berlanjut hingga mencapai pos pemeriksaan berikutnya (jika ada).
- `CheckpointDelay`: Menentukan jumlah waktu, dalam hitungan detik, untuk menunggu setelah pos pemeriksaan tercapai sebelum melanjutkan. Pilih periode waktu yang menyediakan cukup waktu untuk melakukan verifikasi Anda.

Nilai terakhir yang ditampilkan dalam `CheckpointPercentages` larik menggambarkan persentase grup Auto Scaling yang perlu diganti dengan sukses. Transisi operasi `Successful` setelah persentase ini berhasil diganti dan setiap instance dianggap telah selesai menginisialisasi.

Untuk membuat beberapa pos pemeriksaan

Untuk membuat beberapa pos pemeriksaan, gunakan [start-instance-refresh](#) perintah contoh berikut. Contoh ini mengonfigurasi penyegaran instance yang awalnya menyegarkan satu persen dari grup Auto Scaling. Setelah menunggu 10 menit, kemudian menyegarkan 19 persen berikutnya dan menunggu 10 menit lagi. Akhirnya, itu menyegarkan sisa kelompok sebelum menyelesaikan operasi.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Isi dari `config.json`:

```
{
  "AutoScalingGroupName": "my-asg",
  "Preferences": {
    "InstanceWarmup": 60,
    "MinHealthyPercentage": 80,
    "CheckpointPercentages": [1,20,100],
    "CheckpointDelay": 600
  }
}
```

Untuk membuat pos pemeriksaan tunggal

Untuk membuat pos pemeriksaan tunggal, gunakan [start-instance-refresh](#) perintah contoh berikut. Contoh ini mengonfigurasi penyegaran instance yang awalnya menyegarkan 20 persen grup Auto Scaling. Setelah menunggu 10 menit, kemudian menyegarkan sisa kelompok sebelum menyelesaikan operasi.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Isi dari `config.json`:

```
{
  "AutoScalingGroupName": "my-asg",
  "Preferences": {
    "InstanceWarmup": 60,
```

```
"MinHealthyPercentage": 80,  
"CheckpointPercentages": [20,100],  
"CheckpointDelay": 600  
}  
}
```

Untuk menyegarkan sebagian grup Auto Scaling

Untuk mengganti hanya sebagian dari grup Auto Scaling Anda dan kemudian berhenti sepenuhnya, gunakan perintah contoh [start-instance-refresh](#) berikut. Contoh ini mengonfigurasi penyegaran instance yang awalnya menyegarkan satu persen dari grup Auto Scaling. Setelah menunggu 10 menit, kemudian menyegarkan 19 persen berikutnya sebelum menyelesaikan operasi.

```
aws autoscaling start-instance-refresh --cli-input-json file://config.json
```

Isi dari config.json:

```
{  
  "AutoScalingGroupName": "my-asg",  
  "Preferences": {  
    "InstanceWarmup": 60,  
    "MinHealthyPercentage": 80,  
    "CheckpointPercentages": [1,20],  
    "CheckpointDelay": 600  
  }  
}
```

Ganti instans Auto Scaling berdasarkan masa pakai instans maksimum

Masa pakai instans maksimum menentukan jumlah waktu maksimum (dalam detik) bahwa sebuah instance dapat digunakan sebelum dihentikan dan diganti. Kasus penggunaan umum mungkin merupakan persyaratan untuk mengganti instans Anda sesuai jadwal karena kebijakan keamanan internal atau kontrol kepatuhan eksternal.

Anda harus menentukan nilai minimal 86.400 detik (satu hari). Untuk menghapus nilai yang ditetapkan sebelumnya, tentukan nilai baru 0. Pengaturan ini berlaku untuk semua instans saat ini dan masa depan dalam grup Auto Scaling Anda.

Daftar Isi

- [Pertimbangan](#)
- [Mengatur masa pakai instans maksimum](#)
- [Batasan](#)

Pertimbangan

Berikut ini adalah pertimbangan saat menggunakan fitur ini:

- Setiap kali instance sebelumnya diganti dan instance baru diluncurkan, instance baru menggunakan template peluncuran atau konfigurasi peluncuran yang saat ini dikaitkan dengan grup Auto Scaling. Jika templat peluncuran atau konfigurasi peluncuran menentukan ID Amazon Machine Image (AMI) dari versi aplikasi yang berbeda, versi aplikasi ini akan diterapkan secara otomatis.
- Mengatur masa pakai instans maksimum terlalu rendah dapat menyebabkan instance diganti lebih cepat dari yang diinginkan. Auto Scaling Amazon EC2 biasanya akan menggantikan instans satu per satu, dengan jeda di antara penggantian. Namun, jika masa pakai instans maksimum yang ditentukan tidak menyediakan cukup waktu untuk mengganti setiap instans satu per satu, Auto Scaling Amazon EC2 harus mengganti lebih dari satu instance sekaligus. Beberapa contoh dapat diganti sekaligus, hingga 10 persen dari kapasitas grup Auto Scaling Anda saat ini. Untuk menghindari penggantian terlalu banyak instance sekaligus, setel masa pakai instans maksimum yang lebih lama atau gunakan perlindungan skala masuk instance untuk sementara mencegah instance individual dihentikan. Untuk informasi selengkapnya, lihat [Gunakan perlindungan skala dalam instance](#).
- Secara default, Auto Scaling Amazon EC2 membuat aktivitas penskalaan baru untuk menghentikan instance dan kemudian menghentikannya. Saat instance dihentikan, aktivitas penskalaan lain meluncurkan instance baru. Anda dapat mengubah perilaku ini untuk diluncurkan sebelum mengakhiri dengan menggunakan kebijakan pemeliharaan instans. Untuk informasi selengkapnya, lihat [Kebijakan pemeliharaan instans](#).

Mengatur masa pakai instans maksimum

Saat membuat grup Auto Scaling di konsol, Anda tidak dapat menyetel masa pakai instans maksimum. Namun, setelah grup dibuat, Anda dapat mengeditnya untuk mengatur masa pakai instance maksimum.

Untuk menyetel masa pakai instans maksimum untuk grup (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling, menampilkan informasi tentang grup yang Anda pilih.

3. Pada tab Detail, pilih Konfigurasi lanjutan, Edit.
4. Untuk masa pakai instans maksimum, masukkan jumlah detik maksimum yang dapat digunakan instance.
5. Pilih Perbarui.

Pada tab Aktivitas, di bawah Riwayat aktivitas, Anda dapat melihat penggantian instance dalam grup sepanjang histori.

Untuk mengatur masa pakai instance maksimum untuk grup (AWS CLI)

Anda juga dapat menggunakan AWS CLI untuk mengatur masa pakai instans maksimum untuk grup Auto Scaling baru atau yang sudah ada.

Untuk grup Auto Scaling baru, gunakan perintah. [create-auto-scaling-group](#)

```
aws autoscaling create-auto-scaling-group --cli-input-json file:///~/config.json
```

Berikut ini adalah contoh file `config.json` yang menampilkan masa aktif instans maksimum 2592000 detik (30 hari).

```
{
  "AutoScalingGroupName": "my-asg",
  "LaunchTemplate": {
    "LaunchTemplateName": "my-launch-template",
    "Version": "Default"
  },
  "MinSize": 1,
  "MaxSize": 5,
  "MaxInstanceLifetime": 2592000,
  "VPCZoneIdentifier": "subnet-5ea0c127, subnet-6194ea3b, subnet-c934b782",
  "Tags": []
}
```

```
}
```

Untuk grup Auto Scaling yang ada, gunakan perintah. [update-auto-scaling-group](#)

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-existing-asg --max-instance-lifetime 2592000
```

Untuk memverifikasi masa aktif instans maksimum untuk grup Auto Scaling

Gunakan perintah [describe-auto-scaling-groups](#).

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Batasan

- Masa pakai maksimum tidak dijamin tepat untuk setiap contoh: Instans tidak dijamin akan diganti hanya pada akhir durasi maksimumnya. Dalam beberapa situasi, Auto Scaling Amazon EC2 mungkin perlu mulai mengganti instans segera setelah Anda memperbarui parameter masa pakai instans maksimum. Alasan perilaku ini adalah untuk menghindari penggantian semua instance pada saat yang bersamaan.
- Perlindungan penskalaan instans dihormati: Auto Scaling Amazon EC2 memberikan perlindungan penskalaan instans untuk membantu Anda mengontrol instans mana yang dapat dihentikan. Jika perlindungan ini diaktifkan pada instans, Auto Scaling Amazon EC2 tidak akan menghentikan instans meskipun telah mencapai masa pakai instans maksimumnya.
- Instans dihentikan sebelum diluncurkan: Jika hanya ada satu instance dalam grup Auto Scaling, fitur masa pakai instans maksimum dapat mengakibatkan pemadaman karena Amazon EC2 Auto Scaling menghentikan instance dan kemudian meluncurkan instance baru secara default. Untuk mengubah perilaku ini untuk diluncurkan sebelum mengakhiri, lihat [Kebijakan pemeliharaan instans](#).

Skalakan ukuran grup Auto Scaling Anda

Penskalaan adalah kemampuan untuk meningkatkan atau menurunkan kapasitas komputasi aplikasi Anda. Penskalaan dimulai dengan suatu peristiwa, atau tindakan penskalaan, yang menginstruksikan grup Auto Scaling untuk meluncurkan atau menghentikan instance Amazon EC2.

Amazon EC2 Auto Scaling menyediakan sejumlah cara untuk menyesuaikan penskalaan yang paling sesuai dengan kebutuhan aplikasi Anda. Oleh karena itu, penting bagi Anda untuk memiliki pemahaman yang baik tentang aplikasi Anda. Perhatikan pertimbangan berikut:

- Peran apa yang harus dijalankan Amazon EC2 Auto Scaling dalam arsitektur aplikasi Anda? Sering kali kita berpikir tentang penskalaan otomatis terutama sebagai cara untuk meningkatkan dan mengurangi kapasitas, tetapi juga berguna untuk mempertahankan jumlah server yang stabil.
- Kendala biaya apakah yang penting bagi Anda? Karena Amazon EC2 Auto Scaling menggunakan instans EC2, Anda hanya membayar untuk sumber daya yang Anda gunakan. Memahami keterbatasan biaya akan membantu Anda memutuskan kapan Anda dapat menskalakan aplikasi, dan seberapa banyak.
- Metrik apa yang penting bagi aplikasi Anda? Amazon CloudWatch mendukung sejumlah metrik berbeda yang dapat Anda gunakan dengan grup Auto Scaling Anda.

Daftar Isi

- [Pilih metode penskalaan Anda](#)
- [Tetapkan batas penskalaan untuk grup Auto Scaling](#)
- [Mengatur pemanasan instance default untuk grup Auto Scaling](#)
- [Skala manual untuk Amazon EC2 Auto Scaling](#)
- [Skala terjadwal untuk Amazon EC2 Auto Scaling](#)
- [Penskalaan dinamis untuk Amazon EC2 Auto Scaling](#)
- [Penskalaan prediktif untuk Auto Scaling Amazon EC2](#)
- [Kontrol instans Auto Scaling mana yang akan dihentikan selama penskalaan](#)
- [Menangguhkan dan melanjutkan proses Auto Scaling Amazon EC2](#)

Pilih metode penskalaan Anda

Amazon EC2 Auto Scaling menyediakan beberapa cara untuk menskalakan grup Auto Scaling Anda.

Pertahankan jumlah instans yang tetap

Default untuk grup Auto Scaling adalah tidak memiliki kebijakan penskalaan terlampir atau tindakan terjadwal, yang menyebabkannya mempertahankan ukuran tetap. Setelah Anda membuat grup Auto Scaling, grup ini dimulai dengan meluncurkan instans yang cukup untuk memenuhi kapasitas yang diinginkan. Jika tidak ada kondisi penskalaan yang melekat pada grup, ia terus mempertahankan kapasitas yang diinginkan bahkan jika sebuah instance menjadi tidak sehat. Auto Scaling Amazon EC2 memantau kesehatan setiap instans di grup Auto Scaling Anda. Ketika menemukan bahwa sebuah instance telah menjadi tidak sehat, ia menggantikannya dengan instance baru. Anda dapat membaca deskripsi yang lebih mendalam tentang proses ini di [Health memeriksa instans dalam grup Auto Scaling](#).

Menskalakan secara manual

Penskalaan manual adalah cara paling dasar untuk menskalakan grup Auto Scaling Anda. Anda dapat memperbarui kapasitas yang diinginkan dari grup Auto Scaling atau menghentikan instance di grup Auto Scaling. Untuk informasi selengkapnya, lihat [Skala manual untuk Amazon EC2 Auto Scaling](#).

Menskalakan berdasarkan jadwal

Penskalaan berdasarkan jadwal berarti tindakan penskalaan dilakukan secara otomatis sebagai fungsi tanggal dan waktu. Ini berguna saat Anda mengetahui kapan tepatnya menambah atau mengurangi jumlah instance dalam grup Anda, hanya karena kebutuhan tersebut muncul pada jadwal yang dapat diprediksi. Untuk informasi selengkapnya, lihat [Skala terjadwal untuk Amazon EC2 Auto Scaling](#).

Skala dinamis berdasarkan permintaan

Cara yang lebih canggih untuk menskalakan sumber daya Anda, menggunakan penskalaan dinamis, memungkinkan Anda menentukan kebijakan penskalaan yang mengubah ukuran grup Auto Scaling secara dinamis untuk memenuhi perubahan permintaan. Misalnya, kita asumsikan Anda memiliki aplikasi web yang saat ini berjalan pada dua instance dan Anda ingin pemanfaatan CPU dari grup Auto Scaling untuk tetap berada di sekitar 50 persen ketika beban pada aplikasi berubah. Metode ini berguna untuk penskalaan karena perubahan lalu lintas terjadi, ketika Anda tidak tahu kapan lalu lintas akan berubah. Anda dapat mengonfigurasi kebijakan penskalaan untuk merespons Anda. Ada beberapa jenis kebijakan (atau kombinasi dari mereka) yang dapat Anda gunakan untuk skala dalam menanggapi perubahan lalu lintas. Untuk informasi selengkapnya, lihat [Penskalaan dinamis untuk Amazon EC2 Auto Scaling](#).

Skala secara proaktif

Anda juga dapat menggabungkan penskalaan prediktif dan penskalaan dinamis (pendekatan proaktif dan reaktif, masing-masing) untuk menskalakan kapasitas EC2 Anda lebih cepat. Gunakan penskalaan prediktif untuk meningkatkan jumlah instans EC2 di grup Auto Scaling Anda sebelum pola harian dan mingguan dalam arus lalu lintas. Untuk informasi selengkapnya, lihat [Penskalaan prediktif untuk Auto Scaling Amazon EC2](#).

Tetapkan batas penskalaan untuk grup Auto Scaling

Batas penskalaan mewakili ukuran grup minimum dan maksimum yang Anda inginkan untuk grup Auto Scaling Anda. Anda menetapkan batas secara terpisah untuk ukuran minimum dan maksimum.

Kapasitas yang diinginkan grup dapat diubah ukurannya menjadi angka yang berada dalam kisaran batas ukuran minimum dan maksimum Anda. Kapasitas yang diinginkan harus sama dengan atau lebih besar dari ukuran kelompok minimum, dan sama dengan atau kurang dari ukuran kelompok maksimum.

- **Kapasitas yang diinginkan:** Merupakan kapasitas awal grup Auto Scaling pada saat pembuatan. Grup Auto Scaling mencoba mempertahankan kapasitas yang diinginkan. Ini dimulai dengan meluncurkan jumlah instance yang ditentukan untuk kapasitas yang diinginkan, dan mempertahankan jumlah instance ini selama tidak ada kebijakan penskalaan atau tindakan terjadwal yang dilampirkan ke grup Auto Scaling.
- **Kapasitas minimum:** Merupakan ukuran grup minimum. Ketika kebijakan penskalaan ditetapkan, mereka tidak dapat mengurangi kapasitas yang diinginkan grup lebih rendah dari kapasitas minimum.
- **Kapasitas maksimum:** Merupakan ukuran grup maksimum. Ketika kebijakan penskalaan ditetapkan, mereka tidak dapat meningkatkan kapasitas yang diinginkan grup lebih tinggi dari kapasitas maksimum.

Batas ukuran minimum dan maksimum juga berlaku dalam skenario berikut:

- Saat Anda menskalakan grup Auto Scaling secara manual dengan memperbarui kapasitas yang diinginkan.
- Ketika tindakan terjadwal berjalan yang memperbarui kapasitas yang diinginkan. Jika tindakan terjadwal berjalan tanpa menentukan batas ukuran minimum dan maksimum baru untuk grup, maka batas ukuran minimum dan maksimum grup saat ini berlaku.

Grup Auto Scaling selalu berusaha mempertahankan kapasitas yang diinginkan. Dalam kasus di mana instance berakhir secara tak terduga (misalnya, karena gangguan Instans Spot, kegagalan pemeriksaan kesehatan, atau tindakan manusia), grup secara otomatis meluncurkan instance baru untuk mempertahankan kapasitas yang diinginkan.

Untuk mengelola pengaturan ini di konsol

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pada panel navigasi, di bawah Auto Scaling, pilih Grup Auto Scaling.
3. Pada halaman grup Auto Scaling, pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

4. Di panel bawah, di tab Detail, lihat atau ubah pengaturan saat ini untuk kapasitas yang diinginkan, minimum, dan maksimum grup. Untuk informasi selengkapnya, lihat [Ubah kapasitas yang diinginkan dari grup Auto Scaling yang ada](#).

Di atas panel Detail, Anda dapat menemukan informasi seperti jumlah instans saat ini di grup Auto Scaling, kapasitas minimum, dan maksimum yang diinginkan, dan kolom status. Jika grup Auto Scaling menggunakan bobot instans, Anda juga dapat menemukan jumlah unit kapasitas yang dikontribusikan ke kapasitas yang diinginkan.

Untuk menambah atau menghapus kolom dari daftar, pilih ikon pengaturan di bagian atas halaman. Kemudian, untuk atribut grup Auto Scaling, aktifkan atau nonaktifkan setiap kolom, lalu pilih Konfirmasi.

Untuk memverifikasi ukuran grup Auto Scaling Anda setelah melakukan perubahan

Kolom Instances menunjukkan jumlah instance yang sedang berjalan. Saat instance diluncurkan atau dihentikan, kolom Status menampilkan status Memperbarui kapasitas, seperti yang ditunjukkan pada gambar berikut.

<input checked="" type="checkbox"/>	Name	Launch template...	Instances	Status	Desired...	Min	Max
<input checked="" type="checkbox"/>	my-asg	my_template Version Def	0	Updating capacity	1	0	1

Tunggu beberapa menit, lalu refresh tampilan untuk melihat status terbaru. Setelah aktivitas penskalaan selesai, kolom Instance menampilkan nilai yang diperbarui.

Anda dapat melihat jumlah instance dan status instans yang sedang berjalan dari tab Manajemen instans, di bawah Instans.

Mengatur pemanasan instance default untuk grup Auto Scaling

CloudWatch mengumpulkan dan menggabungkan data penggunaan, seperti CPU dan jaringan I/O, di seluruh instans Auto Scaling Anda. Anda menggunakan metrik ini untuk membuat kebijakan penskalaan yang menyesuaikan jumlah instance di grup Auto Scaling Anda saat nilai metrik yang dipilih meningkat dan menurun.

Anda dapat menentukan berapa lama setelah instance mencapai InService status yang ditunggu sebelum menyumbangkan data penggunaan ke metrik agregat. Waktu yang ditentukan ini disebut pemanasan instance default. Hal ini membuat penskalaan dinamis tidak terpengaruh oleh metrik untuk instance individual yang belum menangani lalu lintas aplikasi dan yang mungkin mengalami penggunaan sumber daya komputasi yang tinggi untuk sementara waktu.

Untuk mengoptimalkan kinerja kebijakan pelacakan target dan penskalaan langkah, kami sangat menyarankan agar Anda mengaktifkan dan mengonfigurasi pemanasan instans default. Itu tidak diaktifkan atau dikonfigurasi secara default.

Saat mengaktifkan pemanasan instans default, ingatlah bahwa jika grup Auto Scaling disetel untuk menggunakan kebijakan pemeliharaan instans, atau Anda menggunakan penyegaran instans untuk mengganti instance, Anda dapat mencegah instance dihitung ke persentase sehat minimum sebelum selesai diinisialisasi.

Daftar Isi

- [Pertimbangan kinerja penskalaan](#)
- [Pilih waktu pemanasan instance default](#)
- [Aktifkan pemanasan instance default untuk grup](#)
- [Verifikasi pemanasan instance default untuk grup](#)
- [Temukan kebijakan penskalaan dengan waktu pemanasan instance yang ditetapkan sebelumnya](#)
- [Hapus pemanasan instance yang disetel sebelumnya untuk kebijakan penskalaan](#)

Pertimbangan kinerja penskalaan

Sangat berguna bagi sebagian besar aplikasi untuk memiliki satu waktu pemanasan instance default yang berlaku untuk semua fitur, daripada waktu pemanasan yang berbeda untuk fitur yang berbeda. Misalnya, jika Anda tidak menyetel pemanasan instans default, fitur penyegaran instance menggunakan masa tenggang pemeriksaan kesehatan sebagai waktu pemanasan default. Jika Anda memiliki kebijakan pelacakan target dan penskalaan langkah, mereka menggunakan nilai yang ditetapkan untuk cooldown default sebagai waktu pemanasan default. Jika Anda memiliki kebijakan penskalaan prediktif, kebijakan tersebut tidak memiliki waktu pemanasan default.

Saat instans sedang memanaskan, kebijakan penskalaan dinamis Anda hanya dapat ditingkatkan jika nilai metrik dari instans yang tidak memanaskan lebih besar daripada ambang batas tinggi alarm kebijakan (atau pemanfaatan target kebijakan penskalaan pelacakan target). Jika permintaan menurun, penskalaan dinamis menjadi lebih konservatif untuk melindungi ketersediaan aplikasi Anda. Ini memblokir aktivitas penskalaan untuk penskalaan dinamis hingga instance baru selesai melakukan pemanasan.

Saat melakukan scaling, Amazon EC2 Auto Scaling mempertimbangkan instans yang melakukan pemanasan sebagai bagian dari kapasitas grup saat memutuskan berapa banyak instance yang akan ditambahkan ke grup. Oleh karena itu, beberapa pelanggaran alarm yang memerlukan jumlah kapasitas yang sama untuk ditambahkan menghasilkan aktivitas penskalaan tunggal. Tujuannya adalah untuk terus meningkatkan skala, tanpa melakukannya secara berlebihan.

Jika pemanasan instans default tidak diaktifkan, jumlah waktu instance menunggu sebelum mengirim metrik CloudWatch dan menghitungnya ke kapasitas saat ini akan bervariasi dari satu instance ke instance lainnya. Jadi, ada potensi kebijakan penskalaan Anda untuk berkinerja tidak terduga dibandingkan dengan beban kerja aktual yang terjadi.

Misalnya, pertimbangkan aplikasi dengan pola on-and-off beban kerja berulang. Kebijakan penskalaan prediktif digunakan untuk membuat keputusan berulang tentang apakah akan meningkatkan jumlah instance. Karena tidak ada waktu pemanasan default untuk kebijakan penskalaan prediktif, instans mulai berkontribusi pada metrik agregat segera. Jika instance ini memiliki penggunaan sumber daya yang lebih tinggi saat startup, maka menambahkan instance dapat menyebabkan metrik agregat melonjak. Bergantung pada berapa lama waktu yang dibutuhkan untuk menstabilkan penggunaan, ini dapat memengaruhi kebijakan penskalaan dinamis apa pun yang menggunakan metrik ini. Jika ambang batas tinggi alarm kebijakan penskalaan dinamis dilanggar, maka ukuran grup bertambah lagi. Sementara instance baru sedang memanaskan, aktivitas scale-in akan diblokir.

Pilih waktu pemanasan instance default

Kunci untuk menyetel pemanasan instans default adalah menentukan berapa lama instans Anda harus menyelesaikan inisialisasi dan agar konsumsi sumber daya stabil setelah mencapai status. InService Saat memilih waktu pemanasan instans, cobalah untuk menjaga keseimbangan optimal antara mengumpulkan data penggunaan untuk lalu lintas yang sah, dan meminimalkan pengumpulan data yang terkait dengan lonjakan penggunaan sementara saat startup.

Misalkan Anda memiliki grup Auto Scaling yang terpasang pada penyeimbang beban Elastic Load Balancing. Ketika instance baru selesai diluncurkan, mereka terdaftar ke penyeimbang beban sebelum memasuki status. InService Setelah instance memasuki InService status, konsumsi sumber daya masih dapat mengalami lonjakan sementara dan perlu waktu untuk menstabilkan. Misalnya, konsumsi sumber daya untuk server aplikasi yang harus mengunduh dan menyimpan aset besar membutuhkan waktu lebih lama untuk menstabilkan daripada server web ringan tanpa aset besar untuk diunduh. Instance warmup menyediakan penundaan waktu yang diperlukan agar konsumsi sumber daya stabil.

Important

Jika Anda tidak yakin berapa banyak waktu yang Anda butuhkan untuk waktu pemanasan, Anda bisa mulai dengan 300 detik. Kemudian secara bertahap kurangi atau tingkatkan hingga Anda mendapatkan kinerja penskalaan terbaik untuk aplikasi Anda. Anda mungkin perlu melakukan ini beberapa kali untuk melakukannya dengan benar. Atau, jika Anda memiliki kebijakan penskalaan yang memiliki waktu pemanasan (`EstimatedInstanceWarmup`) sendiri, Anda dapat menggunakan nilai ini untuk memulai. Untuk informasi selengkapnya, lihat [Temukan kebijakan penskalaan dengan waktu pemanasan instance yang ditetapkan sebelumnya](#).

Pertimbangkan untuk menggunakan kait siklus hidup untuk kasus penggunaan di mana Anda memiliki tugas konfigurasi atau skrip untuk dijalankan saat startup. Pengait siklus hidup dapat menunda instance baru agar tidak dimasukkan ke dalam layanan hingga selesai diinisialisasi. Mereka sangat berguna jika Anda memiliki skrip bootstrap yang membutuhkan waktu beberapa saat untuk diselesaikan. Jika Anda menambahkan kait siklus hidup, Anda dapat mengurangi nilai pemanasan instance default. Untuk informasi selengkapnya tentang penggunaan kait siklus hidup, lihat [Hook siklus aktif Amazon EC2 Auto Scaling](#)

Aktifkan pemanasan instance default untuk grup

Anda dapat mengaktifkan pemanasan instance default saat membuat grup Auto Scaling. Anda juga dapat mengaktifkannya untuk grup yang ada.

Dengan mengaktifkan fitur pemanasan instance default, Anda tidak lagi harus menentukan nilai untuk parameter pemanasan untuk fitur berikut:

- [Penyegaran contoh](#)
- [Penskalaan pelacakan target](#)
- [Penskalaan langkah](#)

Console

Untuk mengaktifkan pemanasan instance default untuk grup baru (konsol)

Saat Anda membuat grup Auto Scaling, pada halaman Konfigurasi opsi lanjutan, di bawah Pengaturan tambahan, pilih opsi Aktifkan pemanasan instans default. Pilih waktu pemanasan yang Anda butuhkan untuk aplikasi Anda.

AWS CLI

Untuk mengaktifkan pemanasan instance default untuk grup baru (AWS CLI)

Untuk mengaktifkan pemanasan instance default untuk grup Auto Scaling, tambahkan `--default-instance-warmup` opsi dan tentukan nilai, dalam hitungan detik, dari 0 hingga 3600. Setelah diaktifkan, nilai `-1` akan mematikan pengaturan ini.

[create-auto-scaling-group](#) *Perintah berikut membuat grup Auto Scaling dengan nama `my-asg` dan memungkinkan pemanasan instance default dengan nilai 120 detik.*

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg --  
default-instance-warmup 120 ...
```

Tip

Jika perintah ini menimbulkan kesalahan, pastikan Anda telah memperbarui AWS CLI secara lokal ke versi terbaru.

Console

Untuk mengaktifkan pemanasan instance default untuk grup yang ada (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pada bilah navigasi di bagian atas layar, pilih tempat Wilayah AWS Anda membuat grup Auto Scaling.
3. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

4. Pada tab Detail, pilih Konfigurasi lanjutan, Edit.
5. Untuk pemanasan contoh Default, pilih waktu pemanasan yang Anda butuhkan untuk aplikasi Anda.
6. Pilih Perbarui.

AWS CLI

Untuk mengaktifkan pemanasan instance default untuk grup yang ada (AWS CLI)

Contoh berikut menggunakan `update-auto-scaling-group` perintah untuk mengaktifkan pemanasan instance default dengan nilai 120 detik untuk grup Auto Scaling yang ada bernama `my-asg`.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --  
default-instance-warmup 120
```

Tip

Jika perintah ini menimbulkan kesalahan, pastikan Anda telah memperbarui AWS CLI secara lokal ke versi terbaru.

Verifikasi pemanasan instance default untuk grup

Untuk memverifikasi pemanasan instance default untuk grup Auto Scaling (AWS CLI)

Gunakan perintah berikut [describe-auto-scaling-groups](#). Ganti *my-asg* dengan nama grup Auto Scaling Anda.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Berikut ini adalah contoh respons.

```
{
  "AutoScalingGroups": [
    {
      "AutoScalingGroupName": "my-asg",
      "AutoScalingGroupARN": "arn",
      ...
      "DefaultInstanceWarmup": 120
    }
  ]
}
```

Temukan kebijakan penskalaan dengan waktu pemanasan instance yang ditetapkan sebelumnya

Untuk mengidentifikasi apakah Anda memiliki kebijakan yang memiliki waktu pemanasan sendiri `EstimatedInstanceWarmup`, jalankan [perintah deskripsikan](#) kebijakan berikut menggunakan AWS CLI. Ganti *my-asg* dengan nama grup Auto Scaling Anda.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg
--query 'ScalingPolicies[?EstimatedInstanceWarmup!=`null`]'
```

Berikut ini adalah output contoh.

```
[
  {
    "AutoScalingGroupName": "my-asg",
    "PolicyName": "cpu50-target-tracking-scaling-policy",
    "PolicyARN": "arn",
    "PolicyType": "TargetTrackingScaling",
    "StepAdjustments": [],
    "EstimatedInstanceWarmup": 120,
    "Alarms": [{
```

```

    "AlarmARN": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:TargetTracking-my-
asg-AlarmHigh-fc0e4183-23ac-497e-9992-691c9980c38e",
    "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
  },
  {
    "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2",
    "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
  }],
  "TargetTrackingConfiguration":{
    "PredefinedMetricSpecification":{
      "PredefinedMetricType":"ASGAverageCPUUtilization"
    },
    "TargetValue":50.0,
    "DisableScaleIn":false
  },
  "Enabled":true
},
... additional policies ...
]

```

Hapus pemanasan instance yang disetel sebelumnya untuk kebijakan penskalaan

Setelah mengaktifkan pemanasan instance default, perbarui kebijakan penskalaan apa pun yang masih memiliki waktu pemanasan sendiri untuk menghapus nilai yang ditetapkan sebelumnya. Jika tidak, itu akan mengganti pemanasan instance default.

Anda dapat memperbarui kebijakan penskalaan menggunakan konsol AWS CLI, atau AWS SDK. Bagian ini mencakup langkah-langkah untuk konsol. Jika Anda menggunakan AWS CLI atau AWS SDK, pastikan Anda mempertahankan konfigurasi kebijakan yang ada, tetapi hapus `EstimatedInstanceWarmup` properti. Saat memperbarui kebijakan penskalaan yang ada, kebijakan tersebut akan diganti dengan kebijakan yang Anda tentukan saat Anda menelepon secara terprogram. [PutScalingPolicy](#) Nilai asli tidak disimpan.

Untuk menghapus pemanasan instance yang disetel sebelumnya untuk kebijakan penskalaan (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Penskalaan otomatis, di Kebijakan penskalaan dinamis, pilih kebijakan yang Anda minati, lalu pilih Tindakan, Edit.
4. Untuk pemanasan Instance, kosongkan nilai pemanasan instance untuk menggunakan nilai pemanasan instance default sebagai gantinya.
5. Pilih Perbarui.

Skala manual untuk Amazon EC2 Auto Scaling

Anda dapat secara manual menyesuaikan jumlah instans EC2 di grup Auto Scaling kapan saja. Proses mengubah jumlah instance secara manual ini disebut sebagai penskalaan manual. Penskalaan manual adalah alternatif untuk penskalaan otomatis, terutama jika Anda ingin melakukan perubahan kapasitas satu kali.

Setelah Anda menskalakan grup secara manual, Auto Scaling Amazon EC2 melanjutkan aktivitas penskalaan otomatis normal berdasarkan kebijakan penskalaan dan tindakan terjadwal yang Anda tentukan. Untuk grup dengan pemanasan instans default diaktifkan, setiap instance baru akan melalui periode pemanasan sebelum mereka mulai berkontribusi pada metrik yang digunakan untuk penskalaan otomatis. Periode pemanasan ini membantu menstabilkan kelompok pada kapasitas baru. Untuk informasi selengkapnya, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#).

Terkadang, Anda mungkin ingin menonaktifkan sementara kebijakan penskalaan dan tindakan terjadwal sebelum menskalakan grup secara manual. Melakukannya mencegah timbulnya konflik antara tindakan penskalaan manual dan aktivitas penskalaan otomatis. Untuk informasi selengkapnya, lihat [Matikan aktivitas penskalaan](#).

Daftar Isi

- [Ubah kapasitas yang diinginkan dari grup Auto Scaling yang ada](#)
- [Mengakhiri instance di grup Auto Scaling \(AWS CLI\)](#)

Ubah kapasitas yang diinginkan dari grup Auto Scaling yang ada

Saat Anda mengubah kapasitas grup Auto Scaling yang diinginkan, Auto Scaling Amazon EC2 mengelola proses peluncuran dan penghentian instans untuk mencapai ukuran baru yang diinginkan.

Console

Untuk mengubah ukuran grup Auto Scaling Anda

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split ditampilkan di bagian bawah halaman.

3. Pada tab Detail, pilih Detail grup, Edit.
4. Untuk kapasitas yang diinginkan, tambah atau kurangi kapasitas yang diinginkan. Misalnya, untuk menambah ukuran grup dengan satu, jika nilai saat ini 1, masukkan 2.

Jika nilai baru Anda untuk kapasitas yang diinginkan lebih besar dari kapasitas yang diinginkan Min dan kapasitas Max yang diinginkan, kapasitas Max yang diinginkan secara otomatis ditingkatkan ke nilai kapasitas baru yang diinginkan.

5. Setelah selesai, pilih Perbarui.

Verifikasi bahwa ukuran grup yang Anda tentukan menghasilkan jumlah instans yang sama yang diluncurkan. Misalnya, jika Anda menambah ukuran grup satu per satu, verifikasi bahwa grup Auto Scaling Anda telah meluncurkan satu instance tambahan.

Untuk memverifikasi bahwa ukuran grup Auto Scaling Anda telah berubah

1. Pada tab Aktivitas, dalam riwayat Aktivitas, Anda dapat melihat kemajuan aktivitas yang terkait dengan grup Auto Scaling. Kolom Status menunjukkan status instans Anda saat ini. Saat instance Anda diluncurkan, kolom status akan ditampilkan `Not yet in service`. Status berubah menjadi `Successful` setelah instance diluncurkan. Anda juga dapat menggunakan ikon penyegaran untuk melihat status instans Anda saat ini. Untuk informasi selengkapnya, lihat [Memverifikasi aktivitas penskalaan untuk grup Auto Scaling](#).
2. Pada tab Manajemen instans, di Instans, Anda dapat melihat status instance. Butuh waktu singkat untuk meluncurkan sebuah instans.

- Kolom Siklus Aktif menunjukkan status instans Anda. Pada awalnya, instans Anda ada pada status Pending. Setelah instans siap untuk menerima lalu lintas, statusnya adalah InService.
- Kolom status Kesehatan menunjukkan hasil pemeriksaan kesehatan Auto Scaling Amazon EC2 pada instans Anda.

AWS CLI

Contoh berikut mengasumsikan bahwa Anda telah membuat grup Auto Scaling dengan ukuran minimum 1 dan ukuran maksimum 5. Oleh karena itu, grup tersebut saat ini memiliki satu instance yang sedang berjalan.

Untuk mengubah ukuran grup Auto Scaling Anda

Gunakan [set-desired-capacity](#) perintah untuk mengubah ukuran grup Auto Scaling Anda, seperti yang ditunjukkan pada contoh berikut.

```
aws autoscaling set-desired-capacity --auto-scaling-group-name my-asg \  
--desired-capacity 2
```

Jika Anda memilih untuk menghargai periode pendinginan default untuk grup Auto Scaling Anda, Anda harus menentukan opsi `--honor-cooldown` seperti yang ditunjukkan dalam contoh berikut. Untuk informasi selengkapnya, lihat [Menskalakan jeda pakai untuk Amazon EC2 Auto Scaling](#).

```
aws autoscaling set-desired-capacity --auto-scaling-group-name my-asg \  
--desired-capacity 2 --honor-cooldown
```

Untuk memverifikasi ukuran grup Auto Scaling Anda

Gunakan [describe-auto-scaling-groups](#) perintah untuk mengonfirmasi bahwa ukuran grup Auto Scaling Anda telah berubah, seperti pada contoh berikut.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Berikut ini adalah contoh output, yang memberikan rincian tentang grup dan instance yang diluncurkan.

```
{
```

```
"AutoScalingGroups": [
  {
    "AutoScalingGroupName": "my-asg",
    "AutoScalingGroupARN": "arn",
    "LaunchTemplate": {
      "LaunchTemplateName": "my-launch-template",
      "Version": "1",
      "LaunchTemplateId": "lt-050555ad16a3f9c7f"
    },
    "MinSize": 1,
    "MaxSize": 5,
    "DesiredCapacity": 2,
    "DefaultCooldown": 300,
    "AvailabilityZones": [
      "us-west-2a"
    ],
    "LoadBalancerNames": [],
    "TargetGroupARNs": [],
    "HealthCheckType": "EC2",
    "HealthCheckGracePeriod": 300,
    "Instances": [
      {
        "ProtectedFromScaleIn": false,
        "AvailabilityZone": "us-west-2a",
        "LaunchTemplate": {
          "LaunchTemplateName": "my-launch-template",
          "Version": "1",
          "LaunchTemplateId": "lt-050555ad16a3f9c7f"
        },
        "InstanceId": "i-05b4f7d5be44822a6",
        "InstanceType": "t3.micro",
        "HealthStatus": "Healthy",
        "LifecycleState": "Pending"
      },
      {
        "ProtectedFromScaleIn": false,
        "AvailabilityZone": "us-west-2a",
        "LaunchTemplate": {
          "LaunchTemplateName": "my-launch-template",
          "Version": "1",
          "LaunchTemplateId": "lt-050555ad16a3f9c7f"
        },
        "InstanceId": "i-0c20ac468fa3049e8",
        "InstanceType": "t3.micro",
```

```

        "HealthStatus": "Healthy",
        "LifecycleState": "InService"
    }
],
"CreatedTime": "2019-03-18T23:30:42.611Z",
"SuspendedProcesses": [],
"VPCZoneIdentifier": "subnet-c87f2be0",
"EnabledMetrics": [],
"Tags": [],
"TerminationPolicies": [
    "Default"
],
"NewInstancesProtectedFromScaleIn": false,
"ServiceLinkedRoleARN": "arn",
"TrafficSources": []
}
]
}

```

Perhatikan bahwa `DesiredCapacity` menampilkan nilai baru. Grup Auto Scaling Anda telah meluncurkan sebuah instance tambahan.

Mengakhiri instance di grup Auto Scaling ()AWS CLI

Ada kalanya Anda mungkin ingin menskalakan secara manual di grup Auto Scaling tetapi ingin menghentikan instance tertentu. Anda dapat menskalakan secara manual di grup Auto Scaling dengan menggunakan perintah [terminate-instance-in-auto-scaling-group](#) dan menentukan ID instance yang ingin Anda akhiri dan `--should-decrement-desired-capacity` opsi seperti yang ditunjukkan pada contoh berikut.

```
aws autoscaling terminate-instance-in-auto-scaling-group \
  --instance-id i-026e4c9f62c3e448c --should-decrement-desired-capacity
```

Berikut ini adalah contoh output, yang memberikan rincian tentang aktivitas penskalaan.

```
{
  "Activities": [
    {
      "ActivityId": "b8d62b03-10d8-9df4-7377-e464ab6bd0cb",
      "AutoScalingGroupName": "my-asg",

```

```
"Description": "Terminating EC2 instance: i-026e4c9f62c3e448c",
  "Cause": "At 2023-09-23T06:39:59Z instance i-026e4c9f62c3e448c was taken
out of service in response to a user request, shrinking the capacity from 1 to 0.",
  "StartTime": "2023-09-23T06:39:59.015000+00:00",
  "StatusCode": "InProgress",
  "Progress": 0,
  "Details": "{\"Subnet ID\":\"subnet-6194ea3b\",\"Availability Zone\":\"us-
west-2c\"}"
}
```

Opsi ini tidak tersedia di konsol. Namun, Anda dapat menggunakan halaman Instans konsol Amazon EC2 untuk menghentikan instance di grup Auto Scaling. Ketika Anda melakukannya, Amazon EC2 Auto Scaling mendeteksi bahwa instans tidak lagi berjalan dan menggantinya secara otomatis sebagai bagian dari proses pemeriksaan kesehatan. Dibutuhkan satu atau dua menit setelah Anda menghentikan instance sebelum instance baru diluncurkan. Untuk informasi tentang cara menghentikan instance, lihat [Mengakhiri instance](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Jika Anda menghentikan instans di grup dan menyebabkan distribusi tidak merata di seluruh Availability Zone, Auto Scaling Amazon EC2 menyeimbangkan kembali grup untuk membuat kembali distribusi yang merata kecuali Anda menanggukannya. AZRebalance Untuk informasi selengkapnya, lihat [Menanggukannya dan melanjutkan proses Auto Scaling Amazon EC2](#).

Skala terjadwal untuk Amazon EC2 Auto Scaling

Dengan penskalaan terjadwal, Anda dapat mengatur penskalaan otomatis untuk aplikasi berdasarkan perubahan beban yang dapat diprediksi. Anda membuat tindakan terjadwal yang meningkatkan atau mengurangi kapasitas yang diinginkan grup Anda pada waktu tertentu.

Misalnya, Anda mengalami pola lalu lintas mingguan reguler di mana beban meningkat pertengahan minggu dan menurun menjelang akhir minggu. Anda dapat mengonfigurasi jadwal penskalaan di Amazon EC2 Auto Scaling yang sejajar dengan pola ini:

- Pada Rabu pagi, satu tindakan terjadwal meningkatkan kapasitas dengan meningkatkan kapasitas yang diinginkan sebelumnya dari grup Auto Scaling.
- Pada Jumat malam, tindakan terjadwal lainnya mengurangi kapasitas dengan mengurangi kapasitas yang diinginkan sebelumnya dari grup Auto Scaling.

Tindakan penskalaan terjadwal ini memungkinkan Anda mengoptimalkan biaya dan kinerja. Aplikasi Anda memiliki kapasitas yang cukup untuk menangani puncak lalu lintas tengah minggu, tetapi tidak menyediakan kapasitas yang tidak dibutuhkan secara berlebihan di lain waktu.

Anda dapat menggunakan kebijakan penskalaan dan penskalaan terjadwal bersama-sama untuk mendapatkan manfaat dari kedua pendekatan penskalaan. Setelah tindakan penskalaan terjadwal berjalan, kebijakan penskalaan dapat terus membuat keputusan tentang apakah akan meningkatkan kapasitas skala lebih lanjut. Ini membantu memastikan bahwa Anda memiliki kapasitas yang cukup untuk menangani beban untuk aplikasi Anda. Meskipun aplikasi Anda menskalakan sesuai dengan permintaan, kapasitas saat ini harus berada di antara kapasitas minimum dan maksimum yang ditetapkan oleh tindakan terjadwal Anda.

Daftar Isi

- [Cara kerja penskalaan terjadwal](#)
- [Jadwal berulang](#)
- [Zona waktu](#)
- [Pertimbangan](#)
- [Buat tindakan terjadwal](#)
- [Lihat detail tindakan terjadwal](#)
- [Verifikasi aktivitas penskalaan](#)
- [Hapus tindakan terjadwal](#)
- [Batasan](#)

Cara kerja penskalaan terjadwal

Untuk menggunakan penskalaan terjadwal, buat tindakan terjadwal, yang memberi tahu Auto Scaling Amazon EC2 untuk melakukan aktivitas penskalaan pada waktu tertentu. Saat membuat tindakan terjadwal, Anda menentukan grup Auto Scaling, kapan aktivitas penskalaan harus terjadi, kapasitas baru yang diinginkan, dan secara opsional kapasitas minimum baru dan kapasitas maksimum baru. Anda dapat membuat tindakan terjadwal yang menskalakan satu kali saja atau menskalakan berdasarkan jadwal berulang.

Pada waktu yang ditentukan, Amazon EC2 Auto Scaling menskalakan berdasarkan nilai kapasitas baru, dengan membandingkan kapasitas saat ini dengan kapasitas yang diinginkan yang ditentukan.

- Jika kapasitas saat ini kurang dari kapasitas yang diinginkan, Auto Scaling Amazon EC2 akan menurunkan skala, atau menambahkan instans, ke kapasitas yang diinginkan yang ditentukan.
- Jika kapasitas arus lebih besar dari kapasitas yang diinginkan, Amazon EC2 Auto Scaling menskalakan, atau menghapus instans, ke kapasitas yang diinginkan.

Tindakan terjadwal menetapkan kapasitas yang diinginkan, minimum, dan maksimum grup pada tanggal dan waktu yang ditentukan. Anda dapat membuat tindakan terjadwal hanya untuk satu dari kapasitas ini pada satu waktu, misalnya, kapasitas yang diinginkan. Namun, ada beberapa kasus di mana Anda harus memasukkan kapasitas minimum dan maksimum untuk memastikan bahwa kapasitas yang diinginkan yang Anda tentukan dalam tindakan tidak berada di luar batas ini.

Jadwal berulang

Untuk membuat jadwal berulang menggunakan AWS CLI atau SDK, tentukan ekspresi cron dan zona waktu untuk menjelaskan kapan tindakan terjadwal itu akan berulang. Anda dapat secara opsional menentukan tanggal dan waktu untuk waktu mulai, waktu akhir, atau keduanya.

Untuk membuat jadwal berulang menggunakan AWS Management Console, tentukan pola pengulangan, zona waktu, waktu mulai, dan waktu akhir opsional dari tindakan terjadwal Anda. Semua opsi pola pengulangan didasarkan pada ekspresi cron. Atau, Anda dapat menulis ekspresi cron kustom Anda sendiri.

Format ekspresi cron yang didukung terdiri dari lima bidang yang dipisahkan oleh spasi putih: [Minute] [Hour] [Day_of_month] [Month_of_year] [day_of_week]. Misalnya, ekspresi cron `30 6 * * 2` mengonfigurasi tindakan terjadwal yang berulang setiap hari Selasa pukul 6:30 pagi. Tanda bintang digunakan sebagai wildcard untuk mencocokkan semua nilai untuk bidang. Untuk contoh ekspresi cron lainnya, lihat <https://crontab.guru/examples.html>. Untuk informasi tentang menulis ekspresi cron Anda sendiri dalam format ini, lihat [Crontab](#).

Pilih waktu mulai dan akhir Anda dengan hati-hati. Ingatlah hal-hal berikut ini:

- Jika Anda menentukan waktu mulai, Auto Scaling Amazon EC2 melakukan tindakan saat ini, lalu melakukan tindakan berdasarkan pengulangan yang ditentukan.
- Jika Anda menentukan waktu akhir, tindakan berhenti berulang setelah waktu ini. Tindakan terjadwal tidak tersimpan di akun Anda setelah mencapai waktu akhir.
- Waktu mulai dan waktu akhir harus diatur dalam UTC saat Anda menggunakan AWS CLI atau SDK.

Zona waktu

Secara default, jadwal berulang yang Anda tetapkan berada di Coordinated Universal Time (UTC). Anda dapat mengubah zona waktu agar sesuai dengan zona waktu lokal Anda atau zona waktu untuk bagian lain dari jaringan Anda. Ketika Anda menentukan zona waktu yang mengamati Daylight Saving Time (DST), tindakan secara otomatis menyesuaikan untuk DST.

Nilai yang valid adalah nama kanonik untuk zona waktu dari database Zona Waktu Internet Assigned Numbers Authority (IANA). Misalnya, waktu Timur AS secara kanonik diidentifikasi sebagai `America/New_York`. Untuk informasi lebih lanjut, lihat <https://www.iana.org/time-zones>.

Zona waktu berbasis lokasi seperti menyesuaikan `America/New_York` secara otomatis untuk DST. Namun, zona waktu berbasis UTC seperti `Etc/UTC` adalah waktu absolut dan tidak akan menyesuaikan untuk DST.

Misalnya, Anda memiliki jadwal berulang yang zona waktunya `America/New_York`. Tindakan penskalaan pertama terjadi di zona `America/New_York` waktu sebelum DST dimulai. Tindakan penskalaan berikutnya terjadi di zona `America/New_York` waktu setelah DST dimulai. Aksi pertama dimulai pukul 8:00 UTC-5 di waktu setempat, sedangkan yang kedua dimulai pukul 8:00 UTC-4 pada waktu setempat.

Jika Anda membuat tindakan terjadwal menggunakan AWS Management Console dan menentukan zona waktu yang mengamati DST, jadwal berulang dan waktu mulai dan akhir secara otomatis menyesuaikan untuk DST.

Pertimbangan

Saat Anda membuat tindakan terjadwal, ingatlah hal berikut:

- Urutan pelaksanaan untuk tindakan terjadwal dijamin dalam grup yang sama, tetapi tidak untuk tindakan terjadwal di seluruh grup.
- Tindakan terjadwal umumnya berjalan dalam hitungan detik. Namun, tindakan tersebut mungkin tertunda hingga selama dua menit dari waktu mulai yang sudah dijadwalkan. Karena tindakan terjadwal dalam grup Auto Scaling dijalankan sesuai urutan yang ditentukan, tindakan dengan waktu mulai yang dijadwalkan berdekatan satu sama lain dapat memakan waktu lebih lama untuk dijalankan.
- Anda dapat mematikan sementara penskalaan terjadwal untuk grup Auto Scaling dengan menanggukkan `ScheduledActions` proses. Ini membantu Anda mencegah tindakan terjadwal

agar tidak aktif tanpa harus menghapusnya. Anda kemudian dapat melanjutkan penskalaan terjadwal ketika Anda ingin menggunakannya lagi. Untuk informasi selengkapnya, lihat [Menangguhkan dan melanjutkan proses Auto Scaling Amazon EC2](#).

- Setelah membuat tindakan terjadwal, Anda dapat memperbarui pengaturannya kecuali namanya.

Buat tindakan terjadwal

Untuk membuat tindakan terjadwal untuk grup Auto Scaling Anda, gunakan salah satu metode berikut:

Console

Untuk membuat tindakan terjadwal

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Auto scaling, di Tindakan terjadwal, pilih Buat tindakan terjadwal.
4. Masukkan Nama untuk tindakan yang dijadwalkan.
5. Untuk kapasitas yang diinginkan, Min, Max, pilih kapasitas baru yang diinginkan dari grup dan batas ukuran minimum dan maksimum yang baru. Kapasitas yang diinginkan harus sama dengan atau lebih besar dari ukuran kelompok minimum, dan sama dengan atau kurang dari ukuran kelompok maksimum.
6. Untuk Recurrence, pilih salah satu opsi yang tersedia.
 - Jika Anda ingin menskalakan pada jadwal berulang, pilih seberapa sering Amazon EC2 Auto Scaling harus menjalankan tindakan terjadwal.
 - Jika Anda memilih opsi yang dimulai dengan Setiap, ekspresi cron akan dibuat untuk Anda.
 - Jika Anda memilih Cron, masukkan ekspresi cron yang menentukan kapan harus melakukan tindakan.
 - Jika Anda ingin menskalakan hanya sekali, pilih Sekali.
7. Untuk zona waktu, pilih zona waktu. Nilai default-nya Etc/UTC.

Semua zona waktu yang tercantum berasal dari database Zona Waktu IANA. Untuk informasi lebih lanjut, lihat https://en.wikipedia.org/wiki/List_of_tz_database_time_zones.

8. Tentukan tanggal dan waktu untuk waktu mulai spesifik.
 - Jika Anda memilih jadwal berulang, waktu mulai menentukan kapan tindakan terjadwal pertama dalam seri berulang berjalan.
 - Jika Anda memilih Once sebagai pengulangan, waktu mulai menentukan tanggal dan waktu untuk menjalankan tindakan jadwal.
9. (Opsional) Untuk jadwal berulang, Anda dapat menentukan waktu akhir dengan memilih Atur Waktu Akhir dan kemudian memilih tanggal dan waktu untuk Akhir oleh.
10. Pilih Buat. Konsol menampilkan tindakan terjadwal untuk grup Auto Scaling.

AWS CLI

Untuk membuat tindakan terjadwal, Anda dapat menggunakan salah satu perintah contoh berikut. Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri.

Contoh: Untuk skala satu kali saja

Gunakan perintah [put-scheduled-update-group-action](#) berikut dengan `--desired-capacity` opsi `--start-time "YYYY-MM-DDThh:mm:ssZ"` dan.

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-one-time-action \  
  --auto-scaling-group-name my-asg --start-time "2021-03-31T08:00:00Z" --desired-capacity 3
```

Contoh: Untuk menjadwalkan penskalaan pada jadwal berulang

Gunakan perintah [put-scheduled-update-group-action](#) berikut dengan `--desired-capacity` opsi `--recurrence "cron expression"` dan.

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-recurring-action \  
  --auto-scaling-group-name my-asg --recurrence "0 9 * * *" --desired-capacity 3
```

Secara default, Auto Scaling Amazon EC2 menjalankan jadwal pengulangan yang ditentukan berdasarkan zona waktu UTC. Untuk menentukan zona waktu yang berbeda, sertakan `--time-zone` opsi dan nama zona waktu IANA, seperti pada contoh berikut.

```
--time-zone "America/New_York"
```

Untuk informasi lebih lanjut, lihat https://en.wikipedia.org/wiki/List_of_tz_database_time_zones.

Lihat detail tindakan terjadwal

Untuk melihat detail tindakan terjadwal yang akan datang untuk grup Auto Scaling Anda, gunakan salah satu metode berikut:

Console

Untuk melihat detail tindakan terjadwal

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih grup Auto Scaling Anda.
3. Pada tab Penskalaan otomatis, di bagian Tindakan terjadwal, Anda dapat melihat tindakan terjadwal yang akan datang.

Perhatikan bahwa konsol menampilkan nilai untuk Waktu mulai dan Waktu akhir di waktu lokal Anda dengan offset UTC berlaku pada tanggal dan waktu yang ditentukan. Offset UTC adalah perbedaan, dalam jam dan menit, dari waktu setempat ke UTC. Nilai untuk zona Waktu menunjukkan zona waktu yang Anda minta, misalnya, `America/New_York`.

AWS CLI

Gunakan perintah berikut [describe-scheduled-actions](#).

```
aws autoscaling describe-scheduled-actions --auto-scaling-group-name my-asg
```

Jika berhasil, perintah ini mengembalikan output yang serupa dengan yang berikut.

```
{
  "ScheduledUpdateGroupActions": [
```

```
{
  "AutoScalingGroupName": "my-asg",
  "ScheduledActionName": "my-recurring-action",
  "Recurrence": "30 0 1 1,6,12 *",
  "ScheduledActionARN": "arn:aws:autoscaling:us-
west-2:123456789012:scheduledUpdateGroupAction:8e86b655-b2e6-4410-8f29-
b4f094d6871c:autoScalingGroupName/my-asg:scheduledActionName/my-recurring-action",
  "StartTime": "2020-12-01T00:30:00Z",
  "Time": "2020-12-01T00:30:00Z",
  "MinSize": 1,
  "MaxSize": 6,
  "DesiredCapacity": 4
}
]
```

Verifikasi aktivitas penskalaan

Untuk memverifikasi aktivitas penskalaan yang terkait dengan penskalaan terjadwal, lihat.

[Memverifikasi aktivitas penskalaan untuk grup Auto Scaling](#)

Hapus tindakan terjadwal

Untuk menghapus tindakan terjadwal, gunakan salah satu metode berikut:

Console

Untuk menghapus tindakan terjadwal

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih grup Auto Scaling Anda.
3. Pada tab Auto scaling, di Tindakan terjadwal, pilih tindakan terjadwal.
4. Pilih Tindakan, Hapus.
5. Ketika diminta konfirmasi, pilih Ya, Hapus.

AWS CLI

Gunakan perintah berikut [delete-scheduled-action](#).

```
aws autoscaling delete-scheduled-action --auto-scaling-group-name my-asg \  
--scheduled-action-name my-recurring-action
```

Batasan

- Nama-nama tindakan terjadwal harus unik per grup Auto Scaling.
- Tindakan terjadwal harus memiliki nilai waktu yang unik. Jika Anda mencoba menjadwalkan aktivitas pada saat aktivitas penskalaan lain sudah dijadwalkan, panggilan ditolak dan menampilkan kesalahan yang menunjukkan bahwa tindakan terjadwal dengan waktu mulai terjadwal ini sudah ada.
- Anda dapat membuat maksimal 125 tindakan terjadwal untuk setiap grup Auto Scaling.

Penskalaan dinamis untuk Amazon EC2 Auto Scaling

Penskalaan dinamis menskalakan kapasitas grup Auto Scaling Anda saat terjadi perubahan lalu lintas.

Auto Scaling Amazon EC2 mendukung jenis kebijakan penskalaan dinamis berikut:

- Penskalaan pelacakan target —Tingkatkan dan kurangi kapasitas grup saat ini berdasarkan CloudWatch metrik Amazon dan nilai target. Ini bekerja mirip dengan cara termostat Anda mempertahankan suhu rumah Anda—Anda memilih suhu dan termostat melakukan sisanya.
- Penskalaan langkah — Meningkatkan dan mengurangi kapasitas grup saat ini berdasarkan serangkaian penyesuaian penskalaan, yang dikenal sebagai penyesuaian langkah, yang bervariasi berdasarkan ukuran pelanggaran alarm.
- Penskalaan sederhana —Meningkatkan dan mengurangi kapasitas grup saat ini berdasarkan penyesuaian penskalaan tunggal, dengan periode cooldown antara setiap aktivitas penskalaan.

Kami sangat menyarankan agar Anda menggunakan kebijakan penskalaan pelacakan target dan memilih metrik yang berubah berbanding terbalik dengan perubahan kapasitas grup Auto Scaling Anda. Jadi, jika Anda menggandakan ukuran grup Auto Scaling Anda, metriknya berkurang 50 persen. Hal ini memungkinkan data metrik untuk secara akurat memicu peristiwa penskalaan proporsional. Termasuk metrik seperti pemanfaatan CPU rata-rata atau jumlah permintaan rata-rata per target.

Dengan pelacakan target, grup Auto Scaling Anda menskalakan dalam proporsi langsung dengan beban aktual pada aplikasi Anda. Itu berarti bahwa selain memenuhi kebutuhan mendesak akan kapasitas dalam menanggapi perubahan beban, kebijakan pelacakan target juga dapat beradaptasi dengan perubahan beban yang terjadi dari waktu ke waktu, misalnya, karena variasi musim.

Kebijakan pelacakan target juga menghilangkan kebutuhan untuk menentukan CloudWatch alarm dan penyesuaian penskalaan secara manual. Auto Scaling Amazon EC2 menangani ini secara otomatis berdasarkan target yang Anda tetapkan.

Daftar Isi

- [Cara kerja kebijakan penskalaan dinamis](#)
- [Beberapa kebijakan penskalaan dinamis](#)
- [Kebijakan penskalaan pelacakan target untuk Amazon EC2 Auto Scaling](#)
- [Kebijakan penskalaan langkah dan sederhana untuk Amazon EC2 Auto Scaling](#)
- [Menskalakan jeda pakai untuk Amazon EC2 Auto Scaling](#)
- [Penskalaan berdasarkan Amazon SQS](#)
- [Memverifikasi aktivitas penskalaan untuk grup Auto Scaling](#)
- [Menonaktifkan kebijakan penskalaan untuk grup Auto Scaling](#)
- [Menghapus kebijakan penskalaan](#)
- [Contoh kebijakan penskalaan untuk AWS Command Line Interface \(AWS CLI\)](#)

Cara kerja kebijakan penskalaan dinamis

Kebijakan penskalaan dinamis menginstruksikan Auto Scaling Amazon EC2 untuk melacak metrik CloudWatch tertentu, dan menentukan tindakan apa yang harus diambil saat CloudWatch alarm terkait ada di ALARM. Metrik yang digunakan untuk memanggil status alarm adalah agregasi metrik yang berasal dari semua instance dalam grup Auto Scaling. (Misalnya, katakanlah Anda memiliki grup Auto Scaling dengan dua instance di mana satu instance berada pada CPU 60 persen dan yang lainnya pada CPU 40 persen. Rata-rata, mereka berada di CPU 50 persen.) Ketika kebijakan diberlakukan, Auto Scaling Amazon EC2 menyesuaikan kapasitas yang diinginkan grup naik atau turun saat ambang alarm dilanggar.

Ketika kebijakan penskalaan dinamis dipanggil, jika penghitungan kapasitas menghasilkan angka di luar rentang ukuran minimum dan maksimum grup, Auto Scaling Amazon EC2 memastikan bahwa kapasitas baru tidak pernah melampaui batas ukuran minimum dan maksimum. Kapasitas diukur dengan salah satu dari dua cara: menggunakan unit yang sama yang Anda pilih saat Anda mengatur

kapasitas yang diinginkan dalam hal instance, atau menggunakan unit kapasitas (jika [bobot instance](#) diterapkan).

- Contoh 1: Grup Auto Scaling memiliki kapasitas maksimum 3, kapasitas saat ini 2, dan kebijakan penskalaan dinamis yang menambahkan 3 instans. Saat menjalankan kebijakan ini, Auto Scaling Amazon EC2 hanya menambahkan 1 instans ke grup untuk mencegah grup melebihi ukuran maksimumnya.
- Contoh 2: Grup Auto Scaling memiliki kapasitas minimum 2, kapasitas saat ini 3, dan kebijakan penskalaan dinamis yang menghapus 2 instans. Saat menjalankan kebijakan ini, Auto Scaling Amazon EC2 hanya menghapus 1 instans dari grup untuk mencegah grup menjadi kurang dari ukuran minimumnya.

Ketika kapasitas yang diinginkan mencapai batas ukuran maksimum, penskalaan berhenti. Jika permintaan turun dan kapasitas turun, Amazon EC2 Auto Scaling dapat kembali menaikkan skala.

Pengecualiannya adalah saat Anda menggunakan bobot instance. Dalam instance ini, Amazon EC2 Auto Scaling dapat menskalakan di atas batas ukuran maksimum, tetapi hanya sampai dengan bobot maksimal yang Anda inginkan. Tujuannya adalah untuk sedekat mungkin dengan kapasitas baru yang diinginkan tetapi tetap mematuhi strategi alokasi yang ditentukan untuk grup. Strategi alokasi menentukan jenis instance yang akan diluncurkan. Bobot tersebut menentukan berapa banyak unit kapasitas yang dikontribusikan tiap-tiap instance pada kapasitas yang diinginkan grup berdasarkan tipe instance.

- Contoh 3: Grup Auto Scaling memiliki kapasitas maksimum 12, kapasitas saat ini 10, dan kebijakan penskalaan dinamis yang menambahkan 5 unit kapasitas. Jenis instance memiliki satu dari tiga bobot yang ditetapkan: 1, 4, atau 6. Saat menjalankan kebijakan, Amazon EC2 Auto Scaling memilih untuk meluncurkan jenis instans dengan bobot 6 berdasarkan strategi alokasi. Hasil dari peristiwa peningkatan skala ini adalah grup dengan kapasitas yang diinginkan sebesar 12 dan kapasitas saat ini sebesar 16.

Beberapa kebijakan penskalaan dinamis

Dalam sebagian besar kasus, kebijakan penskalaan pelacakan target cukup untuk mengonfigurasi grup Auto Scaling Anda untuk menaikkan atau menurunkan skala secara otomatis. Kebijakan penskalaan pelacakan target memungkinkan Anda memilih hasil yang diinginkan dan meminta grup Auto Scaling menambahkan dan menghapus instance sesuai kebutuhan untuk mencapai hasil tersebut.

Untuk konfigurasi penskalaan tingkat lanjut, grup Auto Scaling Anda dapat memiliki lebih dari satu kebijakan penskalaan. Misalnya, Anda dapat menentukan satu atau lebih kebijakan penskalaan pelacakan target, satu atau lebih kebijakan penskalaan langkah, atau keduanya. Ini memberikan fleksibilitas yang lebih besar untuk mencakup beberapa skenario.

Untuk mengilustrasikan cara kerja bersama beberapa kebijakan penskalaan dinamis, pertimbangkan aplikasi yang menggunakan grup Auto Scaling dan antrean Amazon SQS untuk mengirim permintaan ke satu instans EC2. Untuk membantu memastikan bahwa aplikasi bekerja pada tingkat optimal, ada dua kebijakan yang mengontrol ketika grup Auto Scaling harus ditingkatkan. Salah satunya adalah kebijakan pelacakan target yang menggunakan metrik khusus untuk menambah dan menghapus kapasitas berdasarkan jumlah pesan SQS dalam antrean. Yang lainnya adalah kebijakan penskalaan langkah yang menggunakan CloudWatch `CPUUtilization` metrik Amazon untuk menambah kapasitas ketika instance melebihi 90 persen pemanfaatan untuk jangka waktu tertentu.

Jika ada beberapa kebijakan yang berlaku pada saat yang sama, ada kemungkinan bahwa setiap kebijakan dapat menginstruksikan grup Auto Scaling untuk meningkatkan skala (atau menurunkan skala) pada saat yang sama. Misalnya, ada kemungkinan bahwa `CPUUtilization` metrik melonjak dan melanggar ambang CloudWatch alarm pada saat yang sama ketika metrik kustom SQS melonjak dan melanggar ambang alarm metrik khusus.

Jika situasi ini terjadi, Amazon EC2 Auto Scaling memilih kebijakan yang menyediakan kapasitas terbesar untuk menaikkan dan menurunkan skala. Misalkan, kebijakan untuk `CPUUtilization` meluncurkan satu instance, sementara kebijakan untuk antrean SQS meluncurkan dua instance. Jika kriteria peningkatan skala untuk kedua kebijakan terpenuhi pada saat yang sama, Amazon EC2 Auto Scaling memberikan prioritas pada kebijakan antrean SQS. Hal ini menghasilkan grup Auto Scaling yang meluncurkan dua instance.

Pendekatan pemberian prioritas terhadap kebijakan yang memberikan kapasitas terbesar berlaku bahkan ketika kebijakan menggunakan kriteria yang berbeda untuk penskalaan. Misalnya, jika satu kebijakan menghentikan tiga instance, kebijakan lain mengurangi jumlah instance sebesar 25 persen, dan grup memiliki delapan instance pada saat menurunkan skala, Amazon EC2 Auto Scaling memberikan prioritas pada kebijakan yang menyediakan jumlah instance terbesar untuk grup tersebut. Ini menghasilkan grup Auto Scaling yang menghentikan dua instance (25 persen dari $8 = 2$). Tujuannya adalah untuk mencegah Amazon EC2 Auto Scaling menghapus terlalu banyak instance.

Namun, kami menyarankan Anda berhati-hati ketika menggunakan kebijakan penskalaan pelacakan target dengan kebijakan penskalaan langkah karena adanya konflik di antara kebijakan ini dapat menyebabkan perilaku yang tidak diinginkan. Misalnya, jika kebijakan penskalaan langkah memulai

aktivitas penurunan skala sebelum kebijakan pelacakan target siap untuk diterapkan, aktivitas penurunan skala tidak akan diblokir. Setelah aktivitas peningkatan skala selesai, kebijakan pelacakan target dapat menginstruksikan grup untuk kembali meningkatkan skala.

Kebijakan penskalaan pelacakan target untuk Amazon EC2 Auto Scaling

Kebijakan penskalaan pelacakan target secara otomatis menskalakan kapasitas grup Auto Scaling Anda berdasarkan nilai metrik target. Hal ini memungkinkan aplikasi Anda untuk mempertahankan kinerja optimal dan efisiensi biaya tanpa intervensi manual.

Dengan pelacakan target, Anda memilih metrik dan nilai target untuk mewakili tingkat pemanfaatan atau throughput rata-rata yang ideal untuk aplikasi Anda. Auto Scaling Amazon EC2 membuat dan mengelola alarm yang memanggil peristiwa CloudWatch penskalaan saat metrik menyimpang dari target. Sebagai contoh, ini mirip dengan bagaimana termostat mempertahankan suhu target.

Misalnya, katakanlah Anda saat ini memiliki aplikasi yang berjalan pada dua instance, dan Anda ingin pemanfaatan CPU dari grup Auto Scaling tetap sekitar 50 persen saat beban pada aplikasi berubah. Ini memberi Anda kapasitas ekstra untuk menangani lonjakan lalu lintas tanpa mempertahankan jumlah berlebih dari sumber daya yang tidak aktif.

Anda dapat memenuhi kebutuhan ini dengan membuat kebijakan penskalaan pelacakan target yang menargetkan pemanfaatan CPU rata-rata 50 persen. Kemudian, grup Auto Scaling Anda akan meningkatkan skala, atau meningkatkan kapasitas, ketika CPU melebihi 50 persen untuk menangani peningkatan beban. Ini akan menskalakan, atau mengurangi kapasitas, ketika CPU turun di bawah 50 persen untuk mengoptimalkan biaya selama periode pemanfaatan rendah.

Topik

- [Beberapa kebijakan penskalaan pelacakan target](#)
- [Pilih metrik](#)
- [Tentukan nilai target](#)
- [Tentukan waktu pemanasan instance](#)
- [Pertimbangan](#)
- [Buat kebijakan penskalaan pelacakan target](#)
- [Membuat kebijakan penskalaan pelacakan target untuk Auto Scaling Amazon EC2 menggunakan matematika metrik](#)

Beberapa kebijakan penskalaan pelacakan target

Untuk membantu mengoptimalkan kinerja penskalaan, Anda dapat menggunakan beberapa kebijakan penskalaan pelacakan target secara bersamaan, asalkan masing-masing menggunakan metrik yang berbeda. Misalnya, pemanfaatan dan throughput dapat saling mempengaruhi. Setiap kali salah satu metrik ini berubah, biasanya menyiratkan bahwa metrik lain juga akan terpengaruh. Oleh karena itu, penggunaan beberapa metrik memberikan informasi tambahan tentang beban yang dimiliki grup Auto Scaling Anda. Ini dapat membantu Auto Scaling Amazon EC2 membuat keputusan yang lebih tepat saat menentukan berapa banyak kapasitas yang akan ditambahkan ke grup Anda.

Tujuan Auto Scaling Amazon EC2 adalah untuk selalu memprioritaskan ketersediaan. Ini akan memperkecil grup Auto Scaling jika ada kebijakan pelacakan target yang siap untuk diskalakan. Ini akan diskalakan hanya jika semua kebijakan pelacakan target (dengan bagian penskalaan diaktifkan) siap untuk diskalakan.

Pilih metrik

Anda dapat membuat kebijakan penskalaan pelacakan target dengan metrik yang telah ditentukan sebelumnya atau metrik khusus.

Saat membuat kebijakan penskalaan pelacakan target dengan tipe metrik yang telah ditentukan sebelumnya, Anda memilih satu metrik dari daftar metrik yang telah ditentukan berikut:

- `ASGAverageCPUUtilization`—Penggunaan CPU rata-rata grup Auto Scaling.
- `ASGAverageNetworkIn` Rata-rata jumlah byte yang diterima oleh satu instance pada semua antarmuka jaringan.
- `ASGAverageNetworkOut` Rata-rata jumlah byte yang dikirim keluar dari satu instance pada semua antarmuka jaringan.
- `ALBRequestCountPerTarget`—Rata-rata jumlah permintaan Application Load Balancer per target.

Important

Informasi berharga lainnya tentang metrik untuk pemanfaatan CPU, I/O jaringan, dan jumlah permintaan Application Load Balancer per target dapat ditemukan di [Daftar metrik yang CloudWatch tersedia untuk topik instans Anda di Panduan Pengguna Amazon EC2 untuk](#)

Instans Linux dan CloudWatch metrik untuk topik Application Load Balancer Anda di Panduan Pengguna [untuk Penyeimbang Beban Aplikasi](#), masing-masing.

Anda dapat memilih CloudWatch metrik lain yang tersedia atau metrik Anda sendiri CloudWatch dengan menentukan metrik khusus. Anda harus menggunakan AWS CLI atau SDK untuk membuat kebijakan pelacakan target dengan spesifikasi metrik yang disesuaikan. Untuk contoh yang menentukan spesifikasi metrik yang disesuaikan untuk kebijakan penskalaan pelacakan target menggunakan AWS CLI, lihat. [Contoh kebijakan penskalaan untuk AWS Command Line Interface \(AWS CLI\)](#)

Ingatlah hal-hal berikut ini saat memilih metrik:

- Kami menyarankan Anda hanya menggunakan metrik yang tersedia pada interval satu menit untuk membantu Anda menskalakan lebih cepat dalam menanggapi perubahan pemanfaatan. Pelacakan target akan mengevaluasi metrik yang dikumpulkan pada perincian satu menit untuk semua metrik dan metrik khusus yang telah ditentukan sebelumnya, tetapi metrik yang mendasarinya mungkin lebih jarang mempublikasikan data. Misalnya, semua metrik Amazon EC2 dikirim dalam interval lima menit secara default, tetapi dapat dikonfigurasi hingga satu menit (dikenal sebagai pemantauan terperinci). Pilihan ini terserah layanan individu. Sebagian besar mencoba menggunakan interval sekecil mungkin. Untuk informasi tentang mengaktifkan pemantauan terperinci, lihat [Konfigurasi pemantauan untuk instans Auto Scaling](#).
- Tidak semua metrik khusus berfungsi untuk pelacakan target. Metrik harus berupa metrik penggunaan yang valid dan menjelaskan seberapa sibuk instance. Nilai metrik harus meningkatkan atau menurunkan secara proporsional jumlah instance dalam grup Auto Scaling. Dengan demikian, data metrik dapat digunakan untuk meningkatkan atau menurunkan jumlah instance secara proporsional. Misalnya, pemanfaatan CPU pekerjaan grup Auto Scaling (yaitu, CPUUtilization metrik Amazon EC2 dengan AutoScalingGroupName dimensi metrik), jika beban pada grup Auto Scaling didistribusikan di seluruh instance.
- Metrik berikut tidak berfungsi untuk pelacakan target:
 - Jumlah permintaan yang diterima oleh load balancer di depan grup Auto Scaling (yaitu, metrik Elastic Load Balancing RequestCount). Jumlah permintaan yang diterima oleh load balancer tidak berubah berdasarkan penggunaan grup Auto Scaling.
 - Latensi permintaan penyeimbang muatan (yaitu, metrik Elastic Load Balancing Latency). Permintaan latensi dapat meningkat berdasarkan peningkatan pemanfaatan, tetapi tidak selalu berubah secara proporsional.

- Metrik antrian CloudWatch Amazon SQS. `ApproximateNumberOfMessagesVisible` Jumlah pesan dalam antrian mungkin tidak berubah secara proporsional dengan ukuran grup Auto Scaling yang memproses pesan dari antrian. Namun, metrik kustom yang mengukur jumlah pesan dalam antrian per instans EC2 di grup Auto Scaling dapat berfungsi. Untuk informasi selengkapnya, lihat [Penskalaan berdasarkan Amazon SQS](#).
- Untuk menggunakan `ALBRequestCountPerTarget` metrik, Anda harus menentukan `ResourceLabel` parameter untuk mengidentifikasi kelompok target penyeimbang beban yang terkait dengan metrik. Untuk contoh yang menentukan `ResourceLabel` parameter untuk kebijakan penskalaan pelacakan target menggunakan AWS CLI, lihat. [Contoh kebijakan penskalaan untuk AWS Command Line Interface \(AWS CLI\)](#)
- Ketika metrik memancarkan nilai 0 nyata ke CloudWatch (misalnya, `ALBRequestCountPerTarget`), grup Auto Scaling dapat menskalakan ke 0 ketika tidak ada lalu lintas ke aplikasi Anda untuk jangka waktu yang berkelanjutan. Agar grup Auto Scaling Anda masuk ke 0 saat tidak ada permintaan yang dirutekan, kapasitas minimum grup harus disetel ke 0.
- Alih-alih menerbitkan metrik baru untuk digunakan dalam kebijakan penskalaan, Anda dapat menggunakan matematika metrik untuk menggabungkan metrik yang ada. Untuk informasi selengkapnya, lihat [Membuat kebijakan penskalaan pelacakan target untuk Auto Scaling Amazon EC2 menggunakan matematika metrik](#).

Tentukan nilai target

Saat membuat kebijakan penskalaan pelacakan target, Anda harus menentukan nilai target. Nilai target mewakili pemanfaatan atau throughput rata-rata optimal untuk grup Auto Scaling. Untuk menggunakan biaya sumber daya secara efisien, tetapkan nilai target setinggi mungkin dengan buffer yang masuk akal untuk peningkatan lalu lintas yang tidak terduga. Ketika aplikasi Anda diskalakan secara optimal untuk arus lalu lintas normal, nilai metrik sebenarnya harus berada pada atau tepat di bawah nilai target.

Ketika kebijakan penskalaan didasarkan pada throughput, seperti jumlah permintaan per target untuk Application Load Balancer, I/O jaringan, atau metrik hitungan lainnya, nilai target mewakili throughput rata-rata optimal dari satu instans, untuk periode satu menit.

Tentukan waktu pemanasan instance

Anda dapat secara opsional menentukan jumlah detik yang diperlukan untuk pemanasan instance yang baru diluncurkan. Sampai waktu pemanasan yang ditentukan telah kedaluwarsa, instance tidak dihitung terhadap metrik instans EC2 agregat dari grup Auto Scaling.

Sementara kasus berada dalam periode pemanasan, kebijakan penskalaan Anda hanya akan diperkecil jika nilai metrik dari instans yang tidak memanans lebih besar daripada penggunaan target kebijakan.

Jika kelompok menskalakan lagi, contoh yang masih memanans dihitung sebagai bagian dari kapasitas yang diinginkan untuk aktivitas penskalaan berikutnya. Tujuannya adalah untuk terus (tetapi tidak berlebihan) menskalakan ke luar.

Saat aktivitas penskalaan sedang berlangsung, semua aktivitas penskalaan yang dimulai oleh kebijakan penskalaan diblokir hingga instans selesai memanans. Ketika contoh selesai pemanasan, jika peristiwa skala dalam terjadi, setiap contoh yang saat ini dalam proses penghentian akan dihitung terhadap kapasitas grup saat ini ketika menghitung kapasitas baru yang diinginkan. Oleh karena itu, kami tidak menghapus lebih banyak instance daripada yang diperlukan oleh grup Auto Scaling.

Nilai default

Jika tidak ada nilai yang disetel, maka kebijakan penskalaan akan menggunakan nilai default, yang merupakan nilai untuk [pemanasan instance default](#) yang ditentukan untuk grup. [Jika pemanasan instance default adalah null, maka itu kembali ke nilai cooldown default](#). Sebaiknya gunakan pemanasan instans default untuk mempermudah pembaruan semua kebijakan penskalaan saat waktu pemanasan berubah.

Pertimbangan

Pertimbangan berikut berlaku saat bekerja dengan kebijakan penskalaan pelacakan target:

- Jangan membuat, mengedit, atau menghapus CloudWatch alarm yang digunakan dengan kebijakan penskalaan pelacakan target. Auto Scaling Amazon EC2 membuat dan mengelola CloudWatch alarm yang terkait dengan kebijakan penskalaan pelacakan target Anda dan menghapusnya saat tidak diperlukan lagi.
- Kebijakan penskalaan pelacakan target memprioritaskan ketersediaan selama periode tingkat lalu lintas yang berfluktuasi dengan menskalakan secara lebih bertahap ketika lalu lintas menurun. Jika ingin grup Auto Scaling segera masuk saat beban kerja selesai, Anda dapat menonaktifkan bagian

penskalaan kebijakan. Ini memberi Anda fleksibilitas untuk menggunakan metode scale-in yang paling sesuai dengan kebutuhan Anda saat pemanfaatannya rendah. Untuk memastikan bahwa skala terjadi secepat mungkin, sebaiknya jangan menggunakan kebijakan penskalaan sederhana untuk mencegah periode cooldown ditambahkan.

- Jika metrik kehilangan titik data, ini menyebabkan status CloudWatch alarm berubah menjadi `INSUFFICIENT_DATA`. Ketika ini terjadi, Auto Scaling Amazon EC2 tidak dapat menskalakan grup Anda hingga titik data baru ditemukan.
- Jika metrik jarang dilaporkan oleh desain, matematika metrik dapat membantu. Misalnya, untuk menggunakan nilai terbaru, maka gunakan `FILL(m1, REPEAT)` fungsi di `m1` mana metrik.
- Anda mungkin melihat kesenjangan antara nilai target dan titik data metrik aktualnya. Ini karena kami bertindak konservatif dengan membulatkan ke atas atau ke bawah saat menentukan berapa banyak instance untuk ditambahkan atau dihapus. Hal ini mencegah kami menambahkan jumlah instance yang tidak mencukupi atau menghilangkan terlalu banyak instance. Namun, untuk grup Auto Scaling yang lebih kecil dengan lebih sedikit instance, pemanfaatan grup mungkin tampak jauh dari nilai target. Misalnya, katakanlah Anda menetapkan nilai target sebesar 50 persen untuk pemanfaatan CPU dan grup Auto Scaling Anda kemudian melampaui target. Kami dapat menentukan bahwa penambahan 1,5 instance akan mengurangi penggunaan CPU hingga mendekati 50 persen. Karena tidak mungkin untuk menambahkan 1,5 instance, kami membulatkan ke atas dan menambahkan dua instance. Hal ini dapat mengurangi penggunaan CPU menjadi nilai di bawah 50 persen, tetapi memastikan bahwa aplikasi Anda memiliki sumber daya yang cukup untuk mendukungnya. Demikian pula, jika kami menentukan bahwa menghilangkan 1,5 instance meningkatkan penggunaan CPU Anda hingga di atas 50 persen, kami hanya menghapus satu instance.

Untuk grup Auto Scaling yang lebih besar dengan lebih banyak instance, penggunaan tersebut disebarkan pada sejumlah besar instance, di mana penambahan atau penghapusan instance menyebabkan lebih sedikit kesenjangan antara nilai target dan titik data metrik aktual.

- Kebijakan penskalaan pelacakan target mengasumsikan bahwa harus meningkatkan skala grup Auto Scaling Anda ketika metrik yang telah ditentukan berada di atas nilai target. Anda tidak dapat menggunakan kebijakan penskalaan pelacakan target untuk memperkecil grup Auto Scaling jika metrik yang ditentukan berada di bawah nilai target.

Buat kebijakan penskalaan pelacakan target

Untuk membuat kebijakan penskalaan pelacakan target untuk grup Auto Scaling Anda, gunakan salah satu metode berikut.

Sebelum memulai, konfirmasikan bahwa metrik pilihan Anda tersedia pada interval 1 menit (dibandingkan dengan interval 5 menit default metrik Amazon EC2).

Console

Untuk membuat kebijakan penskalaan pelacakan target untuk grup Auto Scaling baru

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih Buat grup Auto Scaling.
3. Pada Langkah 1, 2, dan 3, pilih opsi yang diinginkan dan lanjutkan ke Langkah 4: Konfigurasi ukuran grup dan kebijakan penskalaan.
4. Di bawah Penskalaan, tentukan rentang yang ingin Anda skalakan dengan memperbarui kapasitas Min yang diinginkan dan kapasitas Maks yang diinginkan. Kedua pengaturan ini memungkinkan grup Auto Scaling Anda untuk menskalakan secara dinamis. Untuk informasi selengkapnya, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).
5. Di bawah Penskalaan otomatis, pilih Kebijakan penskalaan pelacakan target.
6. Untuk menentukan kebijakan, lakukan hal berikut:
 - a. Tentukan nama untuk kebijakan tersebut.
 - b. Untuk Jenis metrik, pilih metrik.

Jika Anda memilih jumlah permintaan Application Load Balancer per target, pilih grup target di grup Target.
 - c. Tentukan Nilai target untuk metrik.
 - d. (Opsional) Untuk pemanasan Instance, perbarui nilai pemanasan instance sesuai kebutuhan.
 - e. (Opsional) Pilih Nonaktifkan penurunan skala untuk membuat hanya kebijakan peningkatan skala. Ini memungkinkan Anda membuat kebijakan penurunan skala terpisah dari jenis yang berbeda jika diinginkan.
7. Lanjutkan untuk membuat grup Auto Scaling. Kebijakan penskalaan Anda akan dibuat setelah grup Auto Scaling dibuat.

Untuk membuat kebijakan penskalaan pelacakan target untuk grup Auto Scaling yang ada

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

3. Verifikasi bahwa batas penskalaan ditetapkan dengan tepat. Misalnya, jika kapasitas yang diinginkan grup Anda sudah maksimal, Anda perlu menentukan maksimum baru untuk meningkatkan skala. Untuk informasi selengkapnya, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).
4. Pada tab Penskalaan otomatis, dalam Kebijakan penskalaan dinamis, pilih Buat kebijakan penskalaan dinamis.
5. Untuk menentukan kebijakan, lakukan hal berikut:
 - a. Untuk jenis Kebijakan, pertahankan default penskalaan pelacakan Target.
 - b. Tentukan nama untuk kebijakan tersebut.
 - c. Untuk Jenis metrik, pilih metrik. Anda hanya dapat memilih satu jenis metrik. Untuk menggunakan lebih dari satu metrik, buat beberapa kebijakan.

Jika Anda memilih jumlah permintaan Application Load Balancer per target, pilih grup target di grup Target.

- d. Tentukan Nilai target untuk metrik.
 - e. (Opsional) Untuk pemanasan Instance, perbarui nilai pemanasan instance sesuai kebutuhan.
 - f. (Opsional) Pilih Nonaktifkan penurunan skala untuk membuat hanya kebijakan peningkatan skala. Ini memungkinkan Anda membuat kebijakan penurunan skala terpisah dari jenis yang berbeda jika diinginkan.
6. Pilih Buat.

AWS CLI

Untuk membuat kebijakan penskalaan pelacakan target, Anda dapat menggunakan contoh berikut untuk membantu Anda memulai. Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri.

Note

Untuk contoh lainnya, lihat [Contoh kebijakan penskalaan untuk AWS Command Line Interface \(AWS CLI\)](#).

Untuk membuat kebijakan penskalaan pelacakan target (AWS CLI)

1. Gunakan cat perintah berikut untuk menyimpan nilai target untuk kebijakan penskalaan Anda dan spesifikasi metrik yang telah ditentukan sebelumnya dalam file JSON yang diberi nama `config.json` di direktori home Anda. Berikut ini adalah contoh konfigurasi pelacakan target yang menjaga pemanfaatan CPU rata-rata sebesar 50 persen.

```
$ cat ~/config.json
{
  "TargetValue": 50.0,
  "PredefinedMetricSpecification":
  {
    "PredefinedMetricType": "ASGAverageCPUUtilization"
  }
}
```

Untuk informasi selengkapnya, lihat [PredefinedMetricSpecification](#) di Referensi API Auto Scaling Amazon EC2.

2. Gunakan `put-scaling-policy` perintah, bersama dengan `config.json` file yang Anda buat di langkah sebelumnya, untuk membuat kebijakan penskalaan Anda.

```
aws autoscaling put-scaling-policy --policy-name cpu50-target-tracking-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
```

Jika berhasil, perintah ini mengembalikan ARN dan nama dari dua CloudWatch alarm yang dibuat atas nama Anda.

```
{
  "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:228f02c2-c665-4bfd-
```

```

aaac-8b04080bea3c:autoScalingGroupName/my-asg:policyName/cpu50-target-tracking-
scaling-policy",
  "Alarms": [
    {
      "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e",
      "AlarmName": "TargetTracking-my-asg-AlarmHigh-
fc0e4183-23ac-497e-9992-691c9980c38e"
    },
    {
      "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2",
      "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-
bd9e-471a352ee1a2"
    }
  ]
}

```

Membuat kebijakan penskalaan pelacakan target untuk Auto Scaling Amazon EC2 menggunakan matematika metrik

Menggunakan matematika metrik, Anda dapat menanyakan beberapa CloudWatch metrik dan menggunakan ekspresi matematika untuk membuat deret waktu baru berdasarkan metrik ini. Anda dapat memvisualisasikan deret waktu yang dihasilkan di CloudWatch konsol dan menambahkannya ke dasbor. Untuk informasi selengkapnya tentang matematika [metrik](#), lihat [Menggunakan matematika metrik](#) di Panduan CloudWatch Pengguna Amazon.

Pertimbangan berikut berlaku untuk ekspresi matematika metrik:

- Anda dapat menanyakan CloudWatch metrik apa pun yang tersedia. Setiap metrik adalah kombinasi unik dari nama metrik, namespace, dan nol atau lebih dimensi.
- Anda dapat menggunakan operator aritmatika (+ - */^), fungsi statistik (seperti AVG atau SUM), atau fungsi lain yang mendukung. CloudWatch
- Anda dapat menggunakan metrik dan hasil ekspresi matematika lainnya dalam rumus ekspresi matematika.
- Setiap ekspresi yang digunakan dalam spesifikasi metrik pada akhirnya harus mengembalikan satu deret waktu.

- Anda dapat memverifikasi bahwa ekspresi matematika metrik valid dengan menggunakan CloudWatch konsol atau CloudWatch [GetMetricData](#) API.

Note

Anda dapat membuat kebijakan penskalaan pelacakan target menggunakan matematika metrik hanya jika Anda menggunakan AWS CLI atau SDK. Fitur ini belum tersedia di konsol dan AWS CloudFormation.

Contoh: backlog antrian Amazon SQS per instance

Untuk menghitung backlog antrean Amazon SQS per instance, ambil perkiraan jumlah pesan yang tersedia untuk diambil dari antrian dan bagi nomor tersebut dengan kapasitas berjalan grup Auto Scaling, yang merupakan jumlah instance dalam status. InService Untuk informasi selengkapnya, lihat [Penskalaan berdasarkan Amazon SQS](#).

Logika untuk ekspresi adalah ini:

sum of (number of messages in the queue)/(number of InService instances)

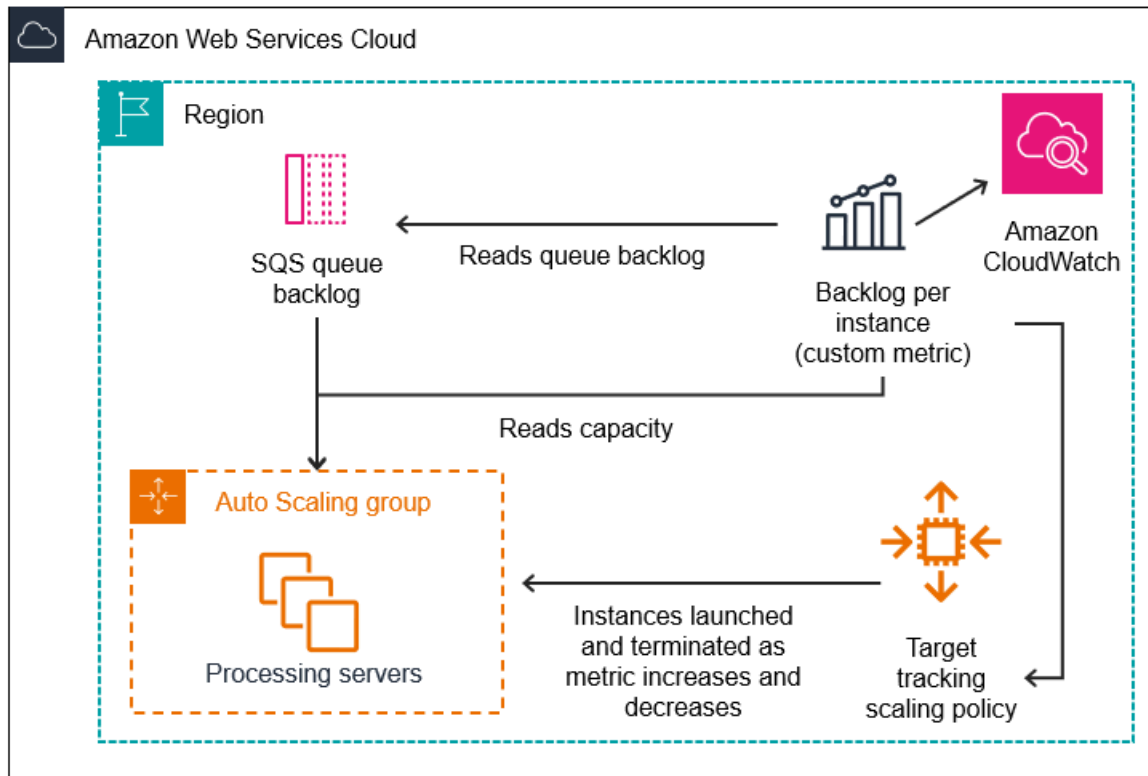
Maka informasi CloudWatch metrik Anda adalah sebagai berikut.

ID	CloudWatch metrik	Statistik	Periode
m1	ApproximateNumberOfMessagesVisible	Jumlah	1 menit
m2	GroupInServiceInstances	Rata-rata	1 menit

ID dan ekspresi matematika metrik Anda adalah sebagai berikut.

ID	Ekspresi
e1	(m1)/(m2)

Diagram berikut menggambarkan arsitektur untuk metrik ini:



Untuk menggunakan matematika metrik ini untuk membuat kebijakan penskalaan pelacakan target (AWS CLI)

1. Simpan ekspresi matematika metrik sebagai bagian dari spesifikasi metrik yang disesuaikan dalam file JSON bernama `config.json`.

Gunakan contoh berikut untuk membantu Anda memulai. Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri.

```
{
  "CustomizedMetricSpecification": {
    "Metrics": [
      {
        "Label": "Get the queue size (the number of messages waiting to be processed)",
        "Id": "m1",
        "MetricStat": {
          "Metric": {
            "MetricName": "ApproximateNumberOfMessagesVisible",
            "Namespace": "AWS/SQS",
```

```
        "Dimensions": [
            {
                "Name": "QueueName",
                "Value": "my-queue"
            }
        ],
        "Stat": "Sum"
    },
    "ReturnData": false
},
{
    "Label": "Get the group size (the number of InService instances)",
    "Id": "m2",
    "MetricStat": {
        "Metric": {
            "MetricName": "GroupInServiceInstances",
            "Namespace": "AWS/AutoScaling",
            "Dimensions": [
                {
                    "Name": "AutoScalingGroupName",
                    "Value": "my-asg"
                }
            ]
        },
        "Stat": "Average"
    },
    "ReturnData": false
},
{
    "Label": "Calculate the backlog per instance",
    "Id": "e1",
    "Expression": "m1 / m2",
    "ReturnData": true
}
],
"TargetValue": 100
}
```

Untuk informasi selengkapnya, lihat [TargetTrackingConfiguration](#) di Referensi API Auto Scaling Amazon EC2.

Note

Berikut adalah beberapa sumber daya tambahan yang dapat membantu Anda menemukan nama metrik, ruang nama, dimensi, dan statistik untuk CloudWatch metrik:

- Untuk informasi tentang metrik yang tersedia untuk AWS layanan, lihat [AWS layanan yang memublikasikan CloudWatch metrik](#) di CloudWatch Panduan Pengguna Amazon.
- [Untuk mendapatkan nama metrik, namespace, dan dimensi yang tepat \(jika ada\) untuk CloudWatch metrik dengan metrik AWS CLI, lihat daftar-metrik.](#)

2. Untuk membuat kebijakan ini, jalankan `put-scaling-policy` perintah menggunakan file JSON sebagai input, seperti yang ditunjukkan dalam contoh berikut.

```
aws autoscaling put-scaling-policy --policy-name sqs-backlog-target-tracking-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
```

Jika berhasil, perintah ini mengembalikan Amazon Resource Name (ARN) kebijakan dan ARN dari dua CloudWatch alarm yang dibuat atas nama Anda.

```
{
  "PolicyARN": "arn:aws:autoscaling:us-west-2:123456789012:scalingPolicy:228f02c2-c665-4bfd-aaac-8b04080bea3c:autoScalingGroupName/my-asg:policyName/sqs-backlog-target-tracking-scaling-policy",
  "Alarms": [
    {
      "AlarmARN": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-fc0e4183-23ac-497e-9992-691c9980c38e",
      "AlarmName": "TargetTracking-my-asg-AlarmHigh-fc0e4183-23ac-497e-9992-691c9980c38e"
    },
    {
      "AlarmARN": "arn:aws:cloudwatch:us-west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-bd9e-471a352ee1a2",

```

```
"AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-  
bd9e-471a352ee1a2"  
  }  
]  
}
```

Note

Jika perintah ini menimbulkan kesalahan, pastikan Anda telah memperbarui AWS CLI secara lokal ke versi terbaru.

Kebijakan penskalaan langkah dan sederhana untuk Amazon EC2 Auto Scaling

Kebijakan penskalaan langkah dan penskalaan sederhana menskalakan kapasitas grup Auto Scaling Anda dalam peningkatan yang telah ditentukan berdasarkan alarm. CloudWatch Anda dapat menentukan kebijakan penskalaan terpisah untuk menangani penskalaan (peningkatan kapasitas) dan penskalaan (penurunan kapasitas) saat ambang alarm dilanggar.

Dengan penskalaan langkah dan penskalaan sederhana, Anda membuat dan mengelola CloudWatch alarm yang menjalankan proses penskalaan. Saat alarm dilanggar, Auto Scaling Amazon EC2 memulai kebijakan penskalaan yang terkait dengan alarm tersebut.

Kami sangat menyarankan agar Anda menggunakan kebijakan penskalaan pelacakan target untuk menskalakan metrik seperti pemanfaatan CPU rata-rata atau jumlah permintaan rata-rata per target. Metrik yang menurun saat kapasitas meningkat dan meningkat ketika kapasitas menurun dapat digunakan untuk meningkatkan dan menurunkan jumlah instance secara proporsional menggunakan pelacakan target. Ini membantu memastikan Amazon EC2 Auto Scaling mengikuti kurva permintaan untuk aplikasi Anda dengan ketat. Untuk informasi selengkapnya, lihat [Kebijakan penskalaan pelacakan target](#).

Daftar Isi

- [Cara kerja kebijakan penskalaan langkah](#)
- [Penyesuaian langkah untuk penskalaan langkah](#)
- [Jenis penyesuaian penskalaan](#)
- [Pemanasan contoh](#)

- [Pertimbangan](#)
- [Buat kebijakan penskalaan langkah untuk skala keluar](#)
- [Buat kebijakan penskalaan langkah untuk skala di](#)
- [Kebijakan penskalaan sederhana](#)

Cara kerja kebijakan penskalaan langkah

Untuk menggunakan penskalaan langkah, pertama-tama Anda membuat CloudWatch alarm yang memantau metrik untuk grup Auto Scaling Anda. Tentukan metrik, nilai ambang batas, dan jumlah periode evaluasi yang menentukan pelanggaran alarm. Kemudian, buat kebijakan penskalaan langkah yang menentukan cara menskalakan grup Anda saat ambang alarm dilanggar.

Tambahkan penyesuaian langkah dalam kebijakan. Anda dapat menentukan penyesuaian langkah yang berbeda berdasarkan ukuran pelanggaran alarm. Sebagai contoh:

- Skalikan 10 instance jika metrik alarm mencapai 60 persen
- Skala hingga 30 instance jika metrik alarm mencapai 75 persen
- Skala hingga 40 instance jika metrik alarm mencapai 85 persen

Jika ambang batas alarm dilanggar untuk jumlah periode evaluasi yang ditentukan, Auto Scaling Amazon EC2 akan menerapkan penyesuaian langkah yang ditentukan dalam kebijakan. Penyesuaian dapat berlanjut untuk pelanggaran alarm tambahan hingga status alarm kembali ke OK.

Setiap instance memiliki periode pemanasan untuk mencegah aktivitas penskalaan menjadi terlalu reaktif terhadap perubahan yang terjadi selama periode waktu yang singkat. Anda dapat mengonfigurasi periode pemanasan secara opsional untuk kebijakan penskalaan Anda. Namun, sebaiknya gunakan pemanasan instans default untuk mempermudah pembaruan semua kebijakan penskalaan saat waktu pemanasan berubah. Untuk informasi selengkapnya, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#).

Kebijakan penskalaan sederhana mirip dengan kebijakan penskalaan langkah, kecuali kebijakan tersebut didasarkan pada penyesuaian penskalaan tunggal, dengan periode cooldown antara setiap aktivitas penskalaan. Untuk informasi selengkapnya, lihat [Kebijakan penskalaan sederhana](#).

Penyesuaian langkah untuk penskalaan langkah

Saat Anda membuat kebijakan penskalaan langkah, Anda menetapkan satu penyesuaian langkah atau lebih yang secara otomatis menskalakan jumlah instance secara dinamis berdasarkan ukuran pelanggaran alarm. Setiap penyesuaian langkah menentukan hal berikut:

- Batas bawah untuk nilai metrik
- Batas atas untuk nilai metrik
- Jumlah yang ditentukan untuk diskalakan, berdasarkan jenis penyesuaian penskalaan

CloudWatch mengumpulkan titik data metrik berdasarkan statistik untuk metrik yang terkait dengan CloudWatch alarm Anda. Ketika alarm dilanggar, kebijakan penskalaan yang sesuai akan dipanggil. Auto Scaling Amazon EC2 menerapkan jenis agregasi ke titik data metrik terbaru CloudWatch dari (sebagai lawan dari data metrik mentah). Penskalaan ini membandingkan nilai metrik agregat ini terhadap batas atas dan bawah yang ditentukan oleh penyesuaian langkah untuk menentukan penyesuaian langkah mana yang harus dilakukan.

Anda menentukan batas atas dan bawah terkait dengan ambang batas pelanggaran. Sebagai contoh, katakanlah Anda membuat CloudWatch alarm dan kebijakan scale-out ketika metrik di atas 50 persen. Anda kemudian membuat alarm kedua dan kebijakan skala dalam ketika metrik di bawah 50 persen. Anda membuat serangkaian penyesuaian langkah dengan jenis penyesuaian `PercentChangeInCapacity` (atau `Persentase grup di konsol`) untuk setiap kebijakan:

Contoh: Penyesuaian langkah untuk kebijakan menskalakan keluar

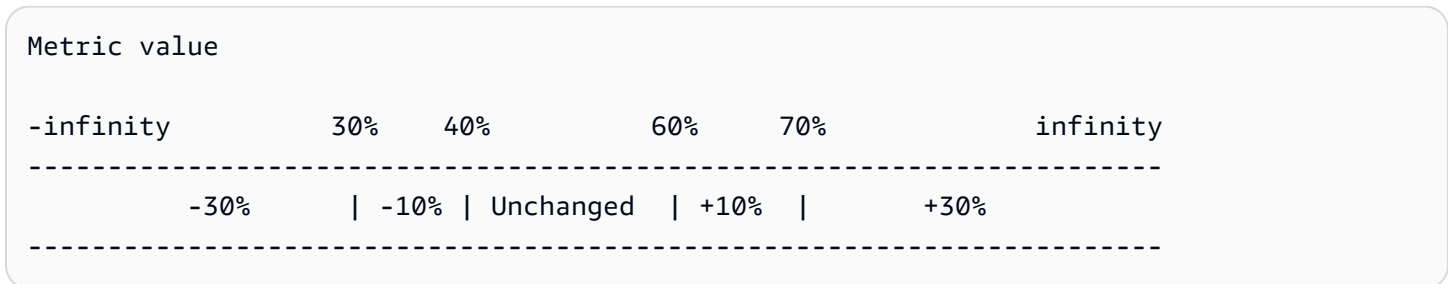
Batas bawah	Batas atas	Penyesuaian
0	10	0
10	20	10
20	nol	30

Contoh: Penyesuaian langkah untuk kebijakan penskalaan kedalam

Batas bawah	Batas atas	Penyesuaian
-10	0	0

Batas bawah	Batas atas	Penyesuaian
-20	-10	-10
nol	-20	-30

Ini membuat konfigurasi penskalaan berikut.



Sekarang, katakanlah Anda menggunakan konfigurasi penskalaan ini pada grup Auto Scaling yang memiliki kapasitas saat ini dan kapasitas yang diinginkan 10. Titik-titik berikut merangkum perilaku konfigurasi penskalaan dalam kaitannya dengan kapasitas grup yang diinginkan dan saat ini:

- Kapasitas yang diinginkan dan kapasitas saat ini dipertahankan sementara nilai metrik agregat lebih besar dari 40 dan kurang dari 60.
- Jika nilai metrik mencapai 60, kapasitas yang diinginkan grup meningkat sebesar 1 instance, menjadi 11 instance, berdasarkan penyesuaian langkah kedua kebijakan peningkatan skala (tambah 10 persen dari 10 instance). Setelah instance baru berjalan dan waktu pemanasan yang ditentukan telah kedaluwarsa, kapasitas grup saat ini meningkat menjadi 11 instance. Jika nilai metrik meningkat menjadi 70 bahkan setelah peningkatan kapasitas ini, kapasitas yang diinginkan grup meningkat sebesar 3 instance, menjadi 14 instance. Hal ini didasarkan pada penyesuaian langkah ketiga dari kebijakan peningkatan skala (tambah 30 persen dari 11 instance, 3,3 instance, dibulatkan ke 3 instance).
- Jika nilai metrik mencapai 40, kapasitas yang diinginkan dari grup menurun sebesar 1 kali, menjadi 13 instance, berdasarkan penyesuaian langkah kedua kebijakan penurunan skala (menghapus 10 persen dari 14 instance, 1,4 instance, dibulatkan ke 1 instance). Jika nilai metrik turun menjadi 30 bahkan setelah penurunan kapasitas ini, kapasitas yang diinginkan dari grup turun sebesar 3 instance lagi, menjadi 10 instance. Ini didasarkan pada penyesuaian langkah ketiga kebijakan scale-in (hapus 30 persen dari 13 instance, 3.9 instance, dibulatkan ke bawah menjadi 3 instance).

Saat Anda menentukan penyesuaian langkah untuk kebijakan penskalaan Anda, perhatikan hal berikut:

- Jika Anda menggunakan AWS Management Console, Anda menentukan batas atas dan bawah sebagai nilai absolut. Jika Anda menggunakan AWS CLI atau SDK, Anda menentukan batas atas dan bawah relatif terhadap ambang batas pelanggaran.
- Rentang penyesuaian langkah Anda tidak dapat tumpang tindih atau memiliki kesenjangan.
- Hanya satu penyesuaian langkah yang dapat memiliki batas bawah nol (negatif tak terhingga). Jika satu penyesuaian langkah memiliki batas bawah negatif, maka harus ada penyesuaian langkah dengan batas bawah nol.
- Hanya satu penyesuaian langkah yang dapat memiliki batas atas nol (positif tak terhingga). Jika satu penyesuaian langkah memiliki batas atas positif, maka harus ada penyesuaian langkah dengan batas atas nol.
- Batas atas dan bawah tidak boleh nol dalam penyesuaian langkah yang sama.
- Apabila nilai metrik berada di atas ambang batas penembusan, batas bawah bersifat inklusif dan batas atas bersifat eksklusif. Apabila nilai metrik berada di bawah ambang batas penembusan, batas bawah bersifat eksklusif dan batas atas bersifat inklusif.

Jenis penyesuaian penskalaan

Anda dapat menentukan kebijakan penskalaan yang melakukan tindakan penskalaan optimal, berdasarkan jenis penyesuaian penskalaan yang Anda pilih. Anda dapat menentukan jenis penyesuaian sebagai persentase kapasitas saat ini dari grup Auto Scaling, atau dalam unit kapasitas. Biasanya unit kapasitas berarti satu contoh, kecuali jika Anda menggunakan fitur bobot instance.

Amazon EC2 Auto Scaling mendukung jenis penyesuaian berikut untuk penskalaan langkah dan penskalaan sederhana:

- `ChangeInCapacity` — Meningkatkan atau mengurangi kapasitas grup saat ini menurut nilai yang telah ditentukan. Nilai positif meningkatkan kapasitas dan nilai penyesuaian negatif menurunkan kapasitas. Misalnya: Jika kapasitas grup saat ini adalah 3 dan penyesuaiannya adalah 5, maka ketika kebijakan ini dilakukan, kami menambahkan 5 unit kapasitas ke kapasitas untuk total 8 unit kapasitas.
- `ExactCapacity` — Mengubah kapasitas grup saat ini ke nilai yang telah ditentukan. Tentukan nilai non-negatif dengan jenis penyesuaian ini. Misalnya: Jika kapasitas grup saat ini adalah 3 dan

penyesuaiannya adalah 5, maka ketika kebijakan ini dilakukan, kami mengubah kapasitas menjadi 5 unit kapasitas.

- `PercentChangeInCapacity` — Meningkatkan atau mengurangi kapasitas grup saat ini berdasarkan persentase yang telah ditentukan. Nilai positif meningkatkan kapasitas dan nilai negatif menurunkan kapasitas. Misalnya: Jika kapasitas saat ini adalah 10 dan penyesuaian 10 persen, maka ketika kebijakan ini dilakukan, kami menambahkan 1 unit kapasitas ke kapasitas untuk total 11 unit kapasitas.

Note

Jika nilai yang dihasilkan bukan bilangan bulat, maka akan dibulatkan sebagai berikut:

- Nilai yang lebih besar dari 1 dibulatkan ke bawah. Misalnya, 12.7 dibulatkan ke 12.
- Nilai antara 0 dan 1 dibulatkan ke 1. Misalnya, .67 dibulatkan ke 1.
- Nilai antara 0 dan -1 dibulatkan ke -1. Misalnya, -.58 dibulatkan ke -1.
- Nilai kurang dari -1 dibulatkan ke atas. Misalnya, -6.67 dibulatkan ke -6.

Dengan `PercentChangeInCapacity`, Anda juga dapat menentukan jumlah minimum instance untuk diskalakan menggunakan parameter `MinAdjustmentMagnitude`. Misalnya, diasumsikan Anda membuat kebijakan yang menambahkan 25 persen dan Anda menentukan kenaikan minimum 2 instance. Jika Anda memiliki grup Auto Scaling dengan 4 instance dan kebijakan penskalaan dijalankan, 25 persen dari 4 adalah 1 instance. Namun, karena Anda menentukan peningkatan minimum sebesar 2, ada 2 instance yang ditambahkan.

Saat Anda menggunakan [bobot instance](#), efek pengaturan `MinAdjustmentMagnitude` parameter ke nilai bukan nol berubah. Nilainya adalah dalam unit kapasitas. Untuk menetapkan jumlah minimum instance ke skala, atur parameter ini ke nilai yang setidaknya sebesar bobot instance terbesar Anda.

Jika Anda menggunakan bobot instans, ingatlah bahwa kapasitas grup Auto Scaling Anda saat ini dapat melebihi kapasitas yang diinginkan sesuai kebutuhan. Jika jumlah absolut Anda yang akan berkurang, atau jumlah yang menurut persentase tersebut akan berkurang, adalah kurang dari perbedaan antara kapasitas saat ini dan yang diinginkan, tidak ada tindakan penskalaan yang diambil. Anda harus mempertimbangkan perilaku ini ketika Anda melihat hasil dari kebijakan penskalaan ketika alarm ambang batas dilanggar. Misalnya, kita asumsikan bahwa kapasitas yang diinginkan adalah 30 dan kapasitas saat ini adalah 32. Ketika alarm melanggar, jika kebijakan penskalaan mengurangi kapasitas yang diinginkan sebesar 1, maka tidak ada tindakan penskalaan yang diambil.

Pemanasan contoh

Untuk penskalaan langkah, Anda dapat menentukan jumlah detik yang diperlukan untuk pemanasan instans yang baru diluncurkan. Sampai waktu pemanasan yang ditentukan telah kedaluwarsa, instance tidak dihitung terhadap metrik instans EC2 agregat dari grup Auto Scaling.

Saat instans berada dalam periode pemanasan, kebijakan penskalaan Anda hanya dapat ditingkatkan jika nilai metrik dari instans yang tidak memanans lebih besar daripada ambang batas tinggi alarm kebijakan.

Jika kelompok menskalakan lagi, contoh yang masih memanans dihitung sebagai bagian dari kapasitas yang diinginkan untuk aktivitas penskalaan berikutnya. Oleh karena itu, beberapa penembusan alarm yang masuk dalam rentang penyesuaian langkah yang sama menyebabkan aktivitas penskalaan tunggal. Tujuannya adalah untuk terus (tetapi tidak berlebihan) menskalakan ke luar.

Misalnya, katakanlah Anda membuat kebijakan dengan dua langkah. Langkah pertama menambahkan 10 persen ketika metrik mencapai 60, dan langkah kedua menambahkan 30 persen ketika metrik mencapai 70 persen. Grup Auto Scaling Anda memiliki kapasitas 10 yang diinginkan dan saat ini. Kapasitas yang diinginkan dan saat ini tidak berubah sedangkan nilai metrik agregat kurang dari 60. Misalkan metrik mencapai 60, jadi 1 instance ditambahkan (10 persen dari 10 instance). Kemudian, metrik mencapai 62 saat instance baru masih memanans. Kebijakan penskalaan menghitung kapasitas baru yang diinginkan berdasarkan kapasitas saat ini, yang masih 10. Namun, kapasitas grup yang diinginkan telah meningkat menjadi 11 instance, sehingga kebijakan penskalaan tidak meningkatkan kapasitas yang diinginkan lebih lanjut. Jika metrik mencapai 70 pada saat instance baru masih pada tahap persiapan, kita harus menambahkan 3 instance (30 persen dari 10 instance). Namun, kapasitas yang diinginkan dari grup tersebut sudah 11, jadi kami hanya menambahkan 2 instance, untuk kapasitas baru yang diinginkan sebesar 13 instance.

Saat aktivitas penskalaan sedang berlangsung, semua aktivitas penskalaan yang dimulai oleh kebijakan penskalaan diblokir hingga instans selesai memanans. Ketika contoh selesai pemanasan, jika peristiwa skala dalam terjadi, setiap contoh yang saat ini dalam proses penghentian akan dihitung terhadap kapasitas grup saat ini ketika menghitung kapasitas baru yang diinginkan. Oleh karena itu, kami tidak menghapus lebih banyak instance daripada yang diperlukan oleh grup Auto Scaling. Misalnya, ketika sebuah instance sudah berakhir, jika alarm dilanggar dalam kisaran penyesuaian langkah yang sama yang mengurangi kapasitas yang diinginkan sebesar 1, maka tidak ada tindakan penskalaan yang diambil.

Nilai default

Jika tidak ada nilai yang disetel, maka kebijakan penskalaan akan menggunakan nilai default, yang merupakan nilai untuk [pemanasan instance default](#) yang ditentukan untuk grup. [Jika pemanasan instance default adalah null, maka itu kembali ke nilai cooldown default.](#)

Pertimbangan

Pertimbangan berikut berlaku saat bekerja dengan langkah dan kebijakan penskalaan sederhana:

- Pertimbangkan apakah Anda dapat memprediksi penyesuaian langkah pada aplikasi dengan cukup akurat untuk menggunakan penskalaan langkah. Jika metrik penskalaan Anda meningkat atau menurun secara proporsional dengan kapasitas target yang dapat diskalakan, sebaiknya gunakan kebijakan penskalaan pelacakan target sebagai gantinya. Anda masih memiliki opsi untuk menggunakan penskalaan langkah sebagai kebijakan tambahan untuk konfigurasi yang lebih canggih. Misalnya, Anda dapat mengonfigurasi respons yang lebih agresif saat pemanfaatan mencapai tingkat tertentu.
- Pastikan untuk memilih margin yang memadai antara ambang batas scale-out dan scale-in untuk mencegah flapping. Flapping adalah lingkaran penskalaan dan penskalaan yang tak terbatas. Artinya, jika tindakan penskalaan diambil, nilai metrik akan berubah dan memulai tindakan penskalaan lain dalam arah sebaliknya.

Buat kebijakan penskalaan langkah untuk skala keluar


Untuk membuat kebijakan penskalaan langkah untuk skala keluar untuk grup Auto Scaling Anda, gunakan salah satu metode berikut:

Console

Langkah 1: Buat CloudWatch alarm untuk ambang batas tinggi metrik

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Jika perlu, ubah Wilayah. Dari bilah navigasi, pilih Wilayah tempat grup Auto Scaling Anda berada.
3. Di panel navigasi, pilih Alarm, Semua alarm, lalu pilih Buat alarm.
4. Pilih Pilih metrik.
5. Pada tab Semua metrik, pilih EC2, Dengan Grup Auto Scaling, dan masukkan nama grup Auto Scaling di bidang pencarian. Lalu, pilih CPUUtilization dan pilih Pilih metrik. Halaman Tentukan metrik dan kondisi akan muncul, memperlihatkan grafik dan informasi lain tentang metrik.


6. Untuk Periode, pilih periode evaluasi untuk alarm, misalnya, 1 menit. Saat Anda mengevaluasi alarm, tiap periode akan digabungkan menjadi satu titik data.

 Note

Periode yang lebih pendek menciptakan alarm yang lebih sensitif.

7. Pada Ketentuan, lakukan hal berikut:

- Untuk Jenis ambang batas, pilih Statis.
- Untuk Kapan pun **CPUUtilization** ada, tentukan apakah Anda ingin nilai metrik lebih besar dari atau lebih besar dari atau sama dengan ambang batas untuk melanggar alarm. Kemudian, di bawah dari, masukkan nilai ambang batas yang ingin Anda tembus alarm.

 Important

Agar alarm dapat digunakan dengan kebijakan penskalaan (metrik tinggi), pastikan Anda tidak memilih kurang dari atau kurang dari atau sama dengan ambang batas.

8. Di bagian Konfigurasi tambahan, lakukan tindakan berikut:

- Agar titik data alarm, masukkan jumlah titik data (periode evaluasi) di mana nilai metrik harus memenuhi kondisi ambang batas untuk alarm. Misalnya, dua periode berturut-turut selama 5 menit akan memakan waktu 10 menit untuk memanggil status alarm.
- Untuk Perlakuan data yang hilang, pilih Perlakukan data yang hilang sebagai buruk (menembus ambang batas). Untuk informasi selengkapnya, lihat [Mengonfigurasi cara CloudWatch alarm menangani data yang hilang](#) di CloudWatch Panduan Pengguna Amazon.

9. Pilih Berikutnya.

Halaman Konfigurasi tindakan muncul.

10. Di bawah Pemberitahuan, pilih topik Amazon SNS untuk memberi tahu saat alarm dalam ALARM status, status, atau OK status. `INSUFFICIENT_DATA`

Agar alarm mengirimkan beberapa notifikasi untuk status alarm yang sama atau untuk status alarm yang berbeda, silakan pilih Tambahkan notifikasi.

Agar alarm tidak mengirim notifikasi, silakan pilih Hapus.

11. Anda dapat membiarkan bagian lain dari halaman Konfigurasi tindakan kosong. Membiarkan bagian lain kosong akan membuat alarm tanpa mengaitkannya dengan kebijakan penskalaan. Anda kemudian dapat mengaitkan alarm dengan kebijakan penskalaan dari konsol Auto Scaling Amazon EC2.
12. Pilih Berikutnya.
13. Masukkan nama (misalnya, `Step-Scaling-AlarmHigh-AddCapacity`) dan, secara opsional, deskripsi untuk alarm, lalu pilih Selanjutnya.
14. Pilih Buat alarm.

Gunakan prosedur berikut untuk melanjutkan di mana Anda tinggalkan setelah membuat CloudWatch alarm Anda.

Langkah 2: Buat kebijakan penskalaan langkah untuk skala keluar

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

3. Verifikasi bahwa batas penskalaan ditetapkan dengan tepat. Misalnya, jika kapasitas yang diinginkan grup Anda sudah maksimal, Anda perlu menentukan maksimum baru untuk meningkatkan skala. Untuk informasi selengkapnya, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).
4. Pada tab Penskalaan otomatis, dalam Kebijakan penskalaan dinamis, pilih Buat kebijakan penskalaan dinamis.
5. Untuk jenis Kebijakan, pilih Penskalaan langkah, lalu tentukan nama untuk kebijakan tersebut.
6. Untuk alarm CloudWatch, pilih alarm Anda. Jika Anda belum membuat alarm, pilih Buat CloudWatch alarm dan selesaikan langkah 4 hingga langkah 14 di prosedur sebelumnya untuk membuat alarm.
7. Tentukan perubahan dalam ukuran grup saat ini yang akan dibuat kebijakan ini ketika dijalankan menggunakan Ambil tindakan. Anda dapat menambahkan jumlah instance atau persentase tertentu dari ukuran grup yang ada, atau mengatur grup ke ukuran yang tepat.

Misalnya, untuk membuat kebijakan penskalaan yang meningkatkan kapasitas grup sebesar 30 persen, pilih `Add`, masukkan `30` di bidang berikutnya, lalu pilih `percent of group`

Secara default, batas bawah untuk penyesuaian langkah ini adalah ambang alarm dan batas atas adalah tak terhingga positif (+).

8. Untuk menambahkan langkah lain, pilih Tambahkan langkah kemudian tentukan jumlah yang akan disesuaikan skalanya dan batas bawah dan atas langkah dibandingkan terhadap ambang batas alarm.
9. Untuk menetapkan jumlah minimum instance untuk diskalakan, perbarui kolom angka dalam Tambahkan unit kapasitas dalam satuan setidaknya 1 unit kapasitas.
10. (Opsional) Untuk pemanasan Instance, perbarui nilai pemanasan instance sesuai kebutuhan.
11. Pilih Buat.

AWS CLI

Untuk membuat kebijakan penskalaan langkah untuk skala keluar (meningkatkan kapasitas), Anda dapat menggunakan perintah contoh berikut. Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri.

Saat menggunakan AWS CLI, pertama-tama Anda membuat kebijakan penskalaan langkah yang memberikan petunjuk ke Auto Scaling Amazon EC2 tentang cara menskalakan saat nilai metrik meningkat. Kemudian, Anda membuat alarm dengan mengidentifikasi metrik yang akan ditonton, menentukan ambang batas tinggi metrik dan detail lain untuk alarm, dan mengaitkan alarm dengan kebijakan penskalaan.

Langkah 1: Buat kebijakan untuk skala keluar

Gunakan [put-scaling-policy](#) perintah berikut untuk membuat kebijakan penskalaan langkah bernama `my-step-scale-out-policy`, dengan jenis penyesuaian `PercentChangeInCapacity` yang meningkatkan kapasitas grup berdasarkan penyesuaian langkah berikut (dengan asumsi ambang CloudWatch alarm 60 persen):

- Tingkatkan jumlah instans sebesar 10 persen ketika nilai metrik lebih besar dari atau sama dengan 60 persen tetapi kurang dari 75 persen
- Tingkatkan jumlah instans sebesar 20 persen ketika nilai metrik lebih besar dari atau sama dengan 75 persen tetapi kurang dari 85 persen
- Tingkatkan jumlah instance sebesar 30 persen ketika nilai metrik lebih besar dari atau sama dengan 85 persen

```
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name my-asg \
  --policy-name my-step-scale-out-policy \
  --policy-type StepScaling \
  --adjustment-type PercentChangeInCapacity \
  --metric-aggregation-type Average \
  --step-adjustments
MetricIntervalLowerBound=0.0,MetricIntervalUpperBound=15.0,ScalingAdjustment=10 \

MetricIntervalLowerBound=15.0,MetricIntervalUpperBound=25.0,ScalingAdjustment=20 \
  MetricIntervalLowerBound=25.0,ScalingAdjustment=30 \
  --min-adjustment-magnitude 1
```

Merekam Amazon Resource Name (ARN) kebijakan. Anda membutuhkannya untuk membuat CloudWatch alarm untuk kebijakan tersebut.

```
{
  "PolicyARN":
  "arn:aws:autoscaling:region:123456789012:scalingPolicy:4ee9e543-86b5-4121-b53b-aa4c23b5bbcc:autoScalingGroupName/my-asg:policyName/my-step-scale-in-policy
}
```

Langkah 2: Buat CloudWatch alarm untuk ambang batas tinggi metrik

Gunakan CloudWatch [put-metric-alarm](#) perintah berikut untuk membuat alarm yang meningkatkan ukuran grup Auto Scaling berdasarkan nilai ambang CPU rata-rata 60 persen untuk setidaknya dua periode evaluasi berturut-turut selama dua menit. Untuk menggunakan metrik kustom Anda sendiri, sebutkan namanya di `--metric-name` dan ruang namanya di `--namespace`.

```
aws cloudwatch put-metric-alarm --alarm-name Step-Scaling-AlarmHigh-AddCapacity \
  --metric-name CPUUtilization --namespace AWS/EC2 --statistic Average \
  --period 120 --evaluation-periods 2 --threshold 60 \
  --comparison-operator GreaterThanOrEqualToThreshold \
  --dimensions "Name=AutoScalingGroupName,Value=my-asg" \
  --alarm-actions PolicyARN
```

Buat kebijakan penskalaan langkah untuk skala di

Untuk membuat kebijakan penskalaan langkah untuk penskalaan grup Auto Scaling Anda, gunakan salah satu metode berikut:

Console

Langkah 1: Buat CloudWatch alarm untuk ambang batas rendah metrik

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Jika perlu, ubah Wilayah. Dari bilah navigasi, pilih Wilayah tempat grup Auto Scaling Anda berada.
3. Di panel navigasi, pilih Alarm, Semua alarm, lalu pilih Buat alarm.
4. Pilih Pilih metrik.
5. Pada tab Semua metrik, pilih EC2, Dengan Grup Auto Scaling, dan masukkan nama grup Auto Scaling di bidang pencarian. Lalu, pilih CPUUtilization dan pilih Pilih metrik. Halaman Tentukan metrik dan kondisi akan muncul, memperlihatkan grafik dan informasi lain tentang metrik.
6. Untuk Periode, pilih periode evaluasi untuk alarm, misalnya, 1 menit. Saat Anda mengevaluasi alarm, tiap periode akan digabungkan menjadi satu titik data.

Note

Periode yang lebih pendek menciptakan alarm yang lebih sensitif.

7. Pada Ketentuan, lakukan hal berikut:
 - Untuk Jenis ambang batas, pilih Statis.
 - Untuk Kapan pun **CPUUtilization** ada, tentukan apakah Anda ingin nilai metrik kurang dari atau kurang dari atau sama dengan ambang batas untuk melanggar alarm. Kemudian, di bawah dari, masukkan nilai ambang batas yang ingin Anda tembus alarm.

Important

Agar alarm dapat digunakan dengan kebijakan scale-in (metrik rendah), pastikan Anda tidak memilih lebih besar dari atau lebih besar dari atau sama dengan ambang batas.

8. Di bagian Konfigurasi tambahan, lakukan tindakan berikut:
 - Agar titik data alarm, masukkan jumlah titik data (periode evaluasi) di mana nilai metrik harus memenuhi kondisi ambang batas untuk alarm. Misalnya, dua periode berturut-turut selama 5 menit akan memakan waktu 10 menit untuk memanggil status alarm.

- Untuk Perlakuan data yang hilang, pilih Perlakukan data yang hilang sebagai buruk (menembus ambang batas). Untuk informasi selengkapnya, lihat [Mengonfigurasi cara CloudWatch alarm menangani data yang hilang](#) di CloudWatch Panduan Pengguna Amazon.

9. Pilih Berikutnya.

Halaman Konfigurasi tindakan muncul.

10. Di bawah Pemberitahuan, pilih topik Amazon SNS untuk memberi tahu saat alarm dalam ALARM status, status, atau OK status. INSUFFICIENT_DATA

Agar alarm mengirimkan beberapa notifikasi untuk status alarm yang sama atau untuk status alarm yang berbeda, silakan pilih Tambahkan notifikasi.

Agar alarm tidak mengirim notifikasi, silakan pilih Hapus.

11. Anda dapat membiarkan bagian lain dari halaman Konfigurasi tindakan kosong. Membiarkan bagian lain kosong akan membuat alarm tanpa mengaitkannya dengan kebijakan penskalaan. Anda kemudian dapat mengaitkan alarm dengan kebijakan penskalaan dari konsol Auto Scaling Amazon EC2.

12. Pilih Berikutnya.

13. Masukkan nama (misalnya, Step-Scaling-AlarmLow-RemoveCapacity) dan, secara opsional, deskripsi untuk alarm, lalu pilih Selanjutnya.

14. Pilih Buat alarm.

Gunakan prosedur berikut untuk melanjutkan di mana Anda tinggalkan setelah membuat CloudWatch alarm Anda.

Langkah 2: Buat kebijakan penskalaan langkah untuk skala

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

3. Verifikasi bahwa batas penskalaan ditetapkan dengan tepat. Misalnya, jika kapasitas yang diinginkan grup Anda sudah minimum, Anda perlu menentukan minimum baru untuk

menskalakan. Untuk informasi selengkapnya, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).

4. Pada tab Penskalaan otomatis, dalam Kebijakan penskalaan dinamis, pilih Buat kebijakan penskalaan dinamis.
5. Untuk jenis Kebijakan, pilih Penskalaan langkah, lalu tentukan nama untuk kebijakan tersebut.
6. Untuk alarm CloudWatch, pilih alarm Anda. Jika Anda belum membuat alarm, pilih Buat CloudWatch alarm dan selesaikan langkah 4 hingga langkah 14 di prosedur sebelumnya untuk membuat alarm.
7. Tentukan perubahan dalam ukuran grup saat ini yang akan dibuat kebijakan ini ketika dijalankan menggunakan Ambil tindakan. Anda dapat menghapus sejumlah instance tertentu atau persentase ukuran grup yang ada, atau mengatur grup ke ukuran yang tepat.

Misalnya, untuk membuat kebijakan penskalaan yang mengurangi kapasitas grup sebanyak dua instance, pilih Remove, masukkan 2 di bidang berikutnya, lalu pilih. `capacity units` Secara default, batas atas untuk penyesuaian langkah ini adalah ambang alarm dan batas bawah negatif (-) tak terhingga.

8. Untuk menambahkan langkah lain, pilih Tambahkan langkah kemudian tentukan jumlah yang akan disesuaikan skalanya dan batas bawah dan atas langkah dibandingkan terhadap ambang batas alarm.
9. Pilih Buat.

AWS CLI

Untuk membuat kebijakan penskalaan langkah untuk skala (kurangi kapasitas), Anda dapat menggunakan perintah contoh berikut. Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri.

Saat menggunakan AWS CLI, pertama-tama Anda membuat kebijakan penskalaan langkah yang memberikan petunjuk ke Auto Scaling Amazon EC2 tentang cara menskalakan saat nilai metrik menurun. Kemudian, Anda membuat alarm dengan mengidentifikasi metrik yang akan ditonton, menentukan ambang batas rendah metrik dan detail lain untuk alarm, dan mengaitkan alarm dengan kebijakan penskalaan.

Langkah 1: Buat kebijakan untuk skala

Gunakan [put-scaling-policy](#) perintah berikut untuk membuat kebijakan penskalaan langkah bernamamy-step-scale-in-policy, dengan jenis penyesuaian ChangeInCapacity yang mengurangi kapasitas grup sebanyak 2 instance saat CloudWatch alarm terkait melanggar nilai ambang batas rendah metrik.

```
aws autoscaling put-scaling-policy \
  --auto-scaling-group-name my-asg \
  --policy-name my-step-scale-in-policy \
  --policy-type StepScaling \
  --adjustment-type ChangeInCapacity \
  --step-adjustments MetricIntervalUpperBound=0.0,ScalingAdjustment=-2
```

Merekam Amazon Resource Name (ARN) kebijakan. Anda membutuhkannya untuk membuat CloudWatch alarm untuk kebijakan tersebut.

```
{
  "PolicyARN": "arn:aws:autoscaling:region:123456789012:scalingPolicy:ac542982-cbeb-4294-891c-a5a941dfa787:autoScalingGroupName/my-asg:policyName/my-step-scale-out-policy"
}
```

Langkah 2: Buat CloudWatch alarm untuk ambang batas rendah metrik

Gunakan CloudWatch [put-metric-alarm](#) perintah berikut untuk membuat alarm yang mengurangi ukuran grup Auto Scaling berdasarkan nilai ambang CPU rata-rata 40 persen untuk setidaknya dua periode evaluasi berturut-turut selama dua menit. Untuk menggunakan metrik kustom Anda sendiri, sebutkan namanya di --metric-name dan ruang namanya di --namespace.

```
aws cloudwatch put-metric-alarm --alarm-name Step-Scaling-AlarmLow-RemoveCapacity \
  --metric-name CPUUtilization --namespace AWS/EC2 --statistic Average \
  --period 120 --evaluation-periods 2 --threshold 40 \
  --comparison-operator LessThanOrEqualToThreshold \
  --dimensions "Name=AutoScalingGroupName,Value=my-asg" \
  --alarm-actions PolicyARN
```

Kebijakan penskalaan sederhana

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan perintah CLI untuk membuat kebijakan penskalaan sederhana. Mereka tetap dalam dokumen ini sebagai referensi untuk setiap

pelanggan yang ingin menggunakannya, tetapi kami menyarankan Anda menggunakan pelacakan target atau kebijakan penskalaan langkah sebagai gantinya.

Mirip dengan kebijakan penskalaan langkah, kebijakan penskalaan sederhana mengharuskan Anda membuat CloudWatch alarm untuk kebijakan penskalaan Anda. Dalam kebijakan yang Anda buat, Anda juga harus menentukan apakah akan menambah atau menghapus instance, dan berapa banyak, atau menyetel grup ke ukuran yang tepat.

Salah satu perbedaan utama antara kebijakan penskalaan langkah dan kebijakan penskalaan sederhana adalah penyesuaian langkah yang Anda dapatkan dengan kebijakan penskalaan langkah. Dengan penskalaan langkah, Anda dapat membuat perubahan yang lebih besar atau lebih kecil pada ukuran grup berdasarkan penyesuaian langkah yang Anda tentukan.

Kebijakan penskalaan sederhana juga harus menunggu aktivitas penskalaan yang sedang berlangsung atau penggantian pemeriksaan kesehatan selesai dan [periode cooldown](#) berakhir sebelum merespons alarm tambahan. Sebaliknya, dengan penskalaan langkah, kebijakan terus merespons alarm tambahan, bahkan saat aktivitas penskalaan atau penggantian pemeriksaan kesehatan sedang berlangsung. Ini berarti bahwa Amazon EC2 Auto Scaling mengevaluasi semua pelanggaran alarm saat menerima pesan alarm. Karena itu, kami menyarankan Anda menggunakan kebijakan penskalaan langkah sebagai gantinya, meskipun Anda hanya memiliki satu penyesuaian penskalaan.

Amazon EC2 Auto Scaling awalnya hanya mendukung kebijakan penskalaan sederhana. Jika Anda membuat kebijakan penskalaan sebelum kebijakan pelacakan target dan penskalaan langkah diperkenalkan, kebijakan Anda diperlakukan sebagai kebijakan penskalaan sederhana.

Buat kebijakan penskalaan sederhana untuk skala keluar

Gunakan [put-scaling-policy](#) perintah berikut untuk membuat kebijakan penskalaan sederhana bernama `my-simple-scale-out-policy`, dengan jenis penyesuaian `PercentChangeInCapacity` yang meningkatkan kapasitas grup sebesar 30 persen saat CloudWatch alarm terkait melanggar nilai ambang batas tinggi metrik.

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-out-policy \  
  --auto-scaling-group-name my-asg --scaling-adjustment 30 \  
  --adjustment-type PercentChangeInCapacity
```

Merekam Amazon Resource Name (ARN) kebijakan. Anda membutuhkannya untuk membuat CloudWatch alarm untuk kebijakan tersebut.

Buat kebijakan penskalaan sederhana untuk skala

Gunakan [put-scaling-policy](#) perintah berikut untuk membuat kebijakan penskalaan sederhana bernama `my-simple-scale-in-policy`, dengan jenis penyesuaian `ChangeInCapacity` yang mengurangi kapasitas grup dengan satu instance saat CloudWatch alarm terkait melanggar nilai ambang batas rendah metrik.

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-in-policy \  
  --auto-scaling-group-name my-asg --scaling-adjustment -1 \  
  --adjustment-type ChangeInCapacity --cooldown 180
```

Merekam Amazon Resource Name (ARN) kebijakan. Anda membutuhkannya untuk membuat CloudWatch alarm untuk kebijakan tersebut.

Menskalakan jeda pakai untuk Amazon EC2 Auto Scaling

Important

Sebagai praktik terbaik, kami menyarankan Anda untuk tidak menggunakan kebijakan penskalaan sederhana dan cooldown penskalaan. Kebijakan penskalaan pelacakan target atau kebijakan penskalaan langkah lebih baik untuk penskalaan kinerja. Untuk kebijakan penskalaan yang mengubah ukuran grup Auto Scaling Anda secara proporsional saat nilai metrik penskalaan menurun atau meningkat, kami menyarankan [pelacakan target](#) menggunakan penskalaan sederhana atau penskalaan langkah.

Saat Anda membuat kebijakan penskalaan sederhana untuk grup Auto Scaling, sebaiknya Anda mengonfigurasi cooldown penskalaan secara bersamaan.

Setelah grup Auto Scaling Anda meluncurkan atau menghentikan instans, grup ini menunggu periode cooldown berakhir sebelum aktivitas penskalaan lebih lanjut yang dimulai oleh kebijakan penskalaan sederhana dapat dimulai. Tujuan dari periode cooldown adalah agar grup Auto Scaling Anda stabil dan mencegahnya meluncurkan atau menghentikan instance tambahan sebelum efek aktivitas penskalaan sebelumnya terlihat.

Misalkan, misalnya, bahwa kebijakan penskalaan sederhana untuk pemanfaatan CPU merekomendasikan peluncuran dua instance. Auto Scaling Amazon EC2 meluncurkan dua instans dan kemudian menjeda aktivitas penskalaan hingga periode cooldown berakhir. Setelah periode

cooldown berakhir, aktivitas penskalaan apa pun yang diprakarsai oleh kebijakan penskalaan sederhana dapat dilanjutkan. Jika pemanfaatan CPU melanggar ambang batas tinggi alarm lagi, grup Auto Scaling keluar lagi, dan periode cooldown berlaku lagi. Namun, jika dua contoh cukup untuk menurunkan nilai metrik, grup tetap pada ukurannya saat ini.

Daftar Isi

- [Pertimbangan](#)
- [Kait siklus hidup dapat menyebabkan penundaan tambahan](#)
- [Ubah periode cooldown default](#)
- [Tetapkan periode cooldown untuk kebijakan penskalaan sederhana tertentu](#)

Pertimbangan

Pertimbangan berikut berlaku saat bekerja dengan kebijakan penskalaan sederhana dan cooldown penskalaan:

- Kebijakan pelacakan target dan penskalaan langkah dapat segera memulai aktivitas penskalaan tanpa menunggu periode cooldown berakhir. Sebagai gantinya, setiap kali grup Auto Scaling Anda meluncurkan instance, masing-masing instance memiliki periode pemanasan. Untuk informasi selengkapnya, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#).
- Ketika tindakan terjadwal dimulai pada waktu yang dijadwalkan, itu juga dapat memulai aktivitas penskalaan segera tanpa menunggu periode cooldown berakhir.
- Jika instans menjadi tidak sehat, Auto Scaling Amazon EC2 tidak menunggu periode cooldown berakhir sebelum mengganti instans yang tidak sehat.
- Saat beberapa instance diluncurkan atau dihentikan, periode cooldown (baik cooldown default atau cooldown khusus kebijakan penskalaan) berlaku mulai saat instance terakhir selesai diluncurkan atau dihentikan.
- Saat Anda menskalakan grup Auto Scaling secara manual, defaultnya adalah tidak menunggu cooldown berakhir. Namun, Anda dapat mengganti perilaku ini dan menghormati cooldown default saat Anda menggunakan AWS CLI atau SDK untuk menskalakan secara manual.
- Secara default, Elastic Load Balancing menunggu 300 detik untuk menyelesaikan proses deregistrasi (connection draining). Jika grup berada di belakang penyeimbang beban Elastic Load Balancing, ia akan menunggu instans penghentian untuk membatalkan pendaftaran sebelum memulai periode cooldown.

Kait siklus hidup dapat menyebabkan penundaan tambahan

Jika [pengait siklus hidup](#) dipanggil, periode cooldown dimulai setelah Anda menyelesaikan tindakan siklus hidup atau setelah periode batas waktu berakhir. Misalnya, pertimbangkan grup Auto Scaling yang memiliki kait siklus hidup misalnya peluncuran. Ketika aplikasi mengalami peningkatan permintaan, grup meluncurkan instance untuk menambah kapasitas. Karena ada kait siklus hidup, instance dimasukkan ke dalam status tunggu dan aktivitas penskalaan karena kebijakan penskalaan sederhana dijeda. Saat instance masuk ke status `InService`, periode jeda pakai dimulai. Ketika periode cooldown berakhir, kegiatan kebijakan penskalaan sederhana dilanjutkan.

Ketika Elastic Load Balancing diaktifkan, untuk tujuan penskalaan, periode cooldown dimulai ketika instance yang dipilih untuk penghentian memulai pengurusan koneksi (penundaan deregistrasi). Periode cooldown tidak menunggu koneksi terkuras selesai atau pengait siklus hidup untuk menyelesaikan aksinya. Ini berarti bahwa setiap aktivitas penskalaan karena kebijakan penskalaan sederhana dapat dilanjutkan segera setelah hasil dari peristiwa penskalaan tercermin dalam kapasitas grup. Jika tidak, menunggu untuk menyelesaikan ketiga aktivitas—pengurusan koneksi, pengait siklus hidup, dan periode cooldown—akan secara signifikan meningkatkan jumlah waktu yang dibutuhkan grup Auto Scaling untuk menjeda penskalaan.

Ubah periode cooldown default

Anda tidak dapat mengatur cooldown default saat pertama kali membuat grup Auto Scaling di konsol Auto Scaling Amazon EC2. Secara default, periode cooldown ini diatur ke 300 detik (5 menit). Jika perlu, Anda dapat memperbarui ini setelah grup dibuat.

Untuk mengubah periode cooldown default (konsol)

Setelah membuat grup Auto Scaling, pada tab Detail, pilih Konfigurasi lanjutan, Edit. Untuk cooldown Default, pilih jumlah waktu yang Anda inginkan berdasarkan waktu startup instans Anda atau kebutuhan aplikasi lainnya.

Untuk mengubah periode cooldown default (AWS CLI)

Gunakan perintah berikut untuk mengubah cooldown default untuk grup Auto Scaling baru atau yang sudah ada. Jika cooldown default tidak ditentukan, nilai default 300 detik digunakan.

- [create-auto-scaling-group](#)
- [update-auto-scaling-group](#)

Untuk mengkonfirmasi nilai cooldown default, gunakan [describe-auto-scaling-groups](#) perintah.

Tetapkan periode cooldown untuk kebijakan penskalaan sederhana tertentu

Secara default, semua kebijakan penskalaan sederhana menggunakan periode cooldown default yang ditentukan untuk grup Auto Scaling. Untuk menetapkan periode cooldown untuk kebijakan penskalaan sederhana tertentu, gunakan parameter cooldown opsional saat Anda membuat atau memperbarui kebijakan. Ketika periode cooldown ditentukan untuk kebijakan, itu akan mengganti cooldown default.

Salah satu penggunaan umum untuk periode cooldown khusus kebijakan penskalaan adalah dengan kebijakan scale-in. Karena kebijakan ini menghentikan instance, Amazon EC2 Auto Scaling membutuhkan waktu yang lebih sedikit untuk menentukan apakah akan menghentikan instance tambahan. Pengakhiran instance akan menjadi operasi yang jauh lebih cepat daripada meluncurkan instance. Oleh karena itu periode jeda pakai default selama 300 detik terlalu lama. Dalam hal ini, periode cooldown khusus kebijakan penskalaan dengan nilai yang lebih rendah untuk kebijakan scale-in Anda dapat membantu Anda mengurangi biaya dengan memungkinkan grup untuk menskalakan lebih cepat.

Untuk membuat atau memperbarui kebijakan penskalaan sederhana di konsol, pilih tab Penskalaan otomatis setelah Anda membuat grup. Untuk membuat atau memperbarui kebijakan penskalaan sederhana menggunakan AWS CLI, gunakan [put-scaling-policy](#) perintah. Untuk informasi selengkapnya, lihat [Kebijakan penskalaan langkah dan sederhana](#).

Penskalaan berdasarkan Amazon SQS

Important

Informasi dan langkah-langkah berikut menunjukkan cara menghitung backlog antrian Amazon SQS per instance menggunakan atribut antrian sebelum mempublikasikannya sebagai metrik kustom. `ApproximateNumberOfMessages` CloudWatch Namun, Anda sekarang dapat menghemat biaya dan upaya untuk menerbitkan metrik Anda sendiri dengan menggunakan matematika metrik. Untuk informasi selengkapnya, lihat [Membuat kebijakan penskalaan pelacakan target untuk Auto Scaling Amazon EC2 menggunakan matematika metrik](#).

Bagian ini menunjukkan cara menskalakan grup Auto Scaling sebagai respons atas perubahan dalam beban sistem di antrian Amazon Simple Queue Service (Amazon SQS). Untuk mempelajari

lebih lanjut tentang cara menggunakan Amazon SQS, lihat [Panduan Pengembang Amazon Simple Queue Service](#).

Ada beberapa skenario di mana Anda mungkin berpikir tentang penskalaan sebagai respons terhadap aktivitas dalam antrean Amazon SQS. Misalnya, kita asumsikan Anda memiliki aplikasi web yang memungkinkan pengguna mengunggah gambar dan menggunakannya secara online. Dalam skenario ini, setiap gambar perlu mengubah ukuran dan mengenkodkannya sebelum dapat diterbitkan. Aplikasi ini berjalan pada instance EC2 dalam grup Auto Scaling, dan dikonfigurasi untuk menangani tingkat pengunggahan standar Anda. Instance yang tidak sehat dihentikan dan diganti untuk memelihara tingkat instance saat ini setiap saat. Aplikasi ini menempatkan data gambar bitmap mentah dalam antrean SQS untuk pemrosesan. Aplikasi ini memproses gambar dan menerbitkan gambar yang diproses di mana gambar tersebut dapat dilihat oleh pengguna. Arsitektur untuk skenario ini berfungsi dengan baik jika jumlah pengunggahan gambar tidak bervariasi dari waktu ke waktu. Tetapi jika jumlah unggahan berubah dari waktu ke waktu, Anda dapat mempertimbangkan menggunakan penskalaan dinamis untuk menskalakan kapasitas grup Auto Scaling Anda.

Daftar Isi

- [Gunakan pelacakan target dengan metrik yang tepat](#)
- [Batasan dan prasyarat](#)
- [Mengonfigurasi penskalaan berdasarkan Amazon SQS](#)
- [Amazon SQS dan perlindungan penskalaan instans](#)

Gunakan pelacakan target dengan metrik yang tepat

Jika Anda menggunakan kebijakan penskalaan pelacakan target berdasarkan metrik antrean Amazon SQS kustom, penskalaan dinamis dapat menyesuaikan kurva permintaan aplikasi Anda secara lebih efektif. Untuk informasi selengkapnya tentang memilih metrik untuk pelacakan target, lihat [Pilih metrik](#).

Masalah dengan menggunakan metrik CloudWatch Amazon SQS seperti `ApproximateNumberOfMessagesVisible` untuk pelacakan target adalah bahwa jumlah pesan dalam antrian mungkin tidak berubah secara proporsional dengan ukuran grup Auto Scaling yang memproses pesan dari antrian. Hal itu karena jumlah pesan dalam antrean SQS Anda tidak hanya menentukan jumlah instance yang diperlukan. Jumlah instance dalam grup Auto Scaling Anda dapat didorong oleh beberapa faktor, termasuk berapa lama waktu yang diperlukan untuk memproses pesan dan jumlah latensi yang dapat diterima (penundaan antrean).

Solusinya adalah menggunakan metrik backlog per instance dengan nilai target menjadi backlog yang dapat diterima per instance untuk dipertahankan. Anda dapat menghitung angka-angka ini sebagai berikut:

- Backlog per instance: Untuk menghitung backlog per instance, mulailah dengan atribut `ApproximateNumberOfMessages` antrian untuk menentukan panjang antrian SQS (jumlah pesan yang tersedia untuk diambil dari antrian). Bagi jumlahnya berdasarkan kapasitas berjalan fleet, yang untuk grup Auto Scaling adalah jumlah instance dalam status `InService`, untuk mendapatkan backlog per instance.
- Backlog yang dapat diterima per instance: Untuk menghitung nilai target Anda, tentukan terlebih dahulu apa yang dapat diterima aplikasi Anda dalam hal latensi. Kemudian, ambil nilai latensi yang dapat diterima dan bagi dengan waktu rata-rata yang diperlukan oleh instance EC2 untuk memproses pesan.

Sebagai contoh, katakanlah Anda saat ini memiliki grup Auto Scaling dengan 10 instance dan jumlah pesan yang terlihat dalam antrian (`ApproximateNumberOfMessages`) adalah 1500. Jika waktu pemrosesan rata-rata adalah 0,1 detik untuk setiap pesan dan latensi terpanjang yang dapat diterima adalah 10 detik, maka backlog yang dapat diterima per instance adalah $10/0,1$, yang sama dengan 100 pesan. Ini berarti bahwa 100 adalah nilai target untuk kebijakan pelacakan target Anda. Ketika backlog per instance mencapai nilai target, peristiwa scale-out akan terjadi. Karena backlog per instance sudah 150 pesan (1500 pesan/10 instance), grup Anda menskalakan, dan skalanya menjadi lima instance untuk mempertahankan proporsi dengan nilai target.

Prosedur berikut mendemonstrasikan cara mempublikasikan metrik khusus dan membuat kebijakan penskalaan pelacakan target yang mengonfigurasi grup Auto Scaling Anda ke skala berdasarkan perhitungan ini.

Important

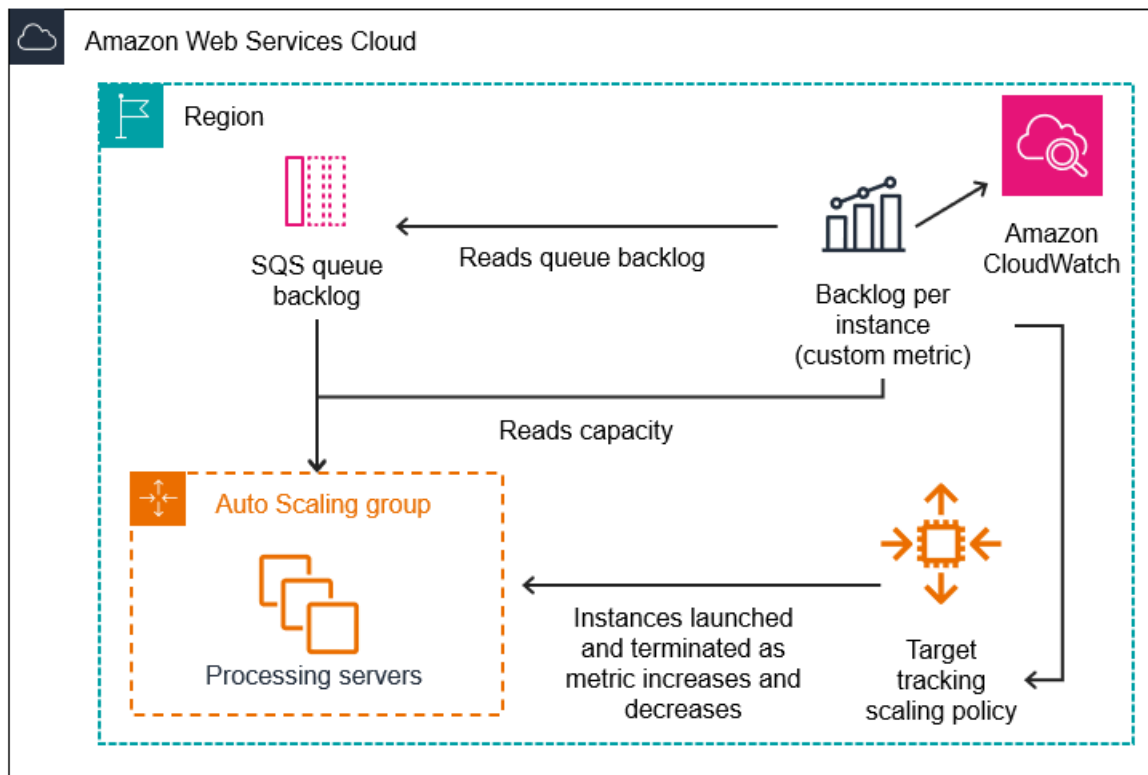
Ingat, untuk mengurangi biaya, gunakan matematika metrik sebagai gantinya. Untuk informasi selengkapnya, lihat [Membuat kebijakan penskalaan pelacakan target untuk Auto Scaling Amazon EC2 menggunakan matematika metrik](#).

Ada tiga bagian utama untuk konfigurasi ini:

- Grup Auto Scaling untuk mengelola instance EC2 untuk tujuan pemrosesan pesan dari antrian SQS.

- Metrik khusus untuk dikirim ke Amazon CloudWatch yang mengukur jumlah pesan dalam antrian per instans EC2 di grup Auto Scaling.
- Kebijakan pelacakan target yang mengonfigurasi grup Auto Scaling Anda untuk menskalakan berdasarkan metrik kustom dan nilai target yang ditetapkan. CloudWatch alarm memanggil kebijakan penskalaan.

Diagram berikut menggambarkan arsitektur konfigurasi ini.



Batasan dan prasyarat

Untuk menggunakan konfigurasi ini, Anda perlu mengetahui batasan berikut:

- Anda harus menggunakan AWS CLI atau SDK untuk mempublikasikan metrik kustom Anda. CloudWatch Anda kemudian dapat memantau metrik Anda dengan AWS Management Console.
- Konsol Auto Scaling Amazon EC2 tidak mendukung kebijakan penskalaan pelacakan target yang menggunakan metrik khusus. Anda harus menggunakan AWS CLI atau SDK untuk menentukan metrik kustom untuk kebijakan penskalaan Anda.

Bagian berikut mengarahkan Anda untuk menggunakan AWS CLI untuk tugas-tugas yang perlu Anda lakukan. Misalnya, untuk mendapatkan data metrik yang mencerminkan penggunaan antrian saat ini, Anda menggunakan perintah SQS. [get-queue-attributes](#) Pastikan bahwa Anda sudah [memasang](#) dan [mengonfigurasi](#) CLI.

Sebelum memulai, Anda harus memiliki antrean Amazon SQS untuk digunakan. Bagian berikut mengasumsikan bahwa Anda sudah memiliki antrean (standar atau FIFO), grup Auto Scaling, dan instance EC2 yang menjalankan aplikasi yang menggunakan antrean. Untuk informasi lebih lanjut tentang Amazon SQS, lihat [Panduan Pengembang Amazon Simple Queue Service](#).

Mengonfigurasi penskalaan berdasarkan Amazon SQS

Tugas

- [Langkah 1: Buat metrik CloudWatch khusus](#)
- [Langkah 2: Buat kebijakan penskalaan pelacakan target](#)
- [Langkah 3: Uji kebijakan penskalaan Anda](#)

Langkah 1: Buat metrik CloudWatch khusus

Metrik khusus ditentukan menggunakan nama metrik dan ruang nama pilihan Anda. Ruang nama untuk metrik kustom tidak dapat dimulai dengan `AWS/` Untuk informasi selengkapnya tentang memublikasikan metrik [kustom, lihat topik Publikasikan metrik](#) kustom di CloudWatch Panduan Pengguna Amazon.

Ikuti prosedur ini untuk membuat metrik kustom dengan terlebih dahulu membaca informasi dari AWS akun Anda. Kemudian, hitung backlog per metrik instance, seperti yang disarankan di bagian sebelumnya. Terakhir, publikasikan nomor ini dengan CloudWatch perincian 1 menit. Jika memungkinkan, kami sangat menyarankan agar Anda menskalakan metrik dengan granularitas 1 menit untuk memastikan respons yang lebih cepat terhadap perubahan beban sistem.

Untuk membuat metrik CloudWatch kustom (AWS CLI)

1. Gunakan [get-queue-attributes](#) perintah SQS untuk mendapatkan jumlah pesan yang menunggu di antrian (`ApproximateNumberOfMessages`).

```
aws sqs get-queue-attributes --queue-url https://  
sqs.region.amazonaws.com/123456789/MyQueue \  
--attribute-names ApproximateNumberOfMessages
```

- Gunakan [describe-auto-scaling-groups](#) perintah untuk mendapatkan kapasitas berjalan grup, yang merupakan jumlah instance dalam status InService siklus hidup. Perintah ini mengembalikan instance grup Auto Scaling beserta keadaan siklus aktifnya.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names my-asg
```

- Hitung backlog per instance dengan membagi perkiraan jumlah pesan yang tersedia untuk diambil dari antrian dengan kapasitas berjalan grup.
- Buat skrip yang berjalan setiap menit untuk mengambil backlog per nilai instance dan mempublikasikannya ke metrik CloudWatch khusus. Saat memublikasikan metrik kustom, Anda menentukan nama metrik, namespace, unit, nilai, dan dimensi nol atau lebih. Dimensi terdiri dari nama dimensi dan nilai dimensi.

Untuk mempublikasikan metrik kustom Anda, ganti nilai placeholder dalam *huruf miring* dengan nama metrik pilihan Anda, nilai metrik, namespace (asalkan tidak dimulai dengan "AWS"), dan dimensi (opsional), lalu jalankan perintah berikut. [put-metric-data](#)

```
aws cloudwatch put-metric-data --metric-name MyBacklogPerInstance --
namespace MyNamespace \
  --unit None --value 20 --
dimensions MyOptionalMetricDimensionName=MyOptionalMetricDimensionValue
```

Setelah aplikasi Anda memancarkan metrik yang diinginkan, data dikirim ke CloudWatch. Metrik terlihat di CloudWatch konsol. Anda dapat mengaksesnya dengan masuk ke AWS Management Console dan menavigasi ke halaman. CloudWatch Lalu, lihat metrik dengan menavigasi ke halaman metrik atau dengan mencarinya menggunakan kotak pencarian. Untuk informasi tentang melihat metrik, lihat [Melihat metrik yang tersedia](#) di CloudWatch Panduan Pengguna Amazon.

Langkah 2: Buat kebijakan penskalaan pelacakan target

Metrik yang Anda buat sekarang dapat ditambahkan ke kebijakan penskalaan pelacakan target.

Untuk membuat kebijakan penskalaan pelacakan target (AWS CLI)

- Gunakan cat perintah berikut untuk menyimpan nilai target untuk kebijakan penskalaan Anda dan spesifikasi metrik yang disesuaikan dalam file JSON yang diberi nama `config.json` di direktori home Anda. Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri. Untuk `TargetValue`, hitung backlog yang dapat diterima per metrik instance dan

masukkan di sini. Untuk menghitung angka ini, tentukan nilai latensi normal dan bagi dengan waktu rata-rata yang diperlukan untuk memproses pesan, seperti yang dijelaskan di bagian sebelumnya.

Jika Anda tidak menentukan dimensi apa pun untuk metrik yang Anda buat di langkah 1, jangan sertakan dimensi apa pun dalam spesifikasi metrik yang disesuaikan.

```
$ cat ~/config.json
{
  "TargetValue":100,
  "CustomizedMetricSpecification":{
    "MetricName":"MyBacklogPerInstance",
    "Namespace":"MyNamespace",
    "Dimensions":[
      {
        "Name":"MyOptionalMetricDimensionName",
        "Value":"MyOptionalMetricDimensionValue"
      }
    ],
    "Statistic":"Average",
    "Unit":"None"
  }
}
```

- Gunakan [put-scaling-policy](#) perintah, bersama dengan `config.json` file yang Anda buat di langkah sebelumnya, untuk membuat kebijakan penskalaan Anda.

```
aws autoscaling put-scaling-policy --policy-name sqs100-target-tracking-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://~/config.json
```

Ini menciptakan dua alarm: satu untuk peningkatan skala dan satu untuk penurunan skala. Ini juga mengembalikan Nama Sumber Daya Amazon (ARN) dari kebijakan yang terdaftar CloudWatch, yang CloudWatch digunakan untuk memanggil penskalaan setiap kali ambang metrik dilanggar.

Langkah 3: Uji kebijakan penskalaan Anda

Setelah penyiapan selesai, periksa apakah kebijakan penskalaan Anda berfungsi. Anda dapat mengujinya dengan meningkatkan jumlah pesan dalam antrian SQS Anda dan kemudian memverifikasi bahwa grup Auto Scaling Anda telah meluncurkan instance EC2 tambahan. Anda juga dapat mengujinya dengan mengurangi jumlah pesan dalam antrian SQS Anda dan kemudian memverifikasi bahwa grup Auto Scaling telah menghentikan instance EC2.

Untuk menguji fungsi peningkatan skala

1. Ikuti langkah-langkah dalam [Membuat antrian standar Amazon SQS dan mengirim pesan atau Membuat antrian FIFO Amazon SQS dan mengirim pesan untuk menambahkan pesan ke antrian](#) Anda. Pastikan Anda telah meningkatkan jumlah pesan di antrian sehingga backlog per instance metrik melebihi nilai target.

Diperlukan beberapa menit agar perubahan Anda memanggil alarm.

2. Gunakan [describe-auto-scaling-groups](#) perintah untuk memverifikasi bahwa grup telah meluncurkan sebuah instance.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Untuk menguji fungsi penurunan skala

1. Ikuti langkah-langkah di [Menerima dan menghapus pesan \(konsol\)](#) untuk menghapus pesan dari antrian. Pastikan Anda telah mengurangi jumlah pesan di antrian sehingga backlog per instance metrik berada di bawah nilai target.

Diperlukan beberapa menit agar perubahan Anda memanggil alarm.

2. Gunakan [describe-auto-scaling-groups](#) perintah untuk memverifikasi bahwa grup telah menghentikan sebuah instance.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Amazon SQS dan perlindungan penskalaan instans

Pesan yang belum diproses pada saat instance dihentikan dikembalikan ke antrian SQS di mana mereka dapat diproses oleh instance lain yang masih berjalan. Untuk aplikasi yang menjalankan

tugas yang berjalan lama, Anda dapat menggunakan perlindungan skala masuk instans secara opsional untuk mengontrol pekerja antrian mana yang dihentikan saat grup Auto Scaling Anda masuk.

Pseudocode berikut menunjukkan satu cara untuk melindungi proses pekerja berbasis antrian yang berjalan lama dari penghentian skala.

```
while (true)
{
    SetInstanceProtection(False);
    Work = GetNextWorkUnit();
    SetInstanceProtection(True);
    ProcessWorkUnit(Work);
    SetInstanceProtection(False);
}
```

Untuk informasi selengkapnya, lihat [Rancang aplikasi Anda di Amazon EC2 Auto Scaling untuk menangani penghentian instans dengan baik](#).

Memverifikasi aktivitas penskalaan untuk grup Auto Scaling

Di bagian Auto Scaling Amazon EC2 di konsol Amazon EC2, riwayat Aktivitas untuk grup Auto Scaling memungkinkan Anda melihat status aktivitas penskalaan saat ini yang sedang berlangsung. Ketika aktivitas penskalaan selesai, Anda dapat melihat apakah itu berhasil atau tidak. Ini sangat berguna saat Anda membuat grup Auto Scaling atau Anda menambahkan kondisi penskalaan ke grup yang ada.

Saat menambahkan pelacakan target, langkah, atau kebijakan penskalaan sederhana ke grup Auto Scaling, Auto Scaling Amazon EC2 segera mulai mengevaluasi kebijakan terhadap metrik. Alarm metrik beralih ke status ALARM saat metrik melanggar ambang batas untuk jumlah periode evaluasi tertentu. Ini berarti bahwa kebijakan penskalaan dapat menghasilkan aktivitas penskalaan segera setelah dibuat. Setelah Auto Scaling Amazon EC2 menyesuaikan kapasitas yang diinginkan sebagai respons terhadap kebijakan penskalaan, Anda dapat memverifikasi aktivitas penskalaan di akun Anda. Jika Anda ingin menerima pemberitahuan email dari Amazon EC2 Auto Scaling yang memberi tahu Anda tentang aktivitas penskalaan, ikuti petunjuknya. [Opsional pemberitahuan Amazon SNS untuk Auto Scaling Amazon EC2](#)

Tip

Dalam prosedur berikut, Anda melihat bagian Riwayat Aktivitas dan Instans untuk grup Auto Scaling. Di keduanya, kolom bernama seharusnya sudah ditampilkan. Untuk menampilkan kolom tersembunyi atau mengubah jumlah baris yang ditampilkan, pilih ikon roda gigi di sudut kanan atas setiap bagian untuk membuka modal preferensi, perbarui pengaturan sesuai kebutuhan, dan pilih Konfirmasi.

Untuk melihat aktivitas penskalaan untuk grup Auto Scaling (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Di bilah navigasi di bagian atas layar, pilih Wilayah grup Auto Scaling Anda.
3. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

4. Pada tab Aktivitas, di bawah Riwayat aktivitas, kolom Status menunjukkan apakah grup Auto Scaling Anda telah berhasil meluncurkan atau menghentikan instance, atau apakah aktivitas penskalaan masih berlangsung.
5. (Opsional) Jika Anda memiliki banyak aktivitas penskalaan, Anda dapat memilih ikon > di tepi atas riwayat aktivitas untuk melihat halaman aktivitas penskalaan berikutnya.
6. Pada tab Manajemen instans, di bawah Instance, kolom Siklus Hidup berisi status instance Anda. Setelah instance dimulai dan kait siklus hidup apa pun selesai, status siklus hidupnya berubah menjadi. InService Kolom Status kesehatan menunjukkan hasil pemeriksaan kesehatan instans EC2 pada instans Anda.

Untuk melihat aktivitas penskalaan untuk grup Auto Scaling (AWS CLI)

Gunakan perintah [describe-scaling-activities](#) berikut ini.

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg
```

Berikut ini adalah output contoh.

Aktivitas penskalaan diurutkan berdasarkan waktu mulai. Aktivitas masih dalam proses dijelaskan terlebih dahulu.

```
{
  "Activities": [
    {
      "ActivityId": "5e3a1f47-2309-415c-bfd8-35aa06300799",
      "AutoScalingGroupName": "my-asg",
      "Description": "Terminating EC2 instance: i-06c4794c2499af1df",
      "Cause": "At 2020-02-11T18:34:10Z a monitor alarm TargetTracking-my-asg-AlarmLow-
b9376cab-18a7-4385-920c-dfa3f7783f82 in state ALARM triggered policy my-target-
tracking-policy changing the desired capacity from 3 to 2. At 2020-02-11T18:34:31Z
an instance was taken out of service in response to a difference between desired and
actual capacity, shrinking the capacity from 3 to 2. At 2020-02-11T18:34:31Z instance
i-06c4794c2499af1df was selected for termination.",
      "StartTime": "2020-02-11T18:34:31.268Z",
      "EndTime": "2020-02-11T18:34:53Z",
      "StatusCode": "Successful",
      "Progress": 100,
      "Details": "{\"Subnet ID\": \"subnet-5ea0c127\", \"Availability Zone\": \"us-west-2a
\"...}\",
      "AutoScalingGroupARN": "arn"
    },
    ...
  ]
}
```

Untuk deskripsi bidang dalam output, lihat [Aktivitas](#) di Referensi API Auto Scaling Amazon EC2.

Untuk bantuan mengambil aktivitas penskalaan untuk grup yang dihapus dan untuk informasi tentang jenis kesalahan yang mungkin Anda temui dan cara menanganinya, lihat [Memecahkan Masalah Auto Scaling Amazon EC2](#)

Menonaktifkan kebijakan penskalaan untuk grup Auto Scaling

Topik ini menjelaskan cara menonaktifkan sementara kebijakan penskalaan sehingga tidak akan memulai perubahan pada jumlah instance yang dimuat oleh grup Auto Scaling. Ketika Anda menonaktifkan kebijakan penskalaan, detail konfigurasi dipertahankan, sehingga Anda dapat dengan cepat mengaktifkan ulang kebijakan tersebut. Ini lebih mudah daripada menghapus kebijakan untuk sementara saat Anda tidak memerlukannya, dan membuat ulang kebijakan nanti.

Ketika kebijakan penskalaan dinonaktifkan, grup Auto Scaling tidak menaikkan atau menurunkan skala untuk alarm metrik yang dilanggar saat kebijakan penskalaan dinonaktifkan. Namun, semua aktivitas penskalaan yang sedang berlangsung tidak dihentikan.

Penting untuk diperhatikan bahwa kebijakan penskalaan yang dinonaktifkan masih dihitung terhadap kuota Anda pada jumlah kebijakan penskalaan yang dapat Anda tambahkan ke grup Auto Scaling.

Untuk menonaktifkan kebijakan penskalaan (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Penskalaan otomatis, di bawah Kebijakan penskalaan dinamis, pilih kotak centang di sudut kanan atas kebijakan penskalaan yang diinginkan.
4. Gulir ke bagian atas kebijakan penskalaan dinamis, lalu pilih Tindakan, Nonaktifkan.

Setelah Anda siap untuk mengaktifkan kembali kebijakan penskalaan, ulangi langkah-langkah ini, lalu pilih Tindakan, Aktifkan. Setelah Anda mengaktifkan ulang kebijakan penskalaan, grup Auto Scaling Anda dapat segera memulai tindakan penskalaan jika ada alarm yang saat ini berada dalam status ALARM.

Untuk menonaktifkan kebijakan penskalaan (AWS CLI)

Gunakan [put-scaling-policy](#) perintah dengan `--no-enabled` opsi sebagai berikut. Tentukan semua opsi dalam perintah tersebut seperti yang Anda akan tentukan saat membuat kebijakan.

```
aws autoscaling put-scaling-policy --auto-scaling-group-name my-asg \  
  --policy-name my-scaling-policy --policy-type TargetTrackingScaling \  
  --estimated-instance-warmup 360 \  
  --target-tracking-configuration '{ "TargetValue": 70,  
"PredefinedMetricSpecification": { "PredefinedMetricType":  
"ASGAverageCPUUtilization" } }' \  
  --no-enabled
```

Untuk mengaktifkan kembali kebijakan penskalaan (AWS CLI)

Gunakan [put-scaling-policy](#) perintah dengan `--enabled` opsi sebagai berikut. Tentukan semua opsi dalam perintah tersebut seperti yang Anda akan tentukan saat membuat kebijakan.

```
aws autoscaling put-scaling-policy --auto-scaling-group-name my-asg \  
  --policy-name my-scaling-policy --policy-type TargetTrackingScaling \  
  --estimated-instance-warmup 360 \  
  --enabled
```

```
--target-tracking-configuration '{ "TargetValue": 70,
"PredefinedMetricSpecification": { "PredefinedMetricType":
"ASGAverageCPUUtilization" } }' \
--enabled
```

Untuk menggambarkan kebijakan penskalaan ()AWS CLI

Gunakan perintah [describe-policies](#) untuk memverifikasi status kebijakan penskalaan yang diaktifkan.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg \
--policy-names my-scaling-policy
```

Berikut ini adalah output contoh.

```
{
  "ScalingPolicies": [
    {
      "AutoScalingGroupName": "my-asg",
      "PolicyName": "my-scaling-policy",
      "PolicyARN": "arn:aws:autoscaling:us-
west-2:123456789012:scalingPolicy:1d52783a-b03b-4710-
bb0e-549fd64378cc:autoScalingGroupName/my-asg:policyName/my-scaling-policy",
      "PolicyType": "TargetTrackingScaling",
      "StepAdjustments": [],
      "Alarms": [
        {
          "AlarmName": "TargetTracking-my-asg-
AlarmHigh-9ca53fdd-7cf5-4223-938a-ae1199204502",
          "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-9ca53fdd-7cf5-4223-938a-
ae1199204502"
        },
        {
          "AlarmName": "TargetTracking-my-asg-AlarmLow-7010c83d-d55a-4a7a-
abe0-1cf8b9de6d6c",
          "AlarmARN": "arn:aws:cloudwatch:us-
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-7010c83d-d55a-4a7a-
abe0-1cf8b9de6d6c"
        }
      ],
      "TargetTrackingConfiguration": {
        "PredefinedMetricSpecification": {
          "PredefinedMetricType": "ASGAverageCPUUtilization"
```

```
        },
        "TargetValue": 70.0,
        "DisableScaleIn": false
    },
    "Enabled": true
}
]
```

Menghapus kebijakan penskalaan

Setelah Anda tidak lagi memerlukan kebijakan penskalaan, Anda dapat menghapusnya. Bergantung pada jenis kebijakan penskalaan, Anda mungkin juga perlu menghapus CloudWatch alarm. Menghapus kebijakan penskalaan pelacakan target juga akan menghapus alarm terkait. CloudWatch Menghapus kebijakan penskalaan langkah atau kebijakan penskalaan sederhana akan menghapus tindakan alarm yang mendasarinya, tetapi tidak menghapus CloudWatch alarm, meskipun tidak lagi memiliki tindakan terkait.

Untuk menghapus kebijakan penskalaan (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Penskalaan otomatis, di bawah Kebijakan penskalaan dinamis, pilih kotak centang di sudut kanan atas kebijakan penskalaan yang diinginkan.
4. Gulir ke bagian atas bagian Kebijakan penskalaan dinamis, lalu pilih Tindakan, Hapus.
5. Ketika diminta untuk mengonfirmasi, pilih Ya, Hapus.
6. (Opsional) Jika Anda menghapus kebijakan penskalaan langkah atau kebijakan penskalaan sederhana, lakukan hal berikut untuk menghapus CloudWatch alarm yang terkait dengan kebijakan tersebut. Anda dapat melewati sublangkah ini untuk menjaga alarm tetap berfungsi di masa mendatang.
 - a. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
 - b. Pada panel navigasi, pilih Alarm.
 - c. Pilih alarm (misalnya, Step-Scaling-AlarmHigh-AddCapacity) dan pilih Tindakan, Hapus.

- d. Saat diminta konfirmasi, pilih Hapus.

Untuk mendapatkan kebijakan penskalaan untuk grup Auto Scaling ()AWS CLI

Sebelum menghapus kebijakan penskalaan, gunakan [perintah deskripsikan kebijakan berikut untuk melihat kebijakan penskalaan](#) apa yang dibuat untuk grup Auto Scaling. Anda dapat menggunakan output saat menghapus kebijakan dan CloudWatch alarm.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg
```

Anda dapat memfilter hasil berdasarkan jenis kebijakan penskalaan menggunakan parameter `--query`. Sintaks ini untuk query bekerja di Linux atau macOS. Di Windows, ubah tanda kutip tunggal menjadi tanda kutip ganda.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg  
--query 'ScalingPolicies[?PolicyType=='TargetTrackingScaling']'
```

Berikut ini adalah output contoh.

```
[  
  {  
    "AutoScalingGroupName": "my-asg",  
    "PolicyName": "cpu50-target-tracking-scaling-policy",  
    "PolicyARN": "PolicyARN",  
    "PolicyType": "TargetTrackingScaling",  
    "StepAdjustments": [],  
    "Alarms": [  
      {  
        "AlarmARN": "arn:aws:cloudwatch:us-  
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmHigh-  
fc0e4183-23ac-497e-9992-691c9980c38e",  
        "AlarmName": "TargetTracking-my-asg-AlarmHigh-  
fc0e4183-23ac-497e-9992-691c9980c38e"  
      },  
      {  
        "AlarmARN": "arn:aws:cloudwatch:us-  
west-2:123456789012:alarm:TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-  
bd9e-471a352ee1a2",  
        "AlarmName": "TargetTracking-my-asg-AlarmLow-61a39305-ed0c-47af-  
bd9e-471a352ee1a2"  
      }  
    ]  
  }  
]
```

```

    }
  ],
  "TargetTrackingConfiguration": {
    "PredefinedMetricSpecification": {
      "PredefinedMetricType": "ASGAverageCPUUtilization"
    },
    "TargetValue": 50.0,
    "DisableScaleIn": false
  },
  "Enabled": true
}
]

```

Untuk menghapus kebijakan penskalaan Anda (AWS CLI)

Gunakan perintah [delete-policy](#) berikut.

```
aws autoscaling delete-policy --auto-scaling-group-name my-asg \
--policy-name cpu50-target-tracking-scaling-policy
```

Untuk menghapus CloudWatch alarm Anda (AWS CLI)

Untuk langkah dan kebijakan penskalaan sederhana, gunakan perintah [hapus-alarm](#) untuk menghapus CloudWatch alarm yang terkait dengan kebijakan. Anda dapat melewati langkah ini untuk menjaga alarm tetap berfungsi di masa mendatang. Anda dapat menghapus satu atau beberapa alarm sekaligus. Misalnya, gunakan perintah berikut untuk menghapus alarm Step-Scaling-AlarmHigh-AddCapacity dan Step-Scaling-AlarmLow-RemoveCapacity.

```
aws cloudwatch delete-alarms --alarm-name Step-Scaling-AlarmHigh-AddCapacity Step-Scaling-AlarmLow-RemoveCapacity
```

Contoh kebijakan penskalaan untuk AWS Command Line Interface (AWS CLI)

Anda dapat membuat kebijakan penskalaan untuk Auto Scaling Amazon EC2 melalui, atau AWS Management Console SDK AWS CLI.

Contoh berikut menunjukkan bagaimana Anda dapat membuat kebijakan penskalaan untuk Amazon EC2 Auto Scaling dengan perintah. AWS CLI [put-scaling-policy](#) Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri.

Untuk memulai menulis kebijakan penskalaan menggunakan AWS CLI, lihat latihan pengantar di [Kebijakan penskalaan pelacakan target](#) dan [Kebijakan penskalaan langkah dan sederhana](#)

Contoh 1: Untuk menerapkan kebijakan penskalaan pelacakan target dengan spesifikasi metrik yang telah ditentukan

```
aws autoscaling put-scaling-policy --policy-name cpu50-target-tracking-scaling-policy \  
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \  
  --target-tracking-configuration file://config.json  
{  
  "TargetValue": 50.0,  
  "PredefinedMetricSpecification": {  
    "PredefinedMetricType": "ASGAverageCPUUtilization"  
  }  
}
```

Untuk informasi selengkapnya, lihat [PredefinedMetricSpecification](#) di Referensi API Auto Scaling Amazon EC2.

Note

Jika file tidak ada di direktori saat ini, ketik path lengkap ke file. Untuk informasi selengkapnya tentang membaca nilai AWS CLI parameter dari file, lihat [Memuat AWS CLI parameter dari file](#) di Panduan AWS Command Line Interface Pengguna.

Contoh 2: Untuk menerapkan kebijakan penskalaan pelacakan target dengan spesifikasi metrik yang disesuaikan

```
aws autoscaling put-scaling-policy --policy-name sqs100-target-tracking-scaling-policy \  
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \  
  --target-tracking-configuration file://config.json  
{  
  "TargetValue": 100.0,  
  "CustomizedMetricSpecification": {  
    "MetricName": "MyBacklogPerInstance",  
    "Namespace": "MyNamespace",  
    "Dimensions": [{  
      "Name": "MyOptionalMetricDimensionName",  
      "Value": "MyOptionalMetricDimensionValue"  
    }  
  ]  
}
```

```

    }],
    "Statistic": "Average",
    "Unit": "None"
  }
}

```

Untuk informasi selengkapnya, lihat [CustomizedMetricSpecification](#) di Referensi API Auto Scaling Amazon EC2.

Contoh 3: Untuk menerapkan kebijakan penskalaan pelacakan target hanya untuk skala keluar

```

aws autoscaling put-scaling-policy --policy-name alb1000-target-tracking-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type TargetTrackingScaling \
  --target-tracking-configuration file://config.json
{
  "TargetValue": 1000.0,
  "PredefinedMetricSpecification": {
    "PredefinedMetricType": "ALBRequestCountPerTarget",
    "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-target-
group/943f017f100becff"
  },
  "DisableScaleIn": true
}

```

Contoh 4: Untuk menerapkan kebijakan penskalaan langkah untuk skala keluar

```

aws autoscaling put-scaling-policy \
  --auto-scaling-group-name my-asg \
  --policy-name my-step-scale-out-policy \
  --policy-type StepScaling \
  --adjustment-type PercentChangeInCapacity \
  --metric-aggregation-type Average \
  --step-adjustments
MetricIntervalLowerBound=10.0,MetricIntervalUpperBound=20.0,ScalingAdjustment=10 \
MetricIntervalLowerBound=20.0,MetricIntervalUpperBound=30.0,ScalingAdjustment=20 \
  MetricIntervalLowerBound=30.0,ScalingAdjustment=30 \
  --min-adjustment-magnitude 1

```

Merekam Amazon Resource Name (ARN) kebijakan. Anda memerlukan ARN saat Anda membuat alarm. CloudWatch

Contoh 5: Untuk menerapkan kebijakan penskalaan langkah untuk skala

```
aws autoscaling put-scaling-policy \  
  --auto-scaling-group-name my-asg \  
  --policy-name my-step-scale-in-policy \  
  --policy-type StepScaling \  
  --adjustment-type ChangeInCapacity \  
  --step-adjustments MetricIntervalUpperBound=0.0,ScalingAdjustment=-2
```

Merekam Amazon Resource Name (ARN) kebijakan. Anda memerlukan ARN saat Anda membuat alarm. CloudWatch

Contoh 6: Untuk menerapkan kebijakan penskalaan sederhana untuk skala keluar

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-out-policy \  
  --auto-scaling-group-name my-asg --scaling-adjustment 30 \  
  --adjustment-type PercentChangeInCapacity --min-adjustment-magnitude 2
```

Merekam Amazon Resource Name (ARN) kebijakan. Anda memerlukan ARN saat Anda membuat alarm. CloudWatch

Contoh 7: Untuk menerapkan kebijakan penskalaan sederhana untuk skala

```
aws autoscaling put-scaling-policy --policy-name my-simple-scale-in-policy \  
  --auto-scaling-group-name my-asg --scaling-adjustment -1 \  
  --adjustment-type ChangeInCapacity --cooldown 180
```

Merekam Amazon Resource Name (ARN) kebijakan. Anda memerlukan ARN saat Anda membuat alarm. CloudWatch

Penskalaan prediktif untuk Auto Scaling Amazon EC2

Penskalaan prediktif bekerja dengan menganalisis data beban historis untuk mendeteksi pola harian atau mingguan dalam arus lalu lintas. Ini menggunakan informasi ini untuk memperkirakan kebutuhan kapasitas masa depan sehingga Amazon EC2 Auto Scaling dapat secara proaktif meningkatkan kapasitas grup Auto Scaling Anda agar sesuai dengan beban yang diantisipasi.

Penskalaan prediktif sangat cocok untuk situasi di mana Anda memiliki:

- Lalu lintas siklus, seperti penggunaan sumber daya yang tinggi selama jam kerja reguler dan penggunaan sumber daya yang rendah selama malam hari dan akhir pekan
- Pola on-and-off beban kerja berulang, seperti pemrosesan batch, pengujian, atau analisis data berkala
- Aplikasi yang membutuhkan waktu lama untuk diinisialisasi, menyebabkan dampak latensi yang nyata pada kinerja aplikasi selama peristiwa penskalaan

Secara umum, jika Anda memiliki pola peningkatan lalu lintas reguler dan aplikasi yang membutuhkan waktu lama untuk diinisialisasi, Anda harus mempertimbangkan untuk menggunakan penskalaan prediktif. Penskalaan prediktif dapat membantu Anda menskalakan lebih cepat dengan meluncurkan kapasitas sebelum beban yang diperkirakan, dibandingkan dengan hanya menggunakan penskalaan dinamis, yang bersifat reaktif. Penskalaan prediktif juga berpotensi menghemat uang pada tagihan EC2 Anda dengan membantu Anda menghindari kebutuhan untuk kapasitas penyediaan yang berlebihan.

Misalnya, pertimbangkan aplikasi yang memiliki penggunaan tinggi selama jam kerja dan penggunaan rendah dalam semalam. Pada awal setiap hari kerja, penskalaan prediktif dapat menambah kapasitas sebelum masuknya lalu lintas pertama. Ini membantu aplikasi Anda mempertahankan ketersediaan dan kinerja tinggi saat beralih dari periode pemanfaatan yang lebih rendah ke periode pemanfaatan yang lebih tinggi. Anda tidak perlu menunggu penskalaan dinamis untuk bereaksi terhadap perubahan lalu lintas. Anda juga tidak perlu menghabiskan waktu untuk meninjau pola pemuatan aplikasi Anda dan mencoba menjadwalkan jumlah kapasitas yang tepat menggunakan penskalaan terjadwal.

Topik

- [Cara kerja penskalaan prediktif](#)
- [Buat kebijakan penskalaan prediktif](#)
- [Evaluasi kebijakan penskalaan prediktif Anda](#)
- [Ganti nilai perkiraan menggunakan tindakan terjadwal](#)
- [Konfigurasi kebijakan penskalaan prediktif lanjutan menggunakan metrik khusus](#)

Cara kerja penskalaan prediktif

Topik ini menjelaskan cara kerja penskalaan prediktif dan menjelaskan apa yang harus dipertimbangkan saat Anda membuat kebijakan penskalaan prediktif.

Topik

- [Cara kerjanya](#)
- [Batas kapasitas maksimum](#)
- [Pertimbangan](#)
- [Wilayah yang Didukung](#)

Cara kerjanya

Untuk menggunakan penskalaan prediktif, buat kebijakan penskalaan prediktif yang menentukan CloudWatch metrik untuk dipantau dan dianalisis. Agar penskalaan prediktif mulai meramalkan nilai masa depan, metrik ini harus memiliki setidaknya 24 jam data.

Setelah Anda membuat kebijakan, penskalaan prediktif mulai menganalisis data metrik hingga 14 hari terakhir untuk mengidentifikasi pola. Ini menggunakan analisis ini untuk menghasilkan perkiraan per jam persyaratan kapasitas untuk 48 jam ke depan. Prakiraan diperbarui setiap 6 jam menggunakan CloudWatch data terbaru. Saat data baru masuk, penskalaan prediktif mampu terus meningkatkan akurasi prakiraan masa depan.

Saat Anda pertama kali mengaktifkan penskalaan prediktif, penskalaan ini berjalan dalam mode perkiraan saja. Dalam mode ini, ini menghasilkan perkiraan kapasitas tetapi tidak benar-benar menskalakan grup Auto Scaling Anda berdasarkan perkiraan tersebut. Ini memungkinkan Anda untuk mengevaluasi keakuratan dan kesesuaian ramalan. Anda dapat melihat data perkiraan dengan menggunakan operasi `GetPredictiveScalingForecast` API atau AWS Management Console.

Setelah Anda meninjau data perkiraan dan memutuskan untuk memulai penskalaan berdasarkan data tersebut, alihkan kebijakan penskalaan ke mode perkiraan dan skala. Dalam mode ini:

- Jika perkiraan mengharapkan peningkatan beban, Auto Scaling Amazon EC2 akan meningkatkan kapasitas dengan meningkatkan skala.
- Jika perkiraan mengharapkan penurunan beban, itu tidak akan skala untuk menghapus kapasitas. Jika Anda ingin menghapus kapasitas yang tidak lagi diperlukan, Anda harus membuat kebijakan penskalaan dinamis.

Secara default, Amazon EC2 Auto Scaling menskalakan grup Auto Scaling Anda pada awal setiap jam berdasarkan perkiraan untuk jam tersebut. Anda dapat menentukan waktu mulai yang lebih awal secara opsional dengan menggunakan `SchedulingBufferTime` properti dalam operasi

`PutScalingPolicy` API atau setelah instans Pra-peluncuran di. AWS Management Console Hal ini menyebabkan Amazon EC2 Auto Scaling meluncurkan instans baru sebelum permintaan yang diperkirakan, memberi mereka waktu untuk boot dan siap menangani lalu lintas.

Untuk mendukung peluncuran instans baru sebelum permintaan yang diperkirakan, kami sangat menyarankan agar Anda mengaktifkan pemanasan instans default untuk grup Auto Scaling Anda. Ini menentukan periode waktu setelah aktivitas scale-out selama Auto Scaling Amazon EC2 tidak akan diskalakan, meskipun kebijakan penskalaan dinamis menunjukkan kapasitas harus dikurangi. Ini membantu Anda memastikan bahwa instans yang baru diluncurkan memiliki waktu yang cukup untuk mulai melayani peningkatan lalu lintas sebelum dipertimbangkan untuk operasi skala. Untuk informasi selengkapnya, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#).

Batas kapasitas maksimum

Grup Auto Scaling memiliki pengaturan kapasitas maksimum yang membatasi jumlah maksimum instans EC2 yang dapat diluncurkan untuk grup. Secara default, ketika kebijakan penskalaan ditetapkan, mereka tidak dapat meningkatkan kapasitas lebih tinggi dari kapasitas maksimumnya.

Atau, Anda dapat mengizinkan kapasitas maksimum grup ditingkatkan secara otomatis jika kapasitas perkiraan mendekati atau melebihi kapasitas maksimum grup Auto Scaling. Untuk mengaktifkan perilaku ini, gunakan `MaxCapacityBuffer` properti `MaxCapacityBreachBehavior` dan dalam operasi `PutScalingPolicy` API atau setelah perilaku kapasitas Maks di AWS Management Console.

Warning

Berhati-hatilah saat memungkinkan kapasitas maksimum ditingkatkan secara otomatis. Hal ini dapat menyebabkan lebih banyak instance diluncurkan daripada yang dimaksudkan jika peningkatan kapasitas maksimum tidak dipantau dan dikelola. Peningkatan kapasitas maksimum kemudian menjadi kapasitas maksimum normal baru untuk grup Auto Scaling hingga Anda memperbaruinya secara manual. Kapasitas maksimum tidak secara otomatis berkurang kembali ke maksimum asli.

Pertimbangan

- Konfirmasikan apakah penskalaan prediktif cocok untuk beban kerja Anda. Beban kerja sangat cocok untuk penskalaan prediktif jika menunjukkan pola beban berulang yang spesifik untuk hari dalam seminggu atau waktu dalam sehari. Untuk memeriksanya, konfigurasi kebijakan

penskalaan prediktif dalam mode hanya perkiraan, lalu lihat rekomendasi di konsol. Auto Scaling Amazon EC2 memberikan rekomendasi berdasarkan pengamatan tentang potensi kinerja kebijakan. Evaluasi perkiraan dan rekomendasi sebelum membiarkan penskalaan prediktif secara aktif menskalakan aplikasi Anda.

- Penskalaan prediktif membutuhkan setidaknya 24 jam data historis untuk memulai peramalan. Namun, perkiraan lebih efektif jika data historis mencakup dua minggu penuh. Jika Anda memperbarui aplikasi Anda dengan membuat grup Auto Scaling baru dan menghapus yang lama, maka grup Auto Scaling baru Anda memerlukan 24 jam data pemuatan historis sebelum penskalaan prediktif dapat mulai menghasilkan prakiraan lagi. Anda dapat menggunakan metrik khusus untuk menggabungkan metrik di seluruh grup Auto Scaling lama dan baru. Jika tidak, Anda mungkin harus menunggu beberapa hari untuk perkiraan yang lebih akurat.
- Pilih metrik pemuatan yang secara akurat mewakili beban penuh pada aplikasi Anda dan merupakan aspek aplikasi Anda yang paling penting untuk diskalakan.
- Menggunakan penskalaan dinamis dengan penskalaan prediktif membantu Anda mengikuti kurva permintaan untuk aplikasi Anda dengan cermat, penskalaan selama periode lalu lintas rendah dan penskalaan saat lalu lintas lebih tinggi dari yang diharapkan. Ketika beberapa kebijakan penskalaan aktif, setiap kebijakan menentukan kapasitas yang diinginkan secara independen, dan kapasitas yang diinginkan diatur secara maksimal. Misalnya, jika 10 instans diperlukan untuk tetap pada pemanfaatan target dalam kebijakan penskalaan pelacakan target, dan 8 instance diperlukan untuk tetap pada pemanfaatan target dalam kebijakan penskalaan prediktif, maka kapasitas yang diinginkan grup ditetapkan ke 10. Jika Anda baru mengenal penskalaan dinamis, sebaiknya gunakan kebijakan penskalaan pelacakan target. Untuk informasi selengkapnya, lihat [Penskalaan dinamis untuk Amazon EC2 Auto Scaling](#).
- Asumsi inti dari penskalaan prediktif adalah bahwa kelompok Auto Scaling homogen dan semua instance memiliki kapasitas yang sama. Jika ini tidak benar untuk grup Anda, kapasitas yang diperkirakan bisa tidak akurat. Oleh karena itu, berhati-hatilah saat membuat kebijakan penskalaan prediktif untuk [grup instance campuran](#) karena instance dari berbagai jenis dapat disediakan dengan kapasitas yang tidak sama. Berikut adalah beberapa contoh di mana kapasitas yang diperkirakan akan tidak akurat:
 - Kebijakan penskalaan prediktif Anda didasarkan pada pemanfaatan CPU, tetapi jumlah vCPU pada setiap instans Auto Scaling bervariasi antar jenis instans.
 - Kebijakan penskalaan prediktif Anda didasarkan pada jaringan masuk atau keluar jaringan, tetapi throughput bandwidth jaringan untuk setiap instans Auto Scaling bervariasi antar jenis instans. Misalnya, tipe instans M5 dan M5n serupa, tetapi tipe instans M5n memberikan throughput jaringan yang jauh lebih tinggi.

Wilayah yang Didukung

Auto Scaling Amazon EC2 mendukung kebijakan penskalaan prediktif Wilayah AWS sebagai berikut: AS Timur (Virginia N.), AS Timur (Ohio), AS Barat (Oregon), AS Barat (California N.), Afrika (Cape Town), Kanada (Tengah), UE (Frankfurt), UE (Irlandia), UE (London), UE (Milan), UE (Paris), UE (Stockholm), Asia Pasifik (Hong Kong), Asia Pasifik (Jakarta), Asia Pasifik (Mumbai), Asia Pasifik (Osaka), Asia Pasifik (Tokyo), Asia Pasifik (Singapura), Asia Pasifik (Seoul), Asia Pasifik (Sydney), Timur Tengah (Bahrain), Timur Tengah (UEA), Amerika Selatan (Sao Paulo), China (Beijing), China (Ningxia), AWS GovCloud (AS-Timur), dan AWS GovCloud (AS-Barat).

Buat kebijakan penskalaan prediktif

Prosedur berikut membantu Anda membuat kebijakan penskalaan prediktif menggunakan atau. AWS Management Console AWS CLI

Jika grup Auto Scaling baru, grup Auto Scaling harus menyediakan setidaknya 24 jam data sebelum Auto Scaling Amazon EC2 dapat menghasilkan perkiraan untuk itu.

Daftar Isi

- [Membuat kebijakan penskalaan prediktif \(konsol\)](#)
- [Buat kebijakan penskalaan prediktif \(\)AWS CLI](#)

Membuat kebijakan penskalaan prediktif (konsol)

Jika ini adalah pertama kalinya Anda membuat kebijakan penskalaan prediktif, sebaiknya gunakan konsol untuk membuat beberapa kebijakan penskalaan prediktif dalam mode perkiraan saja. Ini memungkinkan Anda untuk menguji efek potensial dari berbagai metrik dan nilai target. Anda dapat membuat beberapa kebijakan penskalaan prediktif untuk setiap grup Auto Scaling, tetapi hanya satu kebijakan yang dapat digunakan untuk penskalaan aktif.

Membuat kebijakan penskalaan prediktif di konsol (metrik yang telah ditentukan sebelumnya)

Gunakan prosedur berikut untuk membuat kebijakan penskalaan prediktif menggunakan metrik yang telah ditentukan (CPU, I/O jaringan, atau jumlah permintaan Application Load Balancer per target). Cara termudah untuk membuat kebijakan penskalaan prediktif adalah dengan menggunakan metrik yang telah ditentukan sebelumnya. Jika Anda lebih suka menggunakan metrik khusus, lihat [Membuat kebijakan penskalaan prediktif di konsol \(metrik khusus\)](#).

Untuk membuat kebijakan penskalaan prediktif

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Penskalaan otomatis, di Kebijakan penskalaan, pilih Buat kebijakan penskalaan prediktif.
4. Masukkan nama untuk kebijakan.
5. Aktifkan Skala berdasarkan perkiraan untuk memberikan izin Auto Scaling Amazon EC2 untuk segera memulai penskalaan.

Untuk menjaga kebijakan dalam mode hanya perkiraan, matikan Skala berdasarkan perkiraan.

6. Untuk Metrik, pilih metrik Anda dari daftar opsi. Opsi termasuk CPU, Network In, Network Out, jumlah permintaan Application Load Balancer, dan pasangan metrik kustom.

Jika Anda memilih jumlah permintaan Application Load Balancer per target, maka pilih grup target di grup Target. Jumlah permintaan Application Load Balancer per target hanya didukung jika Anda telah melampirkan grup target Application Load Balancer ke grup Auto Scaling Anda.

Jika Anda memilih Pasangan metrik kustom, pilih metrik individual dari daftar drop-down untuk metrik Load dan Scaling metric.

7. Untuk pemanfaatan Target, masukkan nilai target yang harus dipertahankan oleh Auto Scaling Amazon EC2. Amazon EC2 Auto Scaling meningkatkan kapasitas Anda hingga penggunaan rata-rata sesuai dengan target penggunaan, atau hingga mencapai jumlah maksimum instans yang Anda tentukan.

Jika metrik penskalaan Anda...	Maka pemanfaatan target mewakili...
CPU	Persentase CPU yang idealnya digunakan setiap instance.
Jaringan Dalam	Jumlah rata-rata byte per menit yang idealnya diterima oleh setiap instance.

Jika metrik penskalaan Anda...	Maka pemanfaatan target mewakili...
Jaringan Keluar	Jumlah rata-rata byte per menit yang idealnya dikirim oleh setiap instance.
Application Load Balancer jumlah permintaan per target	Jumlah rata-rata permintaan per menit yang idealnya diterima oleh setiap instance.

8. (Opsional) Untuk instans Pra-peluncuran, pilih seberapa jauh sebelumnya Anda ingin instans diluncurkan sebelum perkiraan panggilan agar beban meningkat.
9. (Opsional) Untuk perilaku kapasitas Maks, pilih apakah akan membiarkan Auto Scaling Amazon EC2 keluar lebih tinggi dari kapasitas maksimum grup jika kapasitas yang diprediksi melebihi maksimum yang ditentukan. Mengaktifkan pengaturan ini memungkinkan skala keluar terjadi selama periode ketika lalu lintas Anda diperkirakan berada pada titik tertinggi.
10. (Opsional) Untuk kapasitas maksimum Buffer di atas kapasitas yang diperkirakan, pilih berapa banyak kapasitas tambahan yang akan digunakan ketika kapasitas yang diprediksi mendekati atau melebihi kapasitas maksimum. Nilai ditentukan sebagai persentase relatif terhadap kapasitas yang diprediksi. Misalnya, jika buffer adalah 10, ini berarti buffer 10 persen. Oleh karena itu, jika kapasitas yang diprediksi adalah 50 dan kapasitas maksimum adalah 40, kapasitas maksimum efektif adalah 55.

Jika disetel ke 0, Auto Scaling Amazon EC2 mungkin menskalakan kapasitas lebih tinggi dari kapasitas maksimum hingga sama tetapi tidak melebihi kapasitas yang diprediksi.


11. Pilih Buat kebijakan penskalaan prediktif.

Membuat kebijakan penskalaan prediktif di konsol (metrik khusus)

Gunakan prosedur berikut untuk membuat kebijakan penskalaan prediktif menggunakan metrik kustom. Metrik khusus dapat mencakup metrik lain yang disediakan oleh CloudWatch atau metrik yang Anda publikasikan. CloudWatch Untuk menggunakan jumlah permintaan CPU, I/O jaringan, atau Application Load Balancer per target, lihat. [Membuat kebijakan penskalaan prediktif di konsol \(metrik yang telah ditentukan sebelumnya\)](#)

Untuk membuat kebijakan penskalaan prediktif menggunakan metrik kustom, Anda harus melakukan hal berikut:

- Anda harus memberikan kueri mentah yang memungkinkan Auto Scaling Amazon EC2 berinteraksi dengan metrik di dalamnya. CloudWatch Untuk informasi selengkapnya, lihat [Konfigurasi kebijakan penskalaan prediktif lanjutan menggunakan metrik khusus](#). Untuk memastikan bahwa Auto Scaling Amazon EC2 dapat mengekstrak data CloudWatch metrik, konfirmasi bahwa setiap kueri mengembalikan titik data. Konfirmasikan ini dengan menggunakan CloudWatch konsol atau operasi CloudWatch [GetMetricData](#)API.

 Note

Kami menyediakan contoh muatan JSON di editor JSON di konsol Auto Scaling Amazon EC2. Contoh-contoh ini memberi Anda referensi untuk pasangan nilai kunci yang diperlukan untuk menambahkan metrik lain yang disediakan oleh AWS atau CloudWatch metrik yang sebelumnya Anda publikasikan. CloudWatch Anda dapat menggunakannya sebagai titik awal, lalu menyesuaikannya dengan kebutuhan Anda.

- Jika Anda menggunakan matematika metrik apa pun, Anda harus membuat JSON secara manual agar sesuai dengan skenario unik Anda. Untuk informasi selengkapnya, lihat [Gunakan ekspresi matematika metrik](#). Sebelum menggunakan matematika metrik dalam kebijakan Anda, konfirmasi bahwa kueri metrik berdasarkan ekspresi matematika metrik valid dan kembalikan satu deret waktu. Konfirmasikan ini dengan menggunakan CloudWatch konsol atau operasi CloudWatch [GetMetricData](#)API.

Jika Anda membuat kesalahan dalam kueri dengan memberikan data yang salah, seperti nama grup Auto Scaling yang salah, perkiraan tidak akan memiliki data apa pun. Untuk memecahkan masalah metrik kustom, lihat. [Pertimbangan dan pemecahan masalah](#)

Untuk membuat kebijakan penskalaan prediktif

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Penskalaan otomatis, di Kebijakan penskalaan, pilih Buat kebijakan penskalaan prediktif.
4. Masukkan nama untuk kebijakan.

5. Aktifkan Skala berdasarkan perkiraan untuk memberikan izin Auto Scaling Amazon EC2 untuk segera memulai penskalaan.

Untuk menjaga kebijakan dalam mode hanya perkiraan, matikan Skala berdasarkan perkiraan.

6. Untuk Metrik, pilih Pasangan metrik kustom.
 - a. Untuk metrik Muat, pilih CloudWatch Metrik khusus untuk menggunakan metrik kustom. Buat payload JSON yang berisi definisi metrik beban untuk kebijakan dan tempelkan ke kotak editor JSON, menggantikan apa yang sudah ada di dalam kotak.
 - b. Untuk metrik Penskalaan, pilih CloudWatch Metrik khusus untuk menggunakan metrik khusus. Buat payload JSON yang berisi definisi metrik penskalaan untuk kebijakan dan tempelkan ke kotak editor JSON, menggantikan apa yang sudah ada di dalam kotak.
 - c. (Opsional) Untuk menambahkan metrik kapasitas khusus, pilih kotak centang untuk Tambahkan metrik kapasitas khusus. Buat payload JSON yang berisi definisi metrik kapasitas untuk kebijakan dan tempelkan ke kotak editor JSON, menggantikan apa yang sudah ada di dalam kotak.

Anda hanya perlu mengaktifkan opsi ini untuk membuat deret waktu baru untuk kapasitas jika data metrik kapasitas Anda mencakup beberapa grup Auto Scaling. Dalam hal ini, Anda harus menggunakan matematika metrik untuk menggabungkan data ke dalam satu deret waktu.

7. Untuk pemanfaatan Target, masukkan nilai target yang harus dipertahankan oleh Auto Scaling Amazon EC2. Amazon EC2 Auto Scaling meningkatkan kapasitas Anda hingga penggunaan rata-rata sesuai dengan target penggunaan, atau hingga mencapai jumlah maksimum instans yang Anda tentukan.
8. (Opsional) Untuk instans Pra-peluncuran, pilih seberapa jauh sebelumnya Anda ingin instans diluncurkan sebelum perkiraan panggilan agar beban meningkat.
9. (Opsional) Untuk perilaku kapasitas Maks, pilih apakah akan membiarkan Auto Scaling Amazon EC2 keluar lebih tinggi dari kapasitas maksimum grup jika kapasitas yang diprediksi melebihi maksimum yang ditentukan. Mengaktifkan pengaturan ini memungkinkan skala keluar terjadi selama periode ketika lalu lintas Anda diperkirakan berada pada titik tertinggi.
10. (Opsional) Untuk kapasitas maksimum Buffer di atas kapasitas yang diperkirakan, pilih berapa banyak kapasitas tambahan yang akan digunakan ketika kapasitas yang diprediksi mendekati atau melebihi kapasitas maksimum. Nilai ditentukan sebagai persentase relatif terhadap kapasitas yang diprediksi. Misalnya, jika buffer adalah 10, ini berarti buffer 10 persen. Oleh

karena itu, jika kapasitas yang diprediksi adalah 50 dan kapasitas maksimum adalah 40, kapasitas maksimum efektif adalah 55.

Jika disetel ke 0, Auto Scaling Amazon EC2 mungkin menskalakan kapasitas lebih tinggi dari kapasitas maksimum hingga sama tetapi tidak melebihi kapasitas yang diprediksi.

11. Pilih Buat kebijakan penskalaan prediktif.

Buat kebijakan penskalaan prediktif ()AWS CLI

Gunakan AWS CLI sebagai berikut untuk mengonfigurasi kebijakan penskalaan prediktif untuk grup Auto Scaling Anda. Ganti setiap *placeholder masukan pengguna* dengan informasi Anda sendiri.

Untuk informasi selengkapnya tentang CloudWatch metrik yang dapat Anda tentukan, lihat [PredictiveScalingMetricSpecification](#) di Referensi API Auto Scaling Amazon EC2.

Contoh 1: Kebijakan penskalaan prediktif yang membuat prakiraan tetapi tidak menskalakan

Contoh kebijakan berikut menunjukkan konfigurasi kebijakan lengkap yang menggunakan metrik pemanfaatan CPU untuk penskalaan prediktif dengan pemanfaatan target. `40 ForecastOnly` mode digunakan secara default, kecuali Anda secara eksplisit menentukan mode mana yang akan digunakan. Simpan konfigurasi ini dalam file bernama `config.json`.

```
{
  "MetricSpecifications": [
    {
      "TargetValue": 40,
      "PredefinedMetricPairSpecification": {
        "PredefinedMetricType": "ASGCPUtilization"
      }
    }
  ]
}
```

Untuk membuat kebijakan dari baris perintah, jalankan [put-scaling-policy](#) perintah dengan file konfigurasi yang ditentukan, seperti yang ditunjukkan dalam contoh berikut.

```
aws autoscaling put-scaling-policy --policy-name cpu40-predictive-scaling-policy \
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \
```

```
--predictive-scaling-configuration file://config.json
```

Jika berhasil, perintah ini mengembalikan Amazon Resource Name (ARN) kebijakan.

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/cpu40-predictive-scaling-
policy",
  "Alarms": []
}
```

Contoh 2: Kebijakan penskalaan prediktif yang memperkirakan dan menskalakan

Untuk kebijakan yang memungkinkan Amazon EC2 Auto Scaling untuk memperkirakan dan menskalakan, tambahkan Mode properti dengan nilai sebesar. `ForecastAndScale` Contoh berikut menunjukkan konfigurasi kebijakan yang menggunakan metrik jumlah permintaan Application Load Balancer. Pemanfaatan target adalah `1000`, dan penskalaan prediktif diatur ke mode. `ForecastAndScale`

```
{
  "MetricSpecifications": [
    {
      "TargetValue": 1000,
      "PredefinedMetricPairSpecification": {
        "PredefinedMetricType": "ALBRequestCount",
        "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-
target-group/943f017f100becff"
      }
    }
  ],
  "Mode": "ForecastAndScale"
}
```

Untuk membuat kebijakan ini, jalankan [put-scaling-policy](#) perintah dengan file konfigurasi yang ditentukan, seperti yang ditunjukkan dalam contoh berikut.

```
aws autoscaling put-scaling-policy --policy-name alb1000-predictive-scaling-policy \
--auto-scaling-group-name my-asg --policy-type PredictiveScaling \
--predictive-scaling-configuration file://config.json
```

Jika berhasil, perintah ini mengembalikan Amazon Resource Name (ARN) kebijakan.

```
{
  "PolicyARN": "arn:aws:autoscaling:region:account-
id:scalingPolicy:19556d63-7914-4997-8c81-d27ca5241386:autoScalingGroupName/my-
asg:policyName/alb1000-predictive-scaling-policy",
  "Alarms": []
}
```

Contoh 3: Kebijakan penskalaan prediktif yang dapat menskalakan lebih tinggi dari kapasitas maksimum

Contoh berikut menunjukkan cara membuat kebijakan yang dapat menskalakan lebih tinggi dari batas ukuran maksimum grup saat Anda membutuhkannya untuk menangani beban yang lebih tinggi dari normal. Secara default, Auto Scaling Amazon EC2 tidak menskalakan kapasitas EC2 Anda lebih tinggi dari kapasitas maksimum yang ditentukan. Namun, mungkin bermanfaat untuk membiarkannya berskala lebih tinggi dengan kapasitas yang sedikit lebih besar untuk menghindari masalah kinerja atau ketersediaan.

Untuk menyediakan ruang bagi Auto Scaling Amazon EC2 untuk menyediakan kapasitas tambahan saat kapasitas diprediksi berada pada atau sangat dekat dengan ukuran maksimum grup Anda, tentukan properti `MaxCapacityBuffer` dan properti, seperti `MaxCapacityBreachBehavior` yang ditunjukkan pada contoh berikut. Anda harus menentukan `MaxCapacityBreachBehavior` dengan nilai `IncreaseMaxCapacity`. Jumlah maksimum instance yang dapat dimiliki grup Anda tergantung pada nilainya. `MaxCapacityBuffer`

```
{
  "MetricSpecifications": [
    {
      "TargetValue": 70,
      "PredefinedMetricPairSpecification": {
        "PredefinedMetricType": "ASGCPUUtilization"
      }
    }
  ],
  "MaxCapacityBreachBehavior": "IncreaseMaxCapacity",
  "MaxCapacityBuffer": 10
}
```

Dalam contoh ini, kebijakan dikonfigurasi untuk menggunakan buffer ("`MaxCapacityBuffer`": 10) 10 persen, jadi jika kapasitas yang diprediksi adalah 50 dan kapasitas maksimum adalah 40, maka kapasitas maksimum efektif adalah 55. Kebijakan yang dapat menskalakan kapasitas lebih

tinggi dari kapasitas maksimum untuk sama tetapi tidak melebihi kapasitas yang diprediksi akan memiliki buffer 0 ("MaxCapacityBuffer": 0).

Untuk membuat kebijakan ini, jalankan [put-scaling-policy](#) perintah dengan file konfigurasi yang ditentukan, seperti yang ditunjukkan dalam contoh berikut.

```
aws autoscaling put-scaling-policy --policy-name cpu70-predictive-scaling-policy \  
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \  
  --predictive-scaling-configuration file://config.json
```

Jika berhasil, perintah ini mengembalikan Amazon Resource Name (ARN) kebijakan.

```
{  
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:d02ef525-8651-4314-  
bf14-888331ebd04f:autoScalingGroupName/my-asg:policyName/cpu70-predictive-scaling-  
policy",  
  "Alarms": []  
}
```

Evaluasi kebijakan penskalaan prediktif Anda

Sebelum Anda menggunakan kebijakan penskalaan prediktif untuk menskalakan grup Auto Scaling, tinjau rekomendasi dan data lain untuk kebijakan Anda di konsol Auto Scaling Amazon EC2. Ini penting karena Anda tidak ingin kebijakan penskalaan prediktif untuk menskalakan kapasitas aktual Anda sampai Anda tahu bahwa prediksinya akurat.

Jika grup Auto Scaling baru, berikan Amazon EC2 Auto Scaling 24 jam untuk membuat perkiraan pertama.

Saat Amazon EC2 Auto Scaling membuat perkiraan, ia menggunakan data historis. Jika grup Auto Scaling Anda belum memiliki banyak data historis terbaru, Auto Scaling Amazon EC2 mungkin untuk sementara mengisi ulang perkiraan dengan agregat yang dibuat dari agregat historis yang tersedia saat ini. Prakiraan diisi kembali hingga dua minggu sebelum tanggal pembuatan kebijakan.

Daftar Isi

- [Lihat rekomendasi penskalaan prediktif Anda](#)
- [Tinjau grafik pemantauan penskalaan prediktif](#)
- [Pantau metrik penskalaan prediktif dengan CloudWatch](#)

Lihat rekomendasi penskalaan prediktif Anda

Untuk analisis yang efektif, Auto Scaling Amazon EC2 harus memiliki setidaknya dua kebijakan penskalaan prediktif untuk dibandingkan. (Namun, Anda masih dapat meninjau temuan untuk satu kebijakan.) Saat membuat beberapa kebijakan, Anda dapat mengevaluasi kebijakan yang menggunakan satu metrik terhadap kebijakan yang menggunakan metrik berbeda. Anda juga dapat mengevaluasi dampak dari nilai target dan kombinasi metrik yang berbeda. Setelah kebijakan penskalaan prediktif dibuat, Auto Scaling Amazon EC2 segera mulai mengevaluasi kebijakan mana yang akan melakukan pekerjaan yang lebih baik dalam menskalakan grup Anda.

Untuk melihat rekomendasi Anda di konsol Auto Scaling Amazon EC2

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Penskalaan otomatis, di bawah Kebijakan penskalaan prediktif, Anda dapat melihat detail tentang kebijakan beserta rekomendasi kami. Rekomendasi memberi tahu Anda apakah kebijakan penskalaan prediktif melakukan pekerjaan yang lebih baik daripada tidak menggunakannya.

Jika Anda tidak yakin apakah kebijakan penskalaan prediktif sesuai untuk grup Anda, tinjau kolom Dampak ketersediaan dan dampak Biaya untuk memilih kebijakan yang tepat. Informasi untuk setiap kolom memberi tahu Anda apa dampak kebijakan tersebut.

- Dampak ketersediaan: Menjelaskan apakah kebijakan akan menghindari dampak negatif terhadap ketersediaan dengan menyediakan instans yang cukup untuk menangani beban kerja, dibandingkan dengan tidak menggunakan kebijakan.
- Dampak biaya: Menjelaskan apakah kebijakan akan menghindari dampak negatif pada biaya Anda dengan tidak menyediakan contoh yang berlebihan, dibandingkan dengan tidak menggunakan kebijakan. Dengan terlalu banyak penyediaan, instans Anda kurang dimanfaatkan atau tidak digunakan, yang hanya menambah dampak biaya.

Jika Anda memiliki beberapa kebijakan, maka tag prediksi terbaik akan ditampilkan di samping nama kebijakan yang memberikan manfaat ketersediaan terbanyak dengan biaya lebih rendah. Lebih banyak bobot diberikan untuk dampak ketersediaan.

4. (Opsional) Untuk memilih periode waktu yang diinginkan untuk hasil rekomendasi, pilih nilai yang Anda inginkan dari dropdown periode Evaluasi: 2 hari, 1 minggu, 2 minggu, 4 minggu, 6 minggu, atau 8 minggu. Secara default, periode evaluasi adalah dua minggu terakhir. Periode evaluasi yang lebih lama memberikan lebih banyak poin data ke hasil rekomendasi. Namun, menambahkan lebih banyak titik data mungkin tidak meningkatkan hasil jika pola beban Anda telah berubah, seperti setelah periode permintaan yang luar biasa. Dalam hal ini, Anda bisa mendapatkan rekomendasi yang lebih fokus dengan melihat data yang lebih baru.

Note

Rekomendasi dibuat hanya untuk kebijakan yang hanya dalam mode Forecast. Fitur rekomendasi bekerja lebih baik ketika kebijakan berada dalam mode Forecast only selama periode evaluasi. Jika Anda memulai kebijakan dalam mode Forecast dan skala dan beralih ke mode Forecast only nanti, temuan untuk kebijakan tersebut cenderung bias. Ini karena kebijakan telah berkontribusi terhadap kapasitas aktual.

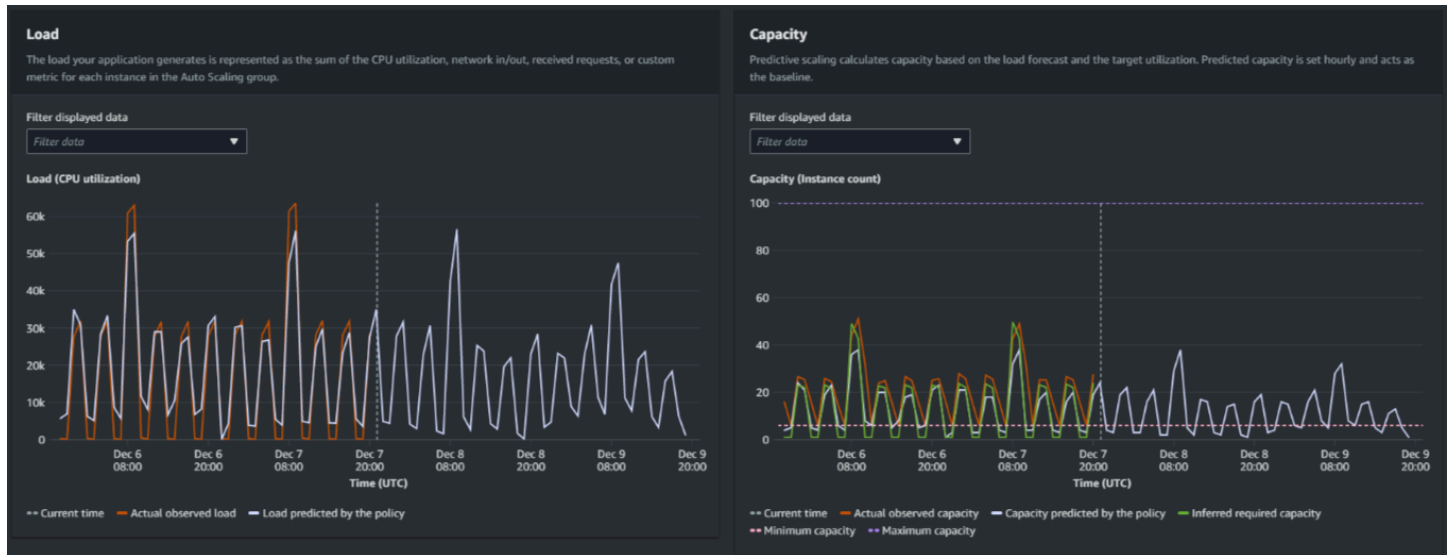
Tinjau grafik pemantauan penskalaan prediktif

Di konsol Auto Scaling Amazon EC2, Anda dapat meninjau perkiraan hari, minggu, atau bulan sebelumnya untuk memvisualisasikan seberapa baik kinerja kebijakan dari waktu ke waktu. Anda juga dapat menggunakan informasi ini untuk mengevaluasi keakuratan prediksi saat memutuskan apakah akan membiarkan kebijakan menskalakan kapasitas aktual Anda.

Untuk meninjau grafik pemantauan penskalaan prediktif di konsol Auto Scaling Amazon EC2

1. Pilih kebijakan dari daftar Kebijakan penskalaan prediktif.
2. Di bagian Pemantauan, Anda dapat melihat perkiraan masa lalu dan masa depan kebijakan Anda untuk beban dan kapasitas terhadap nilai aktual. Grafik beban menunjukkan perkiraan beban dan nilai aktual untuk metrik beban yang Anda pilih. Grafik Kapasitas menunjukkan jumlah contoh yang diprediksi oleh kebijakan. Ini juga mencakup jumlah sebenarnya dari instans yang diluncurkan. Garis vertikal memisahkan nilai historis dari perkiraan masa depan. Grafik ini tersedia segera setelah kebijakan dibuat.
3. (Opsional) Untuk mengubah jumlah data historis yang ditampilkan dalam bagan, pilih nilai yang Anda inginkan dari dropdown Periode evaluasi di bagian atas halaman. Periode evaluasi tidak mengubah data di halaman ini dengan cara apa pun. Itu hanya mengubah jumlah data historis yang ditampilkan.

Gambar berikut menunjukkan grafik Beban dan Kapasitas saat prakiraan telah diterapkan beberapa kali. Prakiraan penskalaan prediktif dimuat berdasarkan data beban historis Anda. Beban yang dihasilkan aplikasi Anda direpresentasikan sebagai jumlah pemanfaatan CPU, masuk/keluar jaringan, permintaan yang diterima, atau metrik kustom untuk setiap instance dalam grup Auto Scaling. Penskalaan prediktif menghitung kebutuhan kapasitas masa depan berdasarkan perkiraan beban dan pemanfaatan target yang ingin Anda capai untuk metrik penskalaan.



Bandingkan data dalam grafik Load

Setiap garis horizontal mewakili serangkaian titik data berbeda yang dilaporkan dalam interval satu jam:

1. Beban yang diamati aktual menggunakan statistik SUM untuk metrik beban yang Anda pilih untuk menunjukkan total beban per jam di masa lalu.
2. Beban yang diprediksi oleh kebijakan menunjukkan prediksi beban per jam. Prediksi ini didasarkan pada dua minggu sebelumnya dari pengamatan beban aktual.

Bandingkan data dalam grafik Kapasitas

Setiap garis horizontal mewakili serangkaian titik data berbeda yang dilaporkan dalam interval satu jam:

1. Kapasitas pengamatan aktual menunjukkan kapasitas aktual grup Auto Scaling Anda di masa lalu, yang bergantung pada kebijakan penskalaan Anda yang lain dan ukuran grup minimum yang berlaku untuk periode waktu yang dipilih.

2. Kapasitas yang diprediksi oleh kebijakan menunjukkan kapasitas dasar yang dapat Anda harapkan pada awal setiap jam ketika kebijakan dalam mode Forecast dan skala.
3. Kapasitas yang diperlukan yang disimpulkan menunjukkan kapasitas ideal untuk mempertahankan metrik penskalaan pada nilai target yang Anda pilih.
4. Kapasitas minimum menunjukkan kapasitas minimum grup Auto Scaling.
5. Kapasitas maksimum menunjukkan kapasitas maksimum grup Auto Scaling.

Untuk tujuan menghitung kapasitas yang diperlukan yang disimpulkan, kita mulai dengan mengasumsikan bahwa setiap instance sama-sama digunakan pada nilai target yang ditentukan. Dalam praktiknya, contoh tidak digunakan secara merata. Dengan mengasumsikan bahwa pemanfaatan tersebar secara seragam antar contoh, bagaimanapun, kita dapat membuat perkiraan kemungkinan jumlah kapasitas yang dibutuhkan. Persyaratan kapasitas kemudian dihitung berbanding terbalik dengan metrik penskalaan yang Anda gunakan untuk kebijakan penskalaan prediktif Anda. Dengan kata lain, ketika kapasitas meningkat, metrik penskalaan menurun pada tingkat yang sama. Misalnya, jika kapasitas berlipat ganda, metrik penskalaan harus berkurang setengahnya.

Rumus untuk kapasitas yang dibutuhkan yang disimpulkan:

$$\text{sum of } (\text{actualCapacityUnits} * \text{scalingMetricValue}) / (\text{targetUtilization})$$

Sebagai contoh, kita mengambil `actualCapacityUnits` (10) dan `scalingMetricValue` (30) untuk satu jam tertentu. Kami kemudian mengambil `targetUtilization` yang Anda tentukan dalam kebijakan penskalaan prediktif Anda (60) dan menghitung kapasitas yang diperlukan yang disimpulkan untuk jam yang sama. Ini mengembalikan nilai 5. Ini berarti bahwa lima adalah jumlah kapasitas yang disimpulkan yang diperlukan untuk mempertahankan kapasitas dalam proporsi terbalik langsung dengan nilai target metrik penskalaan.

Note

Berbagai tuas tersedia bagi Anda untuk menyesuaikan dan meningkatkan penghematan biaya dan ketersediaan aplikasi Anda.

- Anda menggunakan penskalaan prediktif untuk kapasitas dasar dan penskalaan dinamis untuk menangani kapasitas tambahan. Penskalaan dinamis bekerja secara independen dari penskalaan prediktif, penskalaan masuk dan keluar berdasarkan pemanfaatan saat ini. Pertama, Auto Scaling Amazon EC2 menghitung jumlah instans yang disarankan

untuk setiap kebijakan penskalaan dinamis. Kemudian, skala berdasarkan kebijakan yang menyediakan jumlah instans terbesar.

- Agar penskalaan masuk terjadi saat beban berkurang, grup Auto Scaling Anda harus selalu memiliki setidaknya satu kebijakan penskalaan dinamis dengan bagian penskalaan diaktifkan.
- Anda dapat meningkatkan kinerja penskalaan dengan memastikan bahwa kapasitas minimum dan maksimum Anda tidak terlalu membatasi. Kebijakan dengan jumlah instans yang direkomendasikan yang tidak termasuk dalam rentang kapasitas minimum dan maksimum akan dicegah dari penskalaan masuk dan keluar.

Pantau metrik penskalaan prediktif dengan CloudWatch

Bergantung pada kebutuhan Anda, Anda mungkin lebih suka mengakses data pemantauan untuk penskalaan prediktif dari Amazon CloudWatch daripada konsol Auto Scaling Amazon EC2. Setelah Anda membuat kebijakan penskalaan prediktif, kebijakan tersebut mengumpulkan data yang digunakan untuk memperkirakan beban dan kapasitas masa depan Anda. Setelah data ini dikumpulkan, secara otomatis disimpan CloudWatch secara berkala. Kemudian, Anda dapat menggunakannya CloudWatch untuk memvisualisasikan seberapa baik kinerja kebijakan dari waktu ke waktu. Anda juga dapat membuat CloudWatch alarm untuk memberi tahu Anda saat indikator kinerja berubah melampaui batas yang Anda tentukan. CloudWatch

Topik

- [Visualisasikan data perkiraan historis](#)
- [Buat metrik akurasi menggunakan matematika metrik](#)

Visualisasikan data perkiraan historis

Anda dapat melihat data perkiraan beban dan kapasitas untuk kebijakan penskalaan prediktif di CloudWatch. Ini dapat berguna saat memvisualisasikan prakiraan terhadap CloudWatch metrik lain dalam satu grafik. Ini juga dapat membantu saat melihat rentang waktu yang lebih luas sehingga Anda dapat melihat tren dari waktu ke waktu. Anda dapat mengakses metrik historis hingga 15 bulan untuk mendapatkan perspektif yang lebih baik tentang kinerja kebijakan Anda.

Untuk informasi selengkapnya, lihat [Metrik dan dimensi penskalaan prediktif](#).

Untuk melihat data perkiraan historis menggunakan CloudWatch konsol

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Di panel navigasi, pilih Metrik, lalu Semua metrik.
3. Pilih namespace metrik Auto Scaling.
4. Pilih salah satu opsi berikut untuk melihat perkiraan beban atau metrik perkiraan kapasitas:
 - Prakiraan Beban Penskalaan Prediktif
 - Prakiraan Kapasitas Penskalaan Prediktif
5. Di kolom pencarian, masukkan nama kebijakan penskalaan prediktif atau nama grup Auto Scaling, lalu tekan Enter untuk memfilter hasilnya.
6. Untuk membuat grafik sebuah metrik, pilih kotak centang di sebelah metrik. Untuk mengubah nama grafik, pilih ikon pensil. Untuk mengubah rentang waktu, pilih salah satu nilai yang telah ditentukan sebelumnya atau pilih kustom. Untuk informasi selengkapnya, lihat [Membuat grafik metrik](#) di Panduan CloudWatch Pengguna Amazon.
7. Untuk mengubah statistik, pilih tab Metrik bergrafik. Pilih judul kolom atau nilai individual, lalu pilih statistik yang berbeda. Meskipun Anda dapat memilih statistik apa pun untuk setiap metrik, tidak semua statistik berguna PredictiveScalingLoadForecast dan PredictiveScalingCapacityForecast metrik. Misalnya, statistik Rata-rata, Minimum, dan Maksimum berguna, tetapi statistik Jumlah tidak.
8. Untuk menambahkan metrik lain ke grafik, di bawah Browse, pilih Semua, temukan metrik tertentu, lalu pilih kotak centang di sebelahnya. Anda dapat menambahkan hingga 10 metrik.

Misalnya, untuk menambahkan nilai aktual untuk pemanfaatan CPU ke grafik, pilih namespace EC2 dan kemudian pilih By Auto Scaling Group. Kemudian, pilih kotak centang untuk metrik CPUUtilization dan grup Auto Scaling tertentu.
9. (Opsional) Untuk menambahkan grafik ke CloudWatch dasbor, pilih Tindakan, Tambahkan ke dasbor.

Buat metrik akurasi menggunakan matematika metrik

Dengan matematika metrik, Anda dapat menanyakan beberapa CloudWatch metrik dan menggunakan ekspresi matematika untuk membuat deret waktu baru berdasarkan metrik ini. Anda dapat memvisualisasikan deret waktu yang dihasilkan di CloudWatch konsol dan menambahkannya ke dasbor. Untuk informasi selengkapnya tentang matematika [metrik](#), lihat [Menggunakan matematika metrik](#) di Panduan CloudWatch Pengguna Amazon.

Dengan menggunakan matematika metrik, Anda dapat membuat grafik data yang dihasilkan Amazon EC2 Auto Scaling untuk penskalaan prediktif dengan berbagai cara. Ini membantu Anda memantau kinerja kebijakan dari waktu ke waktu, dan membantu Anda memahami apakah kombinasi metrik Anda dapat ditingkatkan.

Misalnya, Anda dapat menggunakan ekspresi matematika metrik untuk memantau [kesalahan persentase absolut rata-rata](#) (MAPE). Metrik MAPE membantu memantau perbedaan antara nilai yang diperkirakan dan nilai aktual yang diamati selama jendela perkiraan tertentu. Perubahan nilai MAPE dapat menunjukkan apakah kinerja kebijakan menurun seiring waktu karena sifat aplikasi Anda berubah. Peningkatan MAPE menandakan kesenjangan yang lebih luas antara nilai yang diperkirakan dan nilai aktual.

Contoh: Ekspresi matematika metrik

Untuk memulai dengan jenis grafik ini, Anda dapat membuat ekspresi matematika metrik seperti yang ditunjukkan pada contoh berikut.

```
{
  "MetricDataQueries": [
    {
      "Expression": "TIME_SERIES(AVG(ABS(m1-m2)/m1))",
      "Id": "e1",
      "Period": 3600,
      "Label": "MeanAbsolutePercentageError",
      "ReturnData": true
    },
    {
      "Id": "m1",
      "Label": "ActualLoadValues",
      "MetricStat": {
        "Metric": {
          "Namespace": "AWS/EC2",
          "MetricName": "CPUUtilization",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg"
            }
          ]
        }
      }
    },
    {
      "Period": 3600,
      "Stat": "Sum"
    }
  ]
}
```

```

    },
    "ReturnData": false
  },
  {
    "Id": "m2",
    "Label": "ForecastedLoadValues",
    "MetricStat": {
      "Metric": {
        "Namespace": "AWS/AutoScaling",
        "MetricName": "PredictiveScalingLoadForecast",
        "Dimensions": [
          {
            "Name": "AutoScalingGroupName",
            "Value": "my-asg"
          },
          {
            "Name": "PolicyName",
            "Value": "my-predictive-scaling-policy"
          },
          {
            "Name": "PairIndex",
            "Value": "0"
          }
        ]
      },
      "Period": 3600,
      "Stat": "Average"
    },
    "ReturnData": false
  }
]
}

```

Alih-alih metrik tunggal, ada array struktur kueri data metrik untuk `MetricDataQueries`. Setiap item `MetricDataQueries` mendapat metrik atau melakukan ekspresi matematika. Item pertama, `e1`, adalah ekspresi matematika. Ekspresi yang ditunjuk menetapkan `ReturnData` parameter ke `true`, yang pada akhirnya menghasilkan deret waktu tunggal. Untuk semua metrik lainnya, `ReturnData` nilainya adalah `false`.

Dalam contoh, ekspresi yang ditunjuk menggunakan nilai aktual dan yang diperkirakan sebagai input dan mengembalikan metrik baru (MAPE). `m1` adalah CloudWatch metrik yang berisi nilai beban aktual (dengan asumsi pemanfaatan CPU adalah metrik beban yang awalnya ditentukan untuk kebijakan

bernamamy-predictive-scaling-policy). m2 adalah CloudWatch metrik yang berisi nilai beban yang diperkirakan. Sintaks matematika untuk metrik MAPE adalah sebagai berikut:

Rata-rata ($\text{abs}((\text{Aktual} - \text{Forecast})/(\text{Aktual}))$)

Visualisasikan metrik akurasi Anda dan atur alarm

Untuk memvisualisasikan data metrik akurasi, pilih tab Metrik di konsol. CloudWatch Anda dapat membuat grafik data dari sana. Untuk informasi selengkapnya, lihat [Menambahkan ekspresi matematika ke CloudWatch grafik](#) di Panduan CloudWatch Pengguna Amazon.

Anda juga dapat mengatur alarm pada metrik yang Anda pantau dari bagian Metrik. Saat berada di tab Graphed metrics, pilih ikon Create alarm di bawah kolom Actions. Ikon Create alarm direpresentasikan sebagai bel kecil. Untuk informasi selengkapnya dan opsi notifikasi, lihat [Membuat CloudWatch alarm berdasarkan ekspresi matematika metrik](#) dan [Memberi tahu pengguna tentang perubahan alarm](#) di Panduan CloudWatch Pengguna Amazon.

Atau, Anda dapat menggunakan [GetMetricData](#) dan [PutMetricAlarm](#) melakukan perhitungan menggunakan matematika metrik dan membuat alarm berdasarkan output.

Ganti nilai perkiraan menggunakan tindakan terjadwal

Terkadang, Anda mungkin memiliki informasi tambahan tentang persyaratan aplikasi future Anda yang tidak dapat diperhitungkan oleh perhitungan perkiraan. Misalnya, perhitungan perkiraan mungkin meremehkan kapasitas yang dibutuhkan untuk acara pemasaran yang akan datang. Anda dapat menggunakan tindakan terjadwal untuk mengganti perkiraan sementara selama periode waktu mendatang. Tindakan terjadwal dapat berjalan secara berulang, atau pada tanggal dan waktu tertentu ketika ada fluktuasi permintaan satu kali.

Misalnya, Anda dapat membuat tindakan terjadwal dengan kapasitas minimum yang lebih tinggi dari yang diperkirakan. Saat runtime, Amazon EC2 Auto Scaling memperbarui kapasitas minimum grup Auto Scaling Anda. Karena penskalaan prediktif mengoptimalkan kapasitas, tindakan terjadwal dengan kapasitas minimum yang lebih tinggi dari nilai perkiraan dihormati. Ini mencegah kapasitas menjadi kurang dari yang diharapkan. Untuk berhenti mengesampingkan perkiraan, gunakan tindakan terjadwal kedua untuk mengembalikan kapasitas minimum ke pengaturan aslinya.

Prosedur berikut menguraikan langkah-langkah untuk mengesampingkan perkiraan selama periode waktu mendatang.

Daftar Isi

- [Langkah 1: \(Opsional\) Analisis data deret waktu](#)
- [Langkah 2: Buat dua tindakan terjadwal](#)

Langkah 1: (Opsional) Analisis data deret waktu

Mulailah dengan menganalisis data deret waktu perkiraan. Ini adalah langkah opsional, tetapi akan sangat membantu jika Anda ingin memahami detail perkiraan.

1. Ambil ramalan

Setelah perkiraan dibuat, Anda dapat menanyakan periode waktu tertentu dalam perkiraan. Tujuan dari kueri ini adalah untuk mendapatkan tampilan lengkap dari data deret waktu untuk periode waktu tertentu.

Kueri Anda dapat mencakup hingga dua hari data perkiraan masa depan. Jika Anda telah menggunakan penskalaan prediktif untuk sementara waktu, Anda juga dapat mengakses data perkiraan sebelumnya. Namun, durasi waktu maksimum antara waktu mulai dan akhir adalah 30 hari.

Untuk mendapatkan perkiraan menggunakan [get-predictive-scaling-forecast](#) AWS CLI perintah, berikan parameter berikut dalam perintah:

- Masukkan nama grup Auto Scaling di parameter. `--auto-scaling-group-name`
- Masukkan nama kebijakan dalam `--policy-name` parameter.
- Masukkan waktu mulai dalam `--start-time` parameter untuk mengembalikan hanya data perkiraan setelah atau pada waktu yang ditentukan.
- Masukkan waktu akhir dalam `--end-time` parameter untuk mengembalikan hanya data perkiraan sebelum waktu yang ditentukan.

```
aws autoscaling get-predictive-scaling-forecast --auto-scaling-group-name my-asg \  
--policy-name cpu40-predictive-scaling-policy \  
--start-time "2021-05-19T17:00:00Z" \  
--end-time "2021-05-19T23:00:00Z"
```

Jika berhasil, perintah mengembalikan data yang mirip dengan contoh berikut.

```
{
```

```
"LoadForecast": [
  {
    "Timestamps": [
      "2021-05-19T17:00:00+00:00",
      "2021-05-19T18:00:00+00:00",
      "2021-05-19T19:00:00+00:00",
      "2021-05-19T20:00:00+00:00",
      "2021-05-19T21:00:00+00:00",
      "2021-05-19T22:00:00+00:00",
      "2021-05-19T23:00:00+00:00"
    ],
    "Values": [
      153.0655799339254,
      128.8288551285919,
      107.1179447150675,
      197.3601844551528,
      626.4039934516954,
      596.9441277518481,
      677.9675713779869
    ],
    "MetricSpecification": {
      "TargetValue": 40.0,
      "PredefinedMetricPairSpecification": {
        "PredefinedMetricType": "ASGCPUUtilization"
      }
    }
  }
],
"CapacityForecast": {
  "Timestamps": [
    "2021-05-19T17:00:00+00:00",
    "2021-05-19T18:00:00+00:00",
    "2021-05-19T19:00:00+00:00",
    "2021-05-19T20:00:00+00:00",
    "2021-05-19T21:00:00+00:00",
    "2021-05-19T22:00:00+00:00",
    "2021-05-19T23:00:00+00:00"
  ],
  "Values": [
    2.0,
    2.0,
    2.0,
    2.0,
    4.0,
```



```
        4.0,  
        4.0  
    ]  
  },  
  "UpdateTime": "2021-05-19T01:52:50.118000+00:00"  
}
```

Tanggapan tersebut mencakup dua prakiraan: `LoadForecast` dan `CapacityForecast`. `LoadForecast` menunjukkan perkiraan beban per jam. `CapacityForecast` menunjukkan nilai perkiraan untuk kapasitas yang dibutuhkan setiap jam untuk menangani beban yang diperkirakan sambil mempertahankan 40,0 (`TargetValue` pemanfaatan CPU rata-rata 40%).

2. Identifikasi periode waktu target

Identifikasi jam atau jam ketika fluktuasi permintaan satu kali harus terjadi. Ingat bahwa tanggal dan waktu yang ditunjukkan dalam perkiraan ada di UTC.

Langkah 2: Buat dua tindakan terjadwal

Selanjutnya, buat dua tindakan terjadwal untuk periode waktu tertentu ketika aplikasi Anda akan memiliki beban yang lebih tinggi dari perkiraan. Misalnya, jika Anda memiliki acara pemasaran yang akan mengarahkan lalu lintas ke situs Anda untuk jangka waktu terbatas, Anda dapat menjadwalkan tindakan satu kali untuk memperbarui kapasitas minimum saat dimulai. Kemudian, jadwalkan tindakan lain untuk mengembalikan kapasitas minimum ke pengaturan asli saat acara berakhir.

Untuk membuat dua tindakan terjadwal untuk acara satu kali (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Auto scaling, di Tindakan terjadwal, pilih Buat tindakan terjadwal.
4. Isi pengaturan tindakan terjadwal berikut:
 - a. Masukkan Nama untuk tindakan yang dijadwalkan.
 - b. Untuk Min, masukkan kapasitas minimum baru untuk grup Auto Scaling Anda. Min harus kurang dari atau sama dengan ukuran maksimum grup. Jika nilai Anda untuk Min lebih besar dari ukuran maksimum grup, Anda harus memperbarui Max.

- c. Untuk Perulangan, pilih Sekali.
 - d. Untuk zona waktu, pilih zona waktu. Jika tidak ada zona waktu yang dipilih, ETC/UTC digunakan secara default.
 - e. Tentukan waktu mulai yang spesifik.
5. Pilih Buat.

Konsol menampilkan tindakan terjadwal untuk grup Auto Scaling.

6. Konfigurasi tindakan terjadwal kedua untuk mengembalikan kapasitas minimum ke pengaturan asli di akhir acara. Penskalaan prediktif dapat menskalakan kapasitas hanya jika nilai yang Anda tetapkan untuk Min lebih rendah dari nilai perkiraan.

Untuk membuat dua tindakan terjadwal untuk acara satu kali ()AWS CLI

Untuk menggunakan AWS CLI untuk membuat tindakan terjadwal, gunakan perintah [put-scheduled-update-group-action](#).

Sebagai contoh, mari kita tentukan jadwal yang mempertahankan kapasitas minimum tiga instance pada 19 Mei pukul 17:00 selama delapan jam. Perintah berikut menunjukkan bagaimana menerapkan skenario ini.

Perintah [put-scheduled-update-group-action](#) pertama menginstruksikan Amazon EC2 Auto Scaling untuk memperbarui kapasitas minimum grup Auto Scaling yang ditentukan pada pukul 17:00 UTC pada 19 Mei 2021.

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-event-start \  
  --auto-scaling-group-name my-asg --start-time "2021-05-19T17:00:00Z" --minimum-  
  capacity 3
```

Perintah kedua menginstruksikan Amazon EC2 Auto Scaling untuk mengatur kapasitas minimum grup menjadi satu pada 1:00 UTC pada 20 Mei 2021.

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-event-end \  
  --auto-scaling-group-name my-asg --start-time "2021-05-20T01:00:00Z" --minimum-  
  capacity 1
```

Setelah Anda menambahkan tindakan terjadwal ini ke grup Auto Scaling, Auto Scaling Amazon EC2 melakukan hal berikut:

- Pada pukul 17:00 UTC pada 19 Mei 2021, aksi terjadwal pertama berjalan. Jika grup saat ini memiliki kurang dari tiga instance, grup tersebut menskalakan menjadi tiga instance. Selama waktu ini dan selama delapan jam ke depan, Auto Scaling Amazon EC2 dapat terus ditingkatkan jika kapasitas yang diprediksi lebih tinggi dari kapasitas aktual atau jika ada kebijakan penskalaan dinamis yang berlaku.
- Pukul 1:00 UTC pada 20 Mei 2021, aksi terjadwal kedua berjalan. Ini mengembalikan kapasitas minimum ke pengaturan aslinya di akhir acara.

Penskalaan berdasarkan jadwal berulang

Untuk mengganti perkiraan untuk periode waktu yang sama setiap minggu, buat dua tindakan terjadwal dan berikan logika waktu dan tanggal menggunakan ekspresi cron.

Format ekspresi cron terdiri dari lima bidang yang dipisahkan oleh spasi: [Minute] [Hour] [Day_of_month] [Month_of_year] [day_of_week]. Bidang dapat berisi nilai apa pun yang diizinkan, termasuk karakter khusus.

Misalnya, ekspresi cron berikut menjalankan aksi setiap hari Selasa pukul 6:30 pagi. Tanda bintang digunakan sebagai wildcard untuk mencocokkan semua nilai untuk bidang.

```
30 6 * * 2
```

Lihat juga

Untuk informasi selengkapnya tentang cara membuat, membuat daftar, mengedit, dan menghapus tindakan terjadwal, lihat [Skala terjadwal untuk Amazon EC2 Auto Scaling](#).

Konfigurasi kebijakan penskalaan prediktif lanjutan menggunakan metrik khusus

Dalam kebijakan penskalaan prediktif, Anda dapat menggunakan metrik yang telah ditentukan atau kustom. Metrik kustom berguna ketika metrik yang telah ditentukan (CPU, I/O jaringan, dan jumlah permintaan Application Load Balancer) tidak cukup menggambarkan pemuatan aplikasi Anda.

Saat membuat kebijakan penskalaan prediktif dengan metrik kustom, Anda dapat menentukan CloudWatch metrik lain yang disediakan oleh AWS, atau Anda dapat menentukan metrik yang

Anda tentukan dan publikasikan sendiri. Anda juga dapat menggunakan matematika metrik untuk menggabungkan dan mengubah metrik yang ada menjadi deret waktu baru yang AWS tidak dilacak secara otomatis. Ketika Anda menggabungkan nilai dalam data Anda, misalnya, dengan menghitung jumlah atau rata-rata baru, itu disebut agregasi. Data yang dihasilkan disebut agregat.

Bagian berikut berisi praktik terbaik dan contoh bagaimana membangun struktur JSON untuk kebijakan tersebut.

Daftar Isi

- [Praktik terbaik](#)
- [Prasyarat](#)
- [Membangun JSON untuk metrik khusus](#)
- [Pertimbangan dan pemecahan masalah](#)
- [Batasan](#)

Praktik terbaik

Praktik terbaik berikut dapat membantu Anda menggunakan metrik kustom secara lebih efektif:

- Untuk spesifikasi metrik beban, metrik yang paling berguna adalah metrik yang mewakili beban pada grup Auto Scaling secara keseluruhan, terlepas dari kapasitas grup.
- Untuk spesifikasi metrik penskalaan, metrik yang paling berguna untuk diskalakan adalah throughput rata-rata atau pemanfaatan per metrik instance.
- Metrik penskalaan harus berbanding terbalik dengan kapasitas. Artinya, jika jumlah instance dalam grup Auto Scaling meningkat, metrik penskalaan akan berkurang kira-kira proporsi yang sama. Untuk memastikan bahwa penskalaan prediktif berperilaku seperti yang diharapkan, metrik beban dan metrik penskalaan juga harus berkorelasi kuat satu sama lain.
- Pemanfaatan target harus sesuai dengan jenis metrik penskalaan. Untuk konfigurasi kebijakan yang menggunakan pemanfaatan CPU, ini adalah persentase target. Untuk konfigurasi kebijakan yang menggunakan throughput, seperti jumlah permintaan atau pesan, ini adalah jumlah target permintaan atau pesan per instance selama interval satu menit.
- Jika rekomendasi ini tidak diikuti, nilai future yang diperkirakan dari deret waktu mungkin akan salah. Untuk memvalidasi bahwa data sudah benar, Anda dapat melihat nilai yang diperkirakan di konsol Auto Scaling Amazon EC2. Atau, setelah Anda membuat kebijakan penskalaan prediktif, periksa `LoadForecast` dan `CapacityForecast` objek yang ditampilkan oleh panggilan ke API. [GetPredictiveScalingForecast](#)

- Kami sangat menyarankan agar Anda mengonfigurasi penskalaan prediktif dalam mode hanya perkiraan sehingga Anda dapat mengevaluasi perkiraan sebelum penskalaan prediktif mulai secara aktif menskalakan kapasitas.

Prasyarat

Untuk menambahkan metrik khusus ke kebijakan penskalaan prediktif, Anda harus memiliki izin. `cloudwatch:GetMetricData`

Untuk menentukan metrik Anda sendiri, bukan metrik yang AWS disediakan, Anda harus terlebih dahulu mempublikasikan metrik Anda. CloudWatch Untuk informasi selengkapnya, lihat [Menerbitkan metrik kustom](#) di Panduan CloudWatch Pengguna Amazon.

Jika Anda mempublikasikan metrik Anda sendiri, pastikan untuk mempublikasikan titik data pada frekuensi minimum lima menit. Auto Scaling Amazon EC2 mengambil titik data berdasarkan panjang CloudWatch periode yang dibutuhkannya. Misalnya, spesifikasi metrik beban menggunakan metrik per jam untuk mengukur beban pada aplikasi Anda. CloudWatch menggunakan data metrik yang Anda publikasikan untuk memberikan nilai data tunggal untuk periode satu jam dengan menggabungkan semua titik data dengan stempel waktu yang termasuk dalam setiap periode satu jam.

Membangun JSON untuk metrik khusus

Bagian berikut berisi contoh cara mengonfigurasi penskalaan prediktif ke data kueri dari CloudWatch. Ada dua metode berbeda untuk mengonfigurasi opsi ini, dan metode yang Anda pilih memengaruhi format mana yang Anda gunakan untuk membuat JSON untuk kebijakan penskalaan prediktif Anda. Saat Anda menggunakan matematika metrik, format JSON bervariasi lebih lanjut berdasarkan matematika metrik yang dilakukan.

1. Untuk membuat kebijakan yang mendapatkan data langsung dari CloudWatch metrik lain yang disediakan oleh AWS atau metrik yang Anda publikasikan CloudWatch, lihat. [Contoh kebijakan penskalaan prediktif dengan metrik pemuatan dan penskalaan khusus \(AWS CLI\)](#)
2. Untuk membuat kebijakan yang dapat menanyakan beberapa CloudWatch metrik dan menggunakan ekspresi matematika untuk membuat deret waktu baru berdasarkan metrik ini, lihat. [Gunakan ekspresi matematika metrik](#)

Contoh kebijakan penskalaan prediktif dengan metrik pemuatan dan penskalaan khusus ()AWS CLI

Untuk membuat kebijakan penskalaan prediktif dengan metrik pemuatan dan penskalaan khusus dengan AWS CLI, simpan argumen `--predictive-scaling-configuration` dalam file JSON bernama `config.json`

Anda mulai menambahkan metrik khusus dengan mengganti nilai yang dapat diganti dalam contoh berikut dengan metrik dan pemanfaatan target Anda.

```
{
  "MetricSpecifications": [
    {
      "TargetValue": 50,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "scaling_metric",
            "MetricStat": {
              "Metric": {
                "MetricName": "MyUtilizationMetric",
                "Namespace": "MyNameSpace",
                "Dimensions": [
                  {
                    "Name": "MyOptionalMetricDimensionName",
                    "Value": "MyOptionalMetricDimensionValue"
                  }
                ]
              },
              "Stat": "Average"
            }
          }
        ]
      },
      "CustomizedLoadMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "load_metric",
            "MetricStat": {
              "Metric": {
                "MetricName": "MyLoadMetric",
                "Namespace": "MyNameSpace",
                "Dimensions": [
                  {
```

```
        "Name": "MyOptionalMetricDimensionName",
        "Value": "MyOptionalMetricDimensionValue"
      }
    ]
  },
  "Stat": "Sum"
}
]
```

Untuk informasi selengkapnya, lihat [MetricDataQuery](#) di Referensi API Auto Scaling Amazon EC2.

Note

Berikut adalah beberapa sumber daya tambahan yang dapat membantu Anda menemukan nama metrik, ruang nama, dimensi, dan statistik untuk CloudWatch metrik:

- Untuk informasi tentang metrik yang tersedia untuk AWS layanan, lihat [AWS layanan yang memublikasikan CloudWatch metrik](#) di CloudWatch Panduan Pengguna Amazon.
- [Untuk mendapatkan nama metrik, namespace, dan dimensi yang tepat \(jika ada\) untuk CloudWatch metrik dengan metrik AWS CLI, lihat daftar-metrik.](#)

Untuk membuat kebijakan ini, jalankan `put-scaling-policy` perintah menggunakan file JSON sebagai input, seperti yang ditunjukkan dalam contoh berikut.

```
aws autoscaling put-scaling-policy --policy-name my-predictive-scaling-policy \  
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \  
  --predictive-scaling-configuration file://config.json
```

Jika berhasil, perintah ini mengembalikan Amazon Resource Name (ARN) kebijakan.

```
{  
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-  
b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-predictive-scaling-policy",  
  "Alarms": []
```

```
}
```

Gunakan ekspresi matematika metrik

Bagian berikut memberikan informasi dan contoh kebijakan penskalaan prediktif yang menunjukkan bagaimana Anda dapat menggunakan matematika metrik dalam kebijakan Anda.

Daftar Isi

- [Memahami matematika metrik](#)
- [Contoh kebijakan penskalaan prediktif yang menggabungkan metrik menggunakan metrik matematika \(\)AWS CLI](#)
- [Contoh kebijakan penskalaan prediktif untuk digunakan dalam skenario penerapan biru/hijau \(\)AWS CLI](#)

Memahami matematika metrik

Jika yang ingin Anda lakukan hanyalah mengumpulkan data metrik yang ada, matematika CloudWatch metrik menghemat upaya dan biaya penerbitan metrik lain. CloudWatch Anda dapat menggunakan metrik apa pun yang AWS menyediakan, dan Anda juga dapat menggunakan metrik yang Anda tentukan sebagai bagian dari aplikasi Anda. Misalnya, Anda mungkin ingin menghitung backlog antrian Amazon SQS per instance. Anda dapat melakukannya dengan mengambil perkiraan jumlah pesan yang tersedia untuk diambil dari antrian dan membagi nomor tersebut dengan kapasitas berjalan grup Auto Scaling.

Untuk informasi selengkapnya, lihat [Menggunakan matematika metrik](#) di Panduan CloudWatch Pengguna Amazon.

Jika Anda memilih untuk menggunakan ekspresi matematika metrik dalam kebijakan penskalaan prediktif Anda, pertimbangkan poin-poin berikut:

- Operasi matematika metrik menggunakan titik data dari kombinasi unik nama metrik, namespace, dan kunci dimensi/pasangan nilai metrik.
- Anda dapat menggunakan operator aritmatika (+ - */^), fungsi statistik (seperti AVG atau SUM), atau fungsi lain yang mendukung. CloudWatch
- Anda dapat menggunakan metrik dan hasil ekspresi matematika lainnya dalam rumus ekspresi matematika.

- Ekspresi matematika metrik Anda dapat terdiri dari agregasi yang berbeda. Namun, ini adalah praktik terbaik untuk hasil agregasi akhir yang digunakan Average untuk metrik penskalaan dan Sum untuk metrik beban.
- Setiap ekspresi yang digunakan dalam spesifikasi metrik pada akhirnya harus mengembalikan satu deret waktu.

Untuk menggunakan matematika metrik, lakukan hal berikut:

- Pilih satu atau beberapa CloudWatch metrik. Kemudian, buat ekspresi. Untuk informasi selengkapnya, lihat [Menggunakan matematika metrik](#) di Panduan CloudWatch Pengguna Amazon.
- Verifikasi bahwa ekspresi matematika metrik valid dengan menggunakan CloudWatch konsol atau CloudWatch [GetMetricData](#) API.

Contoh kebijakan penskalaan prediktif yang menggabungkan metrik menggunakan metrik matematika ()AWS CLI

Terkadang, alih-alih menentukan metrik secara langsung, Anda mungkin perlu terlebih dahulu memproses datanya dengan cara tertentu. Misalnya, Anda mungkin memiliki aplikasi yang menarik pekerjaan dari antrean Amazon SQS, dan Anda mungkin ingin menggunakan jumlah item dalam antrian sebagai kriteria untuk penskalaan prediktif. Jumlah pesan dalam antrian tidak hanya menentukan jumlah instance yang Anda butuhkan. Oleh karena itu, lebih banyak pekerjaan diperlukan untuk membuat metrik yang dapat digunakan untuk menghitung backlog per instance. Untuk informasi selengkapnya, lihat [Penskalaan berdasarkan Amazon SQS](#).

Berikut ini adalah contoh kebijakan penskalaan prediktif untuk skenario ini. Ini menentukan metrik penskalaan dan pemuatan yang didasarkan pada metrik Amazon SQS `ApproximateNumberOfMessagesVisible`, yang merupakan jumlah pesan yang tersedia untuk diambil dari antrian. Ini juga menggunakan metrik `AutoGroupInServiceInstances` Scaling Amazon EC2 dan ekspresi matematika untuk menghitung backlog per instance untuk metrik penskalaan.

```
aws autoscaling put-scaling-policy --policy-name my-sqs-custom-metrics-policy \  
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \  
  --predictive-scaling-configuration file://config.json  
{  
  "MetricSpecifications": [  
    {  
      "TargetValue": 100,
```

```
"CustomizedScalingMetricSpecification": {
  "MetricDataQueries": [
    {
      "Label": "Get the queue size (the number of messages waiting to be
processed)",
      "Id": "queue_size",
      "MetricStat": {
        "Metric": {
          "MetricName": "ApproximateNumberOfMessagesVisible",
          "Namespace": "AWS/SQS",
          "Dimensions": [
            {
              "Name": "QueueName",
              "Value": "my-queue"
            }
          ]
        },
        "Stat": "Sum"
      },
      "ReturnData": false
    },
    {
      "Label": "Get the group size (the number of running instances)",
      "Id": "running_capacity",
      "MetricStat": {
        "Metric": {
          "MetricName": "GroupInServiceInstances",
          "Namespace": "AWS/AutoScaling",
          "Dimensions": [
            {
              "Name": "AutoScalingGroupName",
              "Value": "my-asg"
            }
          ]
        },
        "Stat": "Sum"
      },
      "ReturnData": false
    },
    {
      "Label": "Calculate the backlog per instance",
      "Id": "scaling_metric",
      "Expression": "queue_size / running_capacity",
      "ReturnData": true
    }
  ]
}
```

```

    }
  ]
},
"CustomizedLoadMetricSpecification": {
  "MetricDataQueries": [
    {
      "Id": "load_metric",
      "MetricStat": {
        "Metric": {
          "MetricName": "ApproximateNumberOfMessagesVisible",
          "Namespace": "AWS/SQS",
          "Dimensions": [
            {
              "Name": "QueueName",
              "Value": "my-queue"
            }
          ],
        },
        "Stat": "Sum"
      },
      "ReturnData": true
    }
  ]
}
}

```

Contoh mengembalikan ARN kebijakan.

```

{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-sqs-custom-metrics-policy",
  "Alarms": []
}

```

Contoh kebijakan penskalaan prediktif untuk digunakan dalam skenario penerapan biru/hijau ()AWS CLI

Ekspresi penelusuran menyediakan opsi lanjutan di mana Anda dapat melakukan kueri metrik dari beberapa grup Auto Scaling dan melakukan ekspresi matematika pada mereka. Ini sangat berguna untuk penerapan biru/hijau.

Note

Penerapan biru/hijau adalah metode penerapan di mana Anda membuat dua grup Auto Scaling yang terpisah namun identik. Hanya satu dari kelompok yang menerima lalu lintas produksi. Lalu lintas pengguna awalnya diarahkan ke grup Auto Scaling sebelumnya (“biru”), sedangkan grup baru (“hijau”) digunakan untuk pengujian dan evaluasi versi baru aplikasi atau layanan. Lalu lintas pengguna dialihkan ke grup Auto Scaling hijau setelah penerapan baru diuji dan diterima. Anda kemudian dapat menghapus grup biru setelah penerapan berhasil.

Saat grup Auto Scaling baru dibuat sebagai bagian dari penerapan biru/hijau, riwayat metrik setiap grup dapat secara otomatis disertakan dalam kebijakan penskalaan prediktif tanpa Anda harus mengubah spesifikasi metriknya. Untuk informasi selengkapnya, lihat [Menggunakan kebijakan penskalaan prediktif EC2 Auto Scaling dengan penerapan Biru/Hijau di Blog Komputasi](#). AWS

Contoh kebijakan berikut menunjukkan bagaimana hal ini dapat dilakukan. Dalam contoh ini, kebijakan menggunakan CPUUtilization metrik yang dipancarkan oleh Amazon EC2. Ini menggunakan metrik Auto GroupInServiceInstances Scaling Amazon EC2 dan ekspresi matematika untuk menghitung nilai metrik penskalaan per instance. Ini juga menentukan spesifikasi metrik kapasitas untuk mendapatkan GroupInServiceInstances metrik.

Ekspresi pencarian menemukan instance di beberapa grup Auto Scaling berdasarkan kriteria pencarian yang ditentukan. CPUUtilization Jika nanti Anda membuat grup Auto Scaling baru yang cocok dengan kriteria penelusuran yang sama, instance di grup Auto Scaling baru akan disertakan secara otomatis. CPUUtilization

```
aws autoscaling put-scaling-policy --policy-name my-blue-green-predictive-scaling-policy \  
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \  
  --predictive-scaling-configuration file://config.json  
{  
  "MetricSpecifications": [  
    {  
      "TargetValue": 25,  
      "CustomizedScalingMetricSpecification": {  
        "MetricDataQueries": [  
          {  
            "Id": "load_sum",
```

```

      "Expression": "SUM(SEARCH('{AWS/EC2,AutoScalingGroupName} MetricName=
\"CPUUtilization\" ASG-myapp', 'Sum', 300))",
      "ReturnData": false
    },
    {
      "Id": "capacity_sum",
      "Expression": "SUM(SEARCH('{AWS/AutoScaling,AutoScalingGroupName}
MetricName=\"GroupInServiceInstances\" ASG-myapp', 'Average', 300))",
      "ReturnData": false
    },
    {
      "Id": "weighted_average",
      "Expression": "load_sum / capacity_sum",
      "ReturnData": true
    }
  ]
},
"CustomizedLoadMetricSpecification": {
  "MetricDataQueries": [
    {
      "Id": "load_sum",
      "Expression": "SUM(SEARCH('{AWS/EC2,AutoScalingGroupName} MetricName=
\"CPUUtilization\" ASG-myapp', 'Sum', 3600))"
    }
  ]
},
"CustomizedCapacityMetricSpecification": {
  "MetricDataQueries": [
    {
      "Id": "capacity_sum",
      "Expression": "SUM(SEARCH('{AWS/AutoScaling,AutoScalingGroupName}
MetricName=\"GroupInServiceInstances\" ASG-myapp', 'Average', 300))"
    }
  ]
}
}

```

Contoh mengembalikan ARN kebijakan.

```
{
```

```
"PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-blue-green-predictive-scaling-policy",
  "Alarms": []
}
```

Pertimbangan dan pemecahan masalah

Jika terjadi masalah saat menggunakan metrik kustom, sebaiknya Anda melakukan hal berikut:

- Jika pesan kesalahan disediakan, baca pesan dan selesaikan masalah yang dilaporkan, jika memungkinkan.
- Jika masalah terjadi saat Anda mencoba menggunakan ekspresi penelusuran dalam skenario penerapan biru/hijau, pertama-tama pastikan Anda memahami cara membuat ekspresi penelusuran yang mencari kecocokan sebagian, bukan kecocokan persis. Juga, periksa apakah kueri Anda hanya menemukan grup Auto Scaling yang menjalankan aplikasi tertentu. Untuk informasi selengkapnya tentang sintaks ekspresi penelusuran, lihat [sintaks ekspresi CloudWatch penelusuran](#) di CloudWatch Panduan Pengguna Amazon.
- Jika Anda tidak memvalidasi ekspresi sebelumnya, [put-scaling-policy](#) perintah memvalidasinya saat Anda membuat kebijakan penskalaan. Namun, ada kemungkinan bahwa perintah ini mungkin gagal mengidentifikasi penyebab pasti dari kesalahan yang terdeteksi. Untuk memperbaiki masalah, pecahkan masalah kesalahan yang Anda terima dalam respons dari permintaan ke perintah. [get-metric-data](#) Anda juga dapat memecahkan masalah ekspresi dari konsol. CloudWatch
- Saat Anda melihat grafik Beban dan Kapasitas di konsol, grafik Kapasitas mungkin tidak menampilkan data apa pun. Untuk memastikan bahwa grafik memiliki data lengkap, pastikan Anda mengaktifkan metrik grup secara konsisten untuk grup Auto Scaling Anda. Untuk informasi selengkapnya, lihat [Aktifkan metrik grup Auto Scaling \(konsol\)](#).
- Spesifikasi metrik kapasitas hanya berguna untuk penerapan biru/hijau ketika Anda memiliki aplikasi yang berjalan di grup Auto Scaling yang berbeda selama masa pakainya. Metrik khusus ini memungkinkan Anda menyediakan kapasitas total beberapa grup Auto Scaling. Penskalaan prediktif menggunakan ini untuk menampilkan data historis dalam grafik Kapasitas di konsol.
- Anda harus menentukan `false` `ReturnData` jika `MetricDataQueries` menentukan fungsi `SEARCH ()` sendiri tanpa fungsi matematika seperti `SUM ()`. Ini karena ekspresi penelusuran mungkin mengembalikan beberapa deret waktu, dan spesifikasi metrik berdasarkan ekspresi hanya dapat mengembalikan satu deret waktu.
- Semua metrik yang terlibat dalam ekspresi pencarian harus memiliki resolusi yang sama.

Batasan

- Anda dapat menanyakan titik data hingga 10 metrik dalam satu spesifikasi metrik.
- Untuk tujuan batas ini, satu ekspresi dihitung sebagai satu metrik.

Kontrol instans Auto Scaling mana yang akan dihentikan selama penskalaan

Auto Scaling Amazon EC2 menggunakan kebijakan penghentian untuk memutuskan urutan penghentian instans. Anda dapat menggunakan kebijakan yang telah ditentukan sebelumnya atau membuat kebijakan khusus untuk memenuhi persyaratan spesifik Anda. Dengan menggunakan kebijakan khusus atau perlindungan penskalaan instance, Anda juga dapat mencegah grup Auto Scaling menghentikan instance yang belum siap untuk dihentikan.

Daftar Isi

- [Saat Auto Scaling Amazon EC2 menggunakan kebijakan penghentian](#)
- [Mengonfigurasi kebijakan penghentian untuk Auto Scaling Amazon EC2](#)
- [Membuat kebijakan penghentian kustom dengan Lambda](#)
- [Gunakan perlindungan skala dalam instance](#)
- [Rancang aplikasi Anda di Amazon EC2 Auto Scaling untuk menangani penghentian instans dengan baik](#)

Saat Auto Scaling Amazon EC2 menggunakan kebijakan penghentian

Bagian berikut menjelaskan skenario di mana Auto Scaling Amazon EC2 menggunakan kebijakan penghentian.

Daftar Isi

- [Peristiwa penurunan skala](#)
- [Refresh instans](#)
- [Rebalancing Zona Ketersediaan](#)

Peristiwa penurunan skala

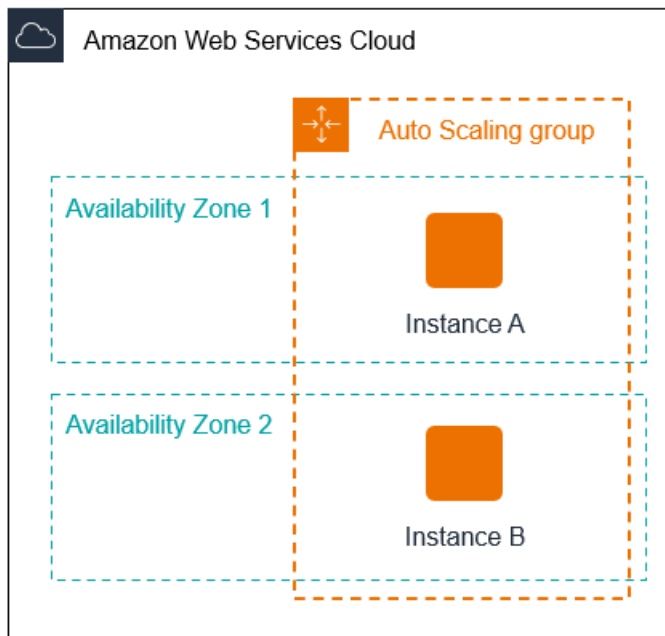
Peristiwa scale-in terjadi ketika ada nilai baru untuk kapasitas yang diinginkan dari grup Auto Scaling yang lebih rendah dari kapasitas grup saat ini.

Peristiwa scale-in terjadi dalam skenario berikut:

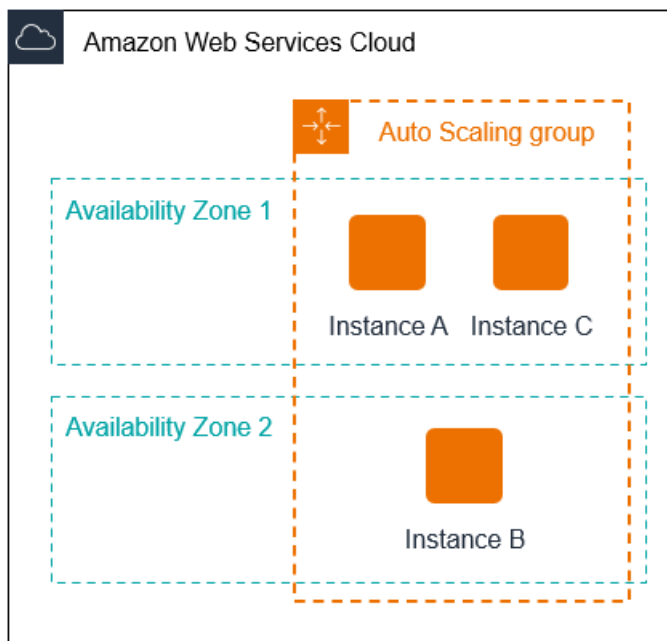
- Saat menggunakan kebijakan penskalaan dinamis dan ukuran grup berkurang sebagai akibat dari perubahan nilai metrik
- Saat menggunakan penskalaan terjadwal dan ukuran grup berkurang sebagai akibat dari tindakan terjadwal
- Ketika Anda secara manual mengurangi ukuran grup

Contoh berikut menunjukkan cara kerja kebijakan penghentian ketika ada peristiwa scale-in.

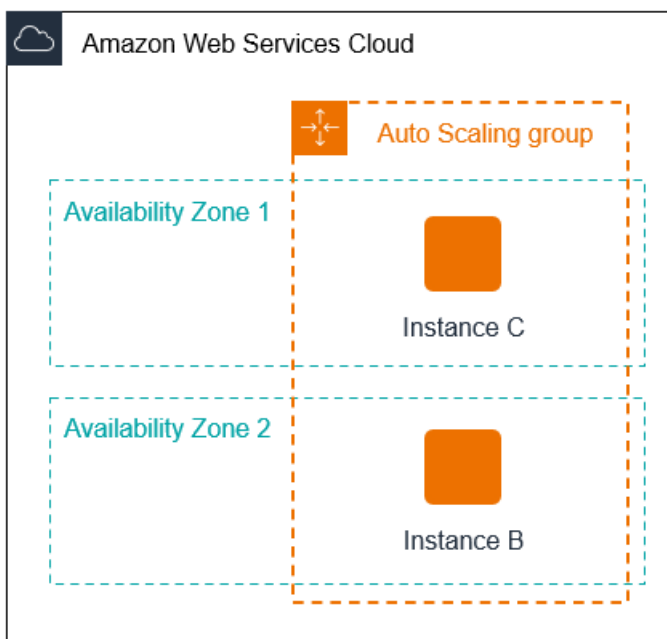
1. Grup Auto Scaling dalam contoh ini memiliki satu tipe instans, dua Availability Zone, dan kapasitas dua instance yang diinginkan. Ini juga memiliki kebijakan penskalaan dinamis yang menambah dan menghapus contoh ketika pemanfaatan sumber daya meningkat atau menurun. Dua contoh dalam grup ini didistribusikan di dua Availability Zone seperti yang ditunjukkan pada diagram berikut.



2. Ketika grup Auto Scaling meningkatkan skala, Amazon EC2 Auto Scaling meluncurkan instance baru. Grup Auto Scaling sekarang memiliki tiga instance, didistribusikan di dua Availability Zone seperti yang ditunjukkan pada diagram berikut.



3. Ketika grup Auto Scaling menurunkan skala, Amazon EC2 Auto Scaling menghentikan salah satu instance.
4. Jika Anda tidak menetapkan kebijakan penghentian tertentu ke grup, Auto Scaling Amazon EC2 menggunakan kebijakan penghentian default. Ini memilih Availability Zone dengan dua instance, dan mengakhiri instance yang diluncurkan dari konfigurasi peluncuran, template peluncuran yang berbeda, atau versi tertua dari template peluncuran saat ini. Jika instans diluncurkan dari template dan versi peluncuran yang sama, Auto Scaling Amazon EC2 akan memilih instans yang paling dekat dengan jam penagihan berikutnya dan menghentikannya.



Refresh instans

Anda dapat memulai penyegaran instans untuk memperbarui instans di grup Auto Scaling Anda. Selama penyegaran instans, Auto Scaling Amazon EC2 menghentikan instans dalam grup dan kemudian meluncurkan penggantian untuk instans yang dihentikan. Kebijakan penghentian untuk grup Auto Scaling mengontrol instance mana yang diganti terlebih dahulu.

Rebalancing Zona Ketersediaan

Auto Scaling Amazon EC2 menyeimbangkan kapasitas Anda secara merata di seluruh Availability Zone yang diaktifkan untuk grup Auto Scaling Anda. Ini membantu mengurangi dampak pemadaman Availability Zone. Jika distribusi kapasitas di seluruh Availability Zone menjadi tidak seimbang, Amazon EC2 Auto Scaling menyeimbangkan kembali grup Auto Scaling dengan meluncurkan instans di Availability Zone yang diaktifkan dengan instans paling sedikit dan instans penghentian di tempat lain. Kebijakan penghentian mengontrol instance mana yang diprioritaskan untuk penghentian terlebih dahulu.

Ada sejumlah alasan mengapa distribusi instans di seluruh Availability Zone dapat menjadi tidak seimbang.

Menghapus instance

Jika Anda melepaskan instans dari grup Auto Scaling, Anda menempatkan instans dalam keadaan siaga, atau Anda secara eksplisit menghentikan instance dan mengurangi kapasitas yang diinginkan, yang mencegah peluncuran instance pengganti, grup dapat menjadi tidak seimbang. Jika ini terjadi, Amazon EC2 Auto Scaling akan mengompensasi ulang Zona Ketersediaan.

Menggunakan Availability Zone yang berbeda dari yang ditentukan semula

Jika Anda memperluas grup Auto Scaling untuk menyertakan Availability Zone tambahan, atau mengubah Availability Zone yang digunakan, Auto Scaling Amazon EC2 akan meluncurkan instans di Availability Zone baru dan menghentikan instans di zona lain untuk membantu memastikan bahwa grup Auto Scaling mencakup Availability Zone secara merata.

Penghentian ketersediaan

Penghentian ketersediaan jarang terjadi. Namun, jika satu Availability Zone menjadi tidak tersedia dan pulih nanti, grup Auto Scaling Anda dapat menjadi tidak seimbang antara Availability Zone. Auto Scaling Amazon EC2 mencoba menyeimbangkan kembali grup secara bertahap, dan penyeimbangan kembali dapat menghentikan instans di zona lain.

Misalnya, bayangkan Anda memiliki grup Auto Scaling yang memiliki satu tipe instans, dua Availability Zone, dan kapasitas dua instans yang diinginkan. Dalam situasi ketika satu Zona Ketersediaan gagal, Amazon EC2 Auto Scaling secara otomatis meluncurkan instance baru dalam Zona Ketersediaan yang sehat untuk menggantikan yang berada di Zona Ketersediaan yang tidak sehat. Kemudian, ketika Availability Zone yang tidak sehat kembali ke status sehat nanti, Auto Scaling Amazon EC2 secara otomatis meluncurkan instance baru di zona ini, yang pada gilirannya menghentikan instance di zona yang tidak terpengaruh.

Note

Ketika melakukan rebalancing, Amazon EC2 Auto Scaling meluncurkan instance baru sebelum menghentikan yang lama, sehingga rebalancing tidak mengganggu kinerja atau ketersediaan aplikasi Anda.

Karena Amazon EC2 Auto Scaling mencoba meluncurkan instans baru sebelum menghentikan instans lama, berada pada atau mendekati kapasitas maksimum yang ditentukan dapat menghambat atau sepenuhnya menghentikan aktivitas penyeimbangan kembali. Untuk menghindari masalah ini, sistem sementara dapat melebihi kapasitas maksimum yang ditentukan dari suatu kelompok dengan margin 10 persen (atau dengan margin satu contoh, mana yang lebih besar) selama aktivitas penyeimbangan kembali. Margin ditingkatkan hanya jika grup berada pada atau mendekati kapasitas maksimum dan memerlukan rebalancing, baik karena rezoning yang diminta pengguna atau untuk mengompensasi masalah ketersediaan zona. Perpanjangan hanya berlangsung selama diperlukan untuk menyeimbangkan kembali grup.

Mengonfigurasi kebijakan penghentian untuk Auto Scaling Amazon EC2

Kebijakan penghentian memberikan kriteria yang diikuti oleh Auto Scaling Amazon EC2 untuk menghentikan instans dalam urutan tertentu.

Secara default, Auto Scaling Amazon EC2 menggunakan kebijakan penghentian yang dirancang untuk menghentikan instans yang menggunakan konfigurasi usang terlebih dahulu. Anda dapat mengubah kebijakan penghentian untuk mengontrol instans mana yang paling penting untuk dihentikan terlebih dahulu.

Saat Auto Scaling Amazon EC2 menghentikan instans, penskalaan akan mencoba menjaga keseimbangan di seluruh Availability Zone yang diaktifkan untuk grup Auto Scaling Anda. Menjaga

keseimbangan Zonal lebih diutamakan daripada kebijakan penghentian. Jika satu Availability Zone memiliki lebih banyak instans daripada yang lain, Auto Scaling Amazon EC2 menerapkan kebijakan penghentian ke zona tidak seimbang terlebih dahulu. Jika Availability Zone seimbang, maka akan berlaku kebijakan penghentian di semua Zona.

Topik

- [Cara kerja kebijakan penghentian default](#)
- [Kebijakan penghentian default dan grup instance campuran](#)
- [Kebijakan penghentian yang telah ditentukan](#)
- [Mengubah kebijakan penghentian untuk grup Auto Scaling](#)

Cara kerja kebijakan penghentian default

Jika Auto Scaling Amazon EC2 perlu menghentikan instans, pertama kali mengidentifikasi Availability Zone (atau Zona) mana yang memiliki instans terbanyak dan setidaknya satu instance yang tidak dilindungi dari skala masuk. Kemudian, ia melanjutkan untuk mengevaluasi instance yang tidak dilindungi dalam Availability Zone yang diidentifikasi sebagai berikut:

Instance yang menggunakan konfigurasi usang

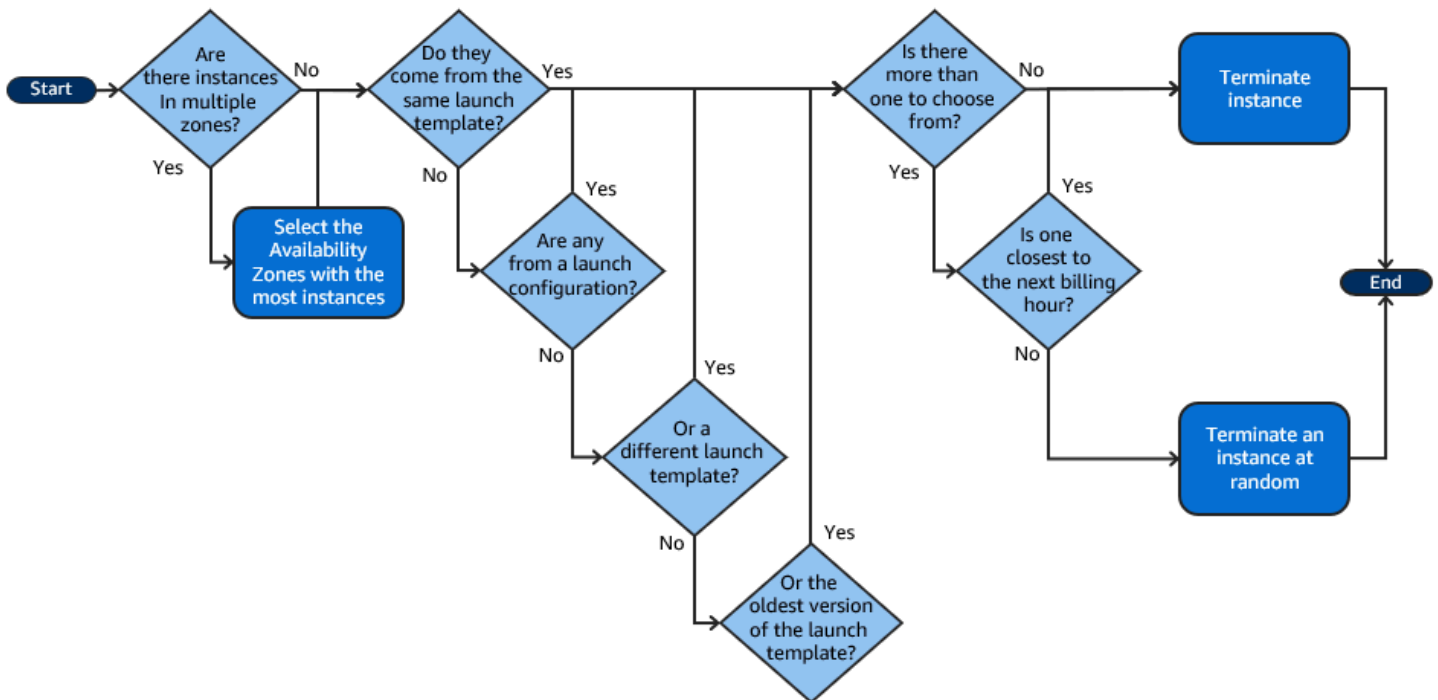
- Untuk grup yang menggunakan templat peluncuran — Tentukan apakah salah satu instance menggunakan konfigurasi yang sudah ketinggalan zaman, dengan memprioritaskan dalam urutan ini:
 1. Pertama, periksa instance yang diluncurkan dengan konfigurasi peluncuran.
 2. Kemudian, periksa instance yang diluncurkan menggunakan template peluncuran yang berbeda alih-alih templat peluncuran saat ini.
 3. Terakhir, periksa instance yang menggunakan versi tertua dari template peluncuran saat ini.
- Untuk grup yang menggunakan konfigurasi peluncuran — Tentukan apakah salah satu instance menggunakan konfigurasi peluncuran tertua.

Jika tidak ditemukan instans dengan konfigurasi usang, atau ada beberapa instans yang dapat dipilih, Auto Scaling Amazon EC2 mempertimbangkan kriteria instans berikutnya yang mendekati jam penagihan berikutnya.

Instans mendekati jam penagihan berikutnya

Tentukan apakah salah satu instans yang memenuhi kriteria sebelumnya paling dekat dengan jam penagihan berikutnya. Jika beberapa instance sama-sama dekat, hentikan satu secara acak. Ini membantu Anda memaksimalkan penggunaan instans Anda yang ditagih setiap jam. Namun, sebagian besar penggunaan EC2 sekarang ditagih per detik, sehingga pengoptimalan ini memberikan manfaat yang lebih sedikit. Untuk informasi selengkapnya, lihat [Harga Amazon EC2](#).

Diagram alir berikut menggambarkan cara kerja kebijakan penghentian default untuk grup yang menggunakan templat peluncuran.



Kebijakan penghentian default dan grup instance campuran

[Auto Scaling Amazon EC2 menerapkan kriteria tambahan saat menghentikan instans dalam grup instans campuran.](#)

Jika Auto Scaling Amazon EC2 perlu menghentikan instans, pertama kali mengidentifikasi opsi pembelian mana (Spot atau Sesuai Permintaan) yang harus dihentikan berdasarkan pengaturan grup. Ini memastikan bahwa kelompok cenderung ke arah rasio yang ditentukan dari instans Spot dan On-Demand dari waktu ke waktu.

Ini kemudian menerapkan kebijakan penghentian secara independen dalam setiap Availability Zone. Ini menentukan Instans Spot atau Sesuai Permintaan di mana Availability Zone akan dihentikan agar Availability Zone tetap seimbang. Logika yang sama berlaku untuk grup instance campuran dengan bobot yang ditentukan untuk tipe instance.

Dalam setiap zona, kebijakan penghentian default berfungsi sebagai berikut untuk menentukan instance yang tidak dilindungi dalam opsi pembelian yang diidentifikasi dapat dihentikan:

1. Tentukan apakah salah satu instans dapat dihentikan untuk meningkatkan keselarasan dengan [strategi alokasi](#) yang ditentukan untuk grup Auto Scaling. Jika tidak ada contoh yang diidentifikasi untuk pengoptimalan, atau ada beberapa contoh untuk dipilih, evaluasi berlanjut.
2. Tentukan apakah salah satu instance menggunakan konfigurasi usang, memprioritaskan dalam urutan ini:
 - a. Pertama, periksa instance yang diluncurkan dengan konfigurasi peluncuran.
 - b. Kemudian, periksa instance yang diluncurkan menggunakan template peluncuran yang berbeda alih-alih templat peluncuran saat ini.
 - c. Terakhir, periksa instance yang menggunakan versi tertua dari template peluncuran saat ini.

Jika tidak ada instance dengan konfigurasi usang yang ditemukan, atau ada beberapa instance untuk dipilih, evaluasi akan berlanjut.

3. Tentukan apakah salah satu instans paling dekat dengan jam penagihan berikutnya. Jika beberapa instance sama-sama dekat, pilih satu secara acak.

Kebijakan penghentian yang telah ditentukan

Anda memilih dari kebijakan penghentian yang telah ditentukan sebelumnya berikut:

- **Default**— Mengakhiri instance sesuai dengan kebijakan penghentian default.
- **AllocationStrategy**— Hentikan instans dalam grup Auto Scaling untuk menyelaraskan instans yang tersisa dengan strategi alokasi untuk jenis instans yang berakhir (baik Instans Spot atau Instans Sesuai Permintaan). Kebijakan ini berguna jika jenis instance pilihan Anda telah berubah. Jika strategi alokasi Spot adalah `lowest-price`, secara bertahap Anda dapat menyeimbangkan kembali distribusi Instance Spot di seluruh kumpulan Spot dengan harga terendah di N. Jika strategi alokasi Spot adalah `capacity-optimized`, secara bertahap Anda dapat menyeimbangkan kembali distribusi Instance Spots di seluruh kumpulan Spot yang memiliki kapasitas Spot yang lebih banyak. Anda juga dapat secara bertahap mengganti Instance On-Demand dari tipe prioritas lebih rendah dengan Instance On-Demand dari tipe prioritas yang lebih tinggi.
- **OldestLaunchTemplate**— Hentikan instance yang memiliki template peluncuran tertua. Dengan kebijakan ini, instance yang menggunakan templat peluncuran non-saat ini dihentikan terlebih dahulu, diikuti dengan instance yang menggunakan versi paling lama dari templat peluncuran saat

ini. Kebijakan ini berguna saat Anda memperbarui sebuah grup dan menghapus instance dari konfigurasi sebelumnya.

- **OldestLaunchConfiguration**— Hentikan instance yang memiliki konfigurasi peluncuran tertua. Kebijakan ini berguna saat Anda memperbarui sebuah grup dan menghapus instance dari konfigurasi sebelumnya. Dengan kebijakan ini, instance yang menggunakan konfigurasi peluncuran noncurrent akan dihentikan terlebih dahulu.
- **ClosestToNextInstanceHour**— Hentikan instance yang paling dekat dengan jam penagihan berikutnya. Kebijakan ini membantu Anda memaksimalkan penggunaan instance yang memiliki biaya per jam.
- **NewestInstance**— Hentikan instance terbaru dalam grup. Kebijakan ini berguna ketika Anda menguji konfigurasi peluncuran baru tetapi tidak ingin tetap membuatnya dalam produksi.
- **OldestInstance**— Hentikan instance tertua dalam grup. Opsi ini berguna saat Anda memutakhirkan instance dalam grup Auto Scaling ke tipe instance EC2 baru. Anda dapat secara bertahap mengganti instance jenis lama dengan instance jenis baru.

Note

Amazon EC2 Auto Scaling selalu menyeimbangkan instance di seluruh Zona Ketersediaan terlebih dahulu, terlepas dari kebijakan penghentian mana yang digunakan. Akibatnya, Anda mungkin menghadapi situasi di mana beberapa instance yang lebih baru dihentikan sebelum instance yang lebih lama. Misalnya, ketika ada Availability Zone yang baru ditambahkan, atau ketika satu Availability Zone memiliki lebih banyak instance daripada Availability Zone lainnya yang digunakan oleh grup.

Mengubah kebijakan penghentian untuk grup Auto Scaling

Untuk mengubah kebijakan penghentian grup Auto Scaling Anda, gunakan salah satu metode berikut.

Console

Anda tidak dapat mengubah kebijakan penghentian saat pertama kali membuat grup Auto Scaling di konsol Auto Scaling Amazon EC2. Kebijakan penghentian default digunakan secara otomatis. Setelah grup Auto Scaling dibuat, Anda dapat mengganti kebijakan default dengan kebijakan penghentian yang berbeda atau beberapa kebijakan penghentian yang tercantum dalam urutan penerapannya.

Untuk mengubah kebijakan penghentian grup Auto Scaling

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Detail, pilih Konfigurasi lanjutan, Edit.
4. Untuk Kebijakan penghentian, pilih satu kebijakan penghentian atau lebih. Jika Anda memilih beberapa kebijakan, letakkan dalam urutan yang Anda inginkan untuk dievaluasi.

Anda dapat memilih Kebijakan penghentian kustom secara opsional dan kemudian memilih fungsi Lambda yang memenuhi kebutuhan Anda. Jika Anda telah membuat versi dan alias untuk fungsi Lambda Anda, Anda dapat memilih versi atau alias dari drop-down Versi/Alias. Untuk menggunakan versi yang tidak dipublikasikan dari fungsi Lambda Anda, tetapkan Versi/Alias ke defaultnya. Untuk informasi selengkapnya, lihat [Membuat kebijakan penghentian kustom dengan Lambda](#).

Note

Saat menggunakan beberapa kebijakan, urutannya harus disetel dengan benar:

- Jika Anda menggunakan kebijakan Default, itu harus menjadi kebijakan terakhir dalam daftar.
- Jika Anda menggunakan kebijakan penghentian kustom, itu harus menjadi kebijakan pertama dalam daftar.

5. Pilih Perbarui.

AWS CLI

Kebijakan penghentian default digunakan secara otomatis kecuali kebijakan yang berbeda ditentukan.

Untuk mengubah kebijakan penghentian grup Auto Scaling

Gunakan salah satu perintah berikut:

- [create-auto-scaling-group](#)

- [update-auto-scaling-group](#)

Anda dapat menggunakan kebijakan penghentian secara individual, atau menggabungkannya ke dalam daftar kebijakan. Misalnya, gunakan perintah berikut untuk memperbarui grup Auto Scaling untuk menggunakan kebijakan `OldestLaunchConfiguration` terlebih dahulu, kemudian menggunakan kebijakan `ClosestToNextInstanceHour`.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --  
termination-policies "OldestLaunchConfiguration" "ClosestToNextInstanceHour"
```

Jika Anda menggunakan kebijakan penghentian `Default`, menjadikannya yang terakhir dalam daftar kebijakan penghentian. Misalnya, `--termination-policies "OldestLaunchConfiguration" "Default"`.

Untuk menggunakan kebijakan penghentian kustom, Anda harus terlebih dahulu membuat kebijakan penghentian Anda menggunakan AWS Lambda. Untuk menentukan fungsi Lambda yang akan digunakan sebagai kebijakan penghentian Anda, jadikan yang pertama dalam daftar kebijakan penghentian. Misalnya, `--termination-policies "arn:aws:lambda:us-west-2:123456789012:function:HelloFunction:prod" "OldestLaunchConfiguration"`. Untuk informasi selengkapnya, lihat [Membuat kebijakan penghentian kustom dengan Lambda](#).

Membuat kebijakan penghentian kustom dengan Lambda

Auto Scaling Amazon EC2 menggunakan kebijakan penghentian untuk memprioritaskan instans mana yang akan dihentikan terlebih dahulu saat mengurangi ukuran grup Auto Scaling Anda (disebut sebagai penskalaan dalam). Grup Auto Scaling Anda menggunakan kebijakan penghentian default, tetapi Anda dapat memilih atau membuat kebijakan penghentian sendiri secara opsional. Untuk informasi selengkapnya tentang memilih kebijakan penghentian yang telah ditentukan sebelumnya, lihat [Mengonfigurasi kebijakan penghentian untuk Auto Scaling Amazon EC2](#).

Dalam topik ini, Anda mempelajari cara membuat kebijakan penghentian kustom menggunakan AWS Lambda fungsi yang dipanggil Auto Scaling Amazon EC2 sebagai respons terhadap peristiwa tertentu. Fungsi Lambda yang Anda buat memproses informasi dalam data input yang dikirim oleh Amazon EC2 Auto Scaling dan menampilkan daftar instance yang siap dihentikan.

Kebijakan penghentian kustom memberikan kontrol yang lebih baik atas instance mana yang dihentikan, dan kapan. Misalnya, saat grup Auto Scaling Anda masuk, Auto Scaling Amazon EC2

tidak dapat menentukan apakah ada beban kerja yang berjalan yang tidak boleh terganggu. Dengan fungsi Lambda, Anda dapat memvalidasi permintaan penghentian dan menunggu hingga beban kerja selesai sebelum mengembalikan ID instans ke Amazon EC2 Auto Scaling untuk penghentian.

Daftar Isi

- [Data input](#)
- [Data respons](#)
- [Pertimbangan](#)
- [Buat fungsi Lambda](#)
- [Batasan](#)

Data input

Auto Scaling Amazon EC2 menghasilkan payload JSON untuk peristiwa penskalaan, dan juga melakukannya saat instance akan dihentikan sebagai akibat dari masa pakai instans maksimum atau fitur penyegaran instans. Ini juga menghasilkan payload JSON untuk peristiwa scale-in yang dapat dimulai saat menyeimbangkan kembali grup Anda di seluruh Availability Zone.

Payload ini berisi informasi tentang kapasitas yang dibutuhkan Amazon EC2 Auto Scaling untuk dihentikan, daftar instans yang disarankan untuk penghentian, dan peristiwa yang memulai penghentian.

Berikut ini adalah contoh payload:

```
{
  "AutoScalingGroupARN": "arn:aws:autoscaling:us-east-1:<account-id>:autoScalingGroup:d4738357-2d40-4038-ae7e-b00ae0227003:autoScalingGroupName/my-asg",
  "AutoScalingGroupName": "my-asg",
  "CapacityToTerminate": [
    {
      "AvailabilityZone": "us-east-1b",
      "Capacity": 2,
      "InstanceMarketOption": "on-demand"
    },
    {
      "AvailabilityZone": "us-east-1b",
      "Capacity": 1,
      "InstanceMarketOption": "spot"
    }
  ],
}
```

```

    {
      "AvailabilityZone": "us-east-1c",
      "Capacity": 3,
      "InstanceMarketOption": "on-demand"
    }
  ],
  "Instances": [
    {
      "AvailabilityZone": "us-east-1b",
      "InstanceId": "i-0056faf8da3e1f75d",
      "InstanceType": "t2.nano",
      "InstanceMarketOption": "on-demand"
    },
    {
      "AvailabilityZone": "us-east-1c",
      "InstanceId": "i-02e1c69383a3ed501",
      "InstanceType": "t2.nano",
      "InstanceMarketOption": "on-demand"
    },
    {
      "AvailabilityZone": "us-east-1c",
      "InstanceId": "i-036bc44b6092c01c7",
      "InstanceType": "t2.nano",
      "InstanceMarketOption": "on-demand"
    },
    ...
  ],
  "Cause": "SCALE_IN"
}

```

Payload termasuk nama grup Auto Scaling, Amazon Resource Name (ARN), dan elemen-elemen berikut:

- `CapacityToTerminate` menjelaskan berapa banyak kapasitas Spot atau Sesuai Permintaan Anda yang ditetapkan untuk dihentikan di Availability Zone tertentu.
- `Instances` mewakili instans yang disarankan oleh Auto Scaling Amazon EC2 untuk penghentian berdasarkan informasi di `CapacityToTerminate`
- `Cause` menjelaskan peristiwa yang menyebabkan penghentian: `SCALE_IN`, `INSTANCE_REFRESH`, `MAX_INSTANCE_LIFETIME`, atau `REBALANCE`.

Informasi berikut menguraikan faktor paling signifikan dalam cara Auto Scaling Amazon EC2 menghasilkan Instances data input:

- Menjaga keseimbangan di seluruh Availability Zone akan diutamakan saat instance dihentikan karena peristiwa scale-in dan penghentian berbasis penyegaran instance. Oleh karena itu, jika satu Availability Zone memiliki lebih banyak instance daripada Availability Zone lainnya yang digunakan oleh grup, data input berisi instance yang memenuhi syarat untuk penghentian hanya dari Availability Zone yang tidak seimbang. Jika Availability Zone yang digunakan oleh grup seimbang, data input berisi instance dari semua Availability Zone untuk grup.
- Saat menggunakan [kebijakan instance campuran](#), menjaga keseimbangan kapasitas Spot dan On-Demand Anda berdasarkan persentase yang Anda inginkan untuk setiap opsi pembelian juga diutamakan. Kami pertama-tama mengidentifikasi mana dari dua jenis (Spot atau On-Demand) yang harus dihentikan. Kami kemudian mengidentifikasi instance mana (dalam opsi pembelian yang diidentifikasi) di Zona Ketersediaan mana yang dapat kami hentikan yang akan menghasilkan Availability Zone yang paling seimbang.

Data respons

Data input dan data respons bekerja sama untuk mempersempit daftar instance yang akan dihentikan.

Dengan input yang diberikan, respons dari fungsi Lambda Anda akan terlihat seperti contoh berikut:

```
{
  "InstanceIDs": [
    "i-02e1c69383a3ed501",
    "i-036bc44b6092c01c7",
    ...
  ]
}
```

InstanceIDs Dalam respon mewakili contoh yang siap untuk dihentikan.

Atau, Anda dapat mengembalikan serangkaian instance berbeda yang siap untuk dihentikan, yang mengesampingkan instance dalam data input. Jika tidak ada instance yang siap dihentikan saat fungsi Lambda Anda dipanggil, Anda juga dapat memilih untuk tidak mengembalikan instance apa pun.

Ketika tidak ada instance yang siap untuk dihentikan, respons dari fungsi Lambda Anda akan terlihat seperti contoh berikut:

```
{  
  "InstanceIDs": [ ]  
}
```

Pertimbangan

Perhatikan pertimbangan berikut saat menggunakan kebijakan penghentian kustom:

- Mengembalikan instance terlebih dahulu dalam data respons tidak menjamin penghentiannya. Jika lebih dari jumlah instans yang diperlukan dikembalikan saat fungsi Lambda Anda dipanggil, Auto Scaling Amazon EC2 mengevaluasi setiap instance terhadap kebijakan penghentian lain yang Anda tentukan untuk grup Auto Scaling Anda. Ketika ada beberapa kebijakan penghentian, ia mencoba menerapkan kebijakan penghentian berikutnya dalam daftar, dan jika ada lebih banyak contoh daripada yang diperlukan untuk mengakhiri, itu beralih ke kebijakan penghentian berikutnya, dan seterusnya. Jika tidak ada kebijakan penghentian lain yang ditentukan, maka kebijakan penghentian default digunakan untuk menentukan instance mana yang akan dihentikan.
- Jika tidak ada instans yang dikembalikan atau fungsi Lambda Anda habis, maka Auto Scaling Amazon EC2 menunggu beberapa saat sebelum menjalankan fungsi Anda lagi. Untuk acara skala apa pun, ia terus berusaha selama kapasitas yang diinginkan grup kurang dari kapasitas saat ini. Misalnya penghentian berbasis refresh, ia terus mencoba selama satu jam. Setelah itu, jika terus gagal untuk menghentikan instance apa pun, operasi penyegaran instance gagal. Dengan masa pakai instans maksimum, Auto Scaling Amazon EC2 terus mencoba menghentikan instans yang diidentifikasi melebihi masa pakai maksimumnya.
- Karena fungsi Anda dicoba berulang kali, pastikan untuk menguji dan memperbaiki kesalahan permanen apa pun dalam kode Anda sebelum menggunakan fungsi Lambda sebagai kebijakan penghentian kustom.
- Jika Anda mengganti data input dengan daftar instans Anda sendiri untuk dihentikan, dan menghentikan instans ini akan membuat Availability Zone tidak seimbang, Auto Scaling Amazon EC2 secara bertahap menyeimbangkan kembali distribusi kapasitas di seluruh Availability Zone. Pertama, ini memanggil fungsi Lambda Anda untuk melihat apakah ada instance yang siap dihentikan sehingga dapat menentukan apakah akan mulai menyeimbangkan kembali. Jika ada instance yang siap dihentikan, itu meluncurkan instance baru terlebih dahulu. Ketika instance selesai diluncurkan, ia kemudian mendeteksi bahwa kapasitas grup Anda saat ini lebih tinggi dari kapasitas yang diinginkan dan memulai acara scale-in.

- Kebijakan penghentian kustom tidak memengaruhi kemampuan Anda untuk juga menggunakan perlindungan skala dalam untuk melindungi instans tertentu agar tidak dihentikan. Untuk informasi selengkapnya, lihat [Gunakan perlindungan skala dalam instance](#).

Buat fungsi Lambda

Mulailah dengan membuat fungsi Lambda, sehingga Anda dapat menentukan Nama Sumber Daya Amazon (ARN) dalam kebijakan penghentian untuk grup Auto Scaling Anda.

Untuk membuat fungsi Lambda (konsol)

1. Buka [halaman Fungsi](#) di konsol Lambda.
2. Pada bilah navigasi di bagian atas layar, pilih Wilayah yang sama dengan yang Anda gunakan saat membuat grup Auto Scaling.
3. Pilih Create function, Author dari awal.
4. Di bawah Informasi dasar, untuk nama Fungsi, masukkan nama fungsi Anda.
5. Pilih Buat fungsi. Anda dikembalikan ke kode dan konfigurasi fungsi.
6. Dengan fungsi Anda masih terbuka di konsol, di bawah Kode fungsi, tempel kode Anda ke editor.
7. Pilih Deploy.
8. Secara opsional, buat versi yang diterbitkan dari fungsi Lambda dengan memilih tab Versi dan kemudian Publikasikan versi baru. Untuk mempelajari lebih lanjut tentang pembuatan versi di Lambda, lihat Versi [fungsi Lambda](#) di Panduan Pengembang AWS Lambda
9. Jika Anda memilih untuk menerbitkan versi, pilih tab Alias jika Anda ingin mengaitkan alias dengan versi fungsi Lambda ini. Untuk mempelajari lebih lanjut tentang alias di Lambda, lihat Alias [fungsi Lambda](#) di Panduan Pengembang AWS Lambda
10. Selanjutnya, pilih tab Konfigurasi dan kemudian Izin.
11. Gulir ke bawah ke kebijakan berbasis sumber daya, lalu pilih Tambahkan izin. Kebijakan berbasis sumber daya digunakan untuk memberikan izin untuk menjalankan fungsi Anda ke prinsipal yang ditentukan dalam kebijakan. Dalam hal ini, prinsipnya adalah peran [terkait layanan Auto Scaling Amazon EC2 yang terkait](#) dengan grup Auto Scaling.
12. Di bagian Pernyataan kebijakan, konfigurasi izin Anda:
 - a. Pilih Akun AWS.

- b. Untuk Principal, masukkan ARN dari peran terkait layanan panggilan, misalnya,
arn:aws:iam::<aws-account-id>:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling
 - c. Untuk Tindakan, pilih lambda: InvokeFunction.
 - d. Untuk ID Pernyataan, masukkan ID pernyataan unik, seperti **AllowInvokeByAutoScaling**.
 - e. Pilih Simpan.
13. Setelah Anda mengikuti petunjuk ini, lanjutkan untuk menentukan ARN fungsi Anda dalam kebijakan penghentian untuk grup Auto Scaling Anda sebagai langkah berikutnya. Untuk informasi selengkapnya, lihat [Mengubah kebijakan penghentian untuk grup Auto Scaling](#).

Note

Untuk contoh yang dapat Anda gunakan sebagai referensi untuk mengembangkan fungsi Lambda, lihat [GitHub repositori untuk Auto Scaling Amazon EC2](#).

Batasan

- Anda hanya dapat menentukan satu fungsi Lambda dalam kebijakan penghentian untuk grup Auto Scaling. Jika ada beberapa kebijakan penghentian yang ditentukan, fungsi Lambda harus ditentukan terlebih dahulu.
- Anda dapat mereferensikan fungsi Lambda Anda menggunakan ARN yang tidak memenuhi syarat (tanpa akhiran) atau ARN yang memenuhi syarat yang memiliki versi atau alias sebagai sufiksnya. Jika ARN yang tidak memenuhi syarat digunakan (misalnya `function:my-function`), kebijakan berbasis sumber daya Anda harus dibuat pada versi fungsi Anda yang tidak dipublikasikan. Jika ARN yang memenuhi syarat digunakan (misalnya, `function:my-function:1` atau `function:my-function:prod`), kebijakan berbasis sumber daya Anda harus dibuat pada versi spesifik yang diterbitkan dari fungsi Anda.
- Anda tidak dapat menggunakan ARN yang memenuhi syarat dengan akhiran. `$LATEST` Jika Anda mencoba menambahkan kebijakan penghentian kustom yang mengacu pada ARN yang memenuhi syarat dengan `$LATEST` akhiran, itu akan mengakibatkan kesalahan.
- Jumlah instans yang disediakan dalam data input dibatasi hingga 30.000 instance. Jika ada lebih dari 30.000 instance yang dapat dihentikan, data input termasuk `"HasMoreInstances": true` untuk menunjukkan bahwa jumlah maksimum instance dikembalikan.

- Waktu berjalan maksimum untuk fungsi Lambda Anda adalah dua detik (2000 milidetik). Sebagai praktik terbaik, Anda harus menetapkan nilai batas waktu fungsi Lambda Anda berdasarkan waktu berjalan yang Anda harapkan. Fungsi Lambda memiliki batas waktu default tiga detik, tetapi ini dapat dikurangi.
- Jika runtime Anda melebihi batas 2 detik, tindakan scale-in apa pun akan ditunda hingga runtime turun di bawah ambang batas ini. Untuk fungsi Lambda dengan runtime yang lebih lama secara konsisten, temukan cara untuk mengurangi runtime, seperti dengan menyimpan hasil di mana mereka dapat diambil selama pemanggilan Lambda berikutnya.

Gunakan perlindungan skala dalam instance

Perlindungan penskalaan instans memberi Anda kontrol atas instans mana yang dapat dihentikan Auto Scaling Amazon EC2. Kasus penggunaan umum untuk fitur ini adalah penskalaan beban kerja berbasis container. Untuk informasi selengkapnya, lihat [Rancang aplikasi Anda di Amazon EC2 Auto Scaling untuk menangani penghentian instans dengan baik](#).

Secara default, perlindungan penskalaan instance dinonaktifkan saat Anda membuat grup Auto Scaling. Ini berarti bahwa Auto Scaling Amazon EC2 dapat menghentikan instans apa pun dalam grup.

Anda dapat melindungi instans segera setelah diluncurkan dengan mengaktifkan setelan perlindungan penskalaan instans pada grup Auto Scaling Anda. Perlindungan skala dalam instance dimulai saat status instance. InService Kemudian, untuk mengontrol instance mana yang dapat dihentikan, nonaktifkan pengaturan perlindungan skala pada instance individual dalam grup Auto Scaling. Dengan demikian, Anda dapat terus melindungi instance tertentu dari penghentian yang tidak diinginkan.

Topik

- [Pertimbangan](#)
- [Ubah perlindungan scale-in untuk grup Auto Scaling](#)
- [Ubah perlindungan scale-in untuk sebuah instance](#)

Pertimbangan

Berikut ini adalah pertimbangan saat menggunakan perlindungan skala dalam instance:

- Jika semua instans dalam grup Auto Scaling dilindungi dari skala masuk, dan peristiwa penskalaan terjadi, kapasitas yang diinginkan akan berkurang. Namun, grup Auto Scaling tidak dapat menghentikan jumlah instance yang diperlukan hingga pengaturan perlindungan penurunan skala instance dinonaktifkan. Dalam AWS Management Console, riwayat Aktivitas untuk grup Auto Scaling menyertakan pesan berikut jika semua instance dalam grup Auto Scaling dilindungi dari skala saat peristiwa penskalaan terjadi: `Could not scale to desired capacity because all remaining instances are protected from scale-in.`
- Jika Anda melepaskan instance yang dilindungi dari scale-in, pengaturan perlindungan skala masuknya akan hilang. Saat Anda melampirkan lagi instance ke grup, ini akan meneruskan pengaturan perlindungan penurunan skala dalam instance saat ini dari grup. Saat Amazon EC2 Auto Scaling meluncurkan instans baru atau memindahkan instance dari kolam hangat ke grup Auto Scaling, instance mewarisi setelan perlindungan penskalaan instans pada grup Auto Scaling.
- Perlindungan penurunan skala instance tidak melindungi instance Auto Scaling dari hal berikut:
 - Pemeriksaan kesehatan pengganti jika instance gagal dalam pemeriksaan kesehatan. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).
 - Temukan gangguan Instance. Instance Spot dihentikan saat kapasitas tidak lagi tersedia atau harga Spot melebihi harga maksimum Anda.
 - Reservasi Blok Kapasitas berakhir. Amazon EC2 merebut kembali instans Blok Kapasitas meskipun dilindungi dari skala masuk.
 - Pengakhiran manual melalui `terminate-instance-in-auto-scaling-group` perintah. Untuk informasi selengkapnya, lihat [Mengakhiri instance di grup Auto Scaling \(AWS CLI\)](#).
 - Penghentian manual melalui konsol Amazon EC2, perintah CLI, dan operasi API. Untuk melindungi instance Auto Scaling dari penghentian manual, aktifkan perlindungan penghentian Amazon EC2. (Ini tidak mencegah Auto Scaling Amazon EC2 menghentikan instans atau penghentian manual melalui perintah.) `terminate-instance-in-auto-scaling-group` Untuk informasi tentang mengaktifkan perlindungan penghentian Amazon EC2 di templat peluncuran, lihat [Buat template peluncuran menggunakan pengaturan lanjutan](#)

Ubah perlindungan scale-in untuk grup Auto Scaling

Anda dapat mengaktifkan atau menonaktifkan pengaturan perlindungan penurunan skala instance untuk grup Auto Scaling. Saat Anda mengaktifkannya, semua instance baru yang diluncurkan oleh grup akan mengaktifkan perlindungan skala masuk instance.

Mengaktifkan atau menonaktifkan setelan ini untuk grup Auto Scaling tidak memengaruhi instans yang ada.

Console

Untuk mengaktifkan perlindungan scale-in untuk grup Auto Scaling baru

Saat Anda membuat grup Auto Scaling, pada halaman Konfigurasi ukuran grup dan kebijakan penskalaan, di bawah Perlindungan penskalaan instans, pilih kotak centang Aktifkan perlindungan skala masuk instans.

Untuk mengaktifkan atau menonaktifkan perlindungan skala dalam untuk grup yang ada

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Detail, pilih Konfigurasi lanjutan, Edit.
4. Untuk perlindungan skala masuk Instance, pilih atau kosongkan kotak centang Aktifkan perlindungan skala masuk instance untuk mengaktifkan atau menonaktifkan opsi ini sesuai kebutuhan.
5. Pilih Perbarui.

AWS CLI

Untuk mengaktifkan perlindungan scale-in untuk grup Auto Scaling baru

Gunakan [create-auto-scaling-group](#) perintah berikut untuk mengaktifkan perlindungan skala dalam instance.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg --new-instances-protected-from-scale-in ...
```

Untuk mengaktifkan perlindungan scale-in untuk grup yang ada

Gunakan [update-auto-scaling-group](#) perintah berikut untuk mengaktifkan perlindungan skala masuk instance untuk grup Auto Scaling yang ditentukan.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --new-  
instances-protected-from-scale-in
```

Untuk menonaktifkan perlindungan scale-in untuk grup yang ada

Gunakan perintah berikut untuk menonaktifkan perlindungan penurunan skala untuk grup tertentu.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --no-new-  
instances-protected-from-scale-in
```

Ubah perlindungan scale-in untuk sebuah instance

Secara default, sebuah instance mendapatkan pengaturan perlindungan penurunan skala dari grup Auto Scaling. Namun, Anda dapat mengaktifkan atau menonaktifkan perlindungan skala masuk instance untuk setiap instance setelah diluncurkan.

Console

Untuk mengaktifkan atau menonaktifkan perlindungan skala dalam sebuah instance

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Manajemen Instance, di Instance, pilih satu instance.
4. Untuk mengaktifkan perlindungan penurunan skala, pilih Tindakan, Atur perlindungan penurunan skala. Saat diminta, pilih Atur perlindungan penurunan skala.
5. Untuk menonaktifkan perlindungan skala instance, pilih Tindakan, Hapus perlindungan penurunan skala. Saat diminta, pilih Hapus perlindungan penurunan skala.

AWS CLI

Untuk mengaktifkan perlindungan scale-in untuk sebuah instance

Gunakan [set-instance-protection](#) perintah berikut untuk mengaktifkan perlindungan skala masuk instance untuk instance yang ditentukan.

```
aws autoscaling set-instance-protection --instance-ids i-5f2e8a0d --auto-scaling-group-name my-asg --protected-from-scale-in
```

Untuk menonaktifkan perlindungan scale-in untuk sebuah instance

Gunakan perintah berikut ini untuk menonaktifkan perlindungan penurunan skala instance untuk instance tertentu.

```
aws autoscaling set-instance-protection --instance-ids i-5f2e8a0d --auto-scaling-group-name my-asg --no-protected-from-scale-in
```

Note

Ingat, perlindungan penskalaan instans tidak menjamin bahwa instans tidak akan dihentikan jika terjadi kesalahan manusia—misalnya, jika seseorang menghentikan instans secara manual menggunakan konsol Amazon EC2 atau AWS CLI. Untuk melindungi instans Anda dari penghentian yang tidak disengaja, Anda dapat menggunakan perlindungan penghentian Amazon EC2. Namun, bahkan dengan perlindungan terminasi dan perlindungan skala masuk instance diaktifkan, data yang disimpan ke penyimpanan instance dapat hilang jika pemeriksaan kesehatan menentukan bahwa suatu instance tidak sehat atau jika grup itu sendiri secara tidak sengaja dihapus. Seperti halnya lingkungan apa pun, praktik terbaik adalah sering mencadangkan data Anda, atau kapan pun sesuai dengan persyaratan kelangsungan bisnis Anda.

Rancang aplikasi Anda di Amazon EC2 Auto Scaling untuk menangani penghentian instans dengan baik

Topik ini mencakup berbagai pendekatan yang dapat Anda ambil jika Anda memiliki aplikasi yang berjalan pada instance yang idealnya tidak boleh dihentikan secara tak terduga saat Amazon EC2 Auto Scaling merespons peristiwa scale-in.

Misalnya, Anda memiliki antrian Amazon SQS yang mengumpulkan pesan masuk untuk pekerjaan yang berjalan lama. Ketika pesan baru tiba, sebuah instance dalam grup Auto Scaling mengambil pesan dan mulai memprosesnya. Setiap pesan membutuhkan waktu 3 jam untuk diproses. Seiring bertambahnya jumlah pesan, instance baru secara otomatis ditambahkan ke grup Auto Scaling.

Karena jumlah pesan berkurang, instance yang ada akan dihentikan secara otomatis. Dalam hal ini, Auto Scaling Amazon EC2 harus memutuskan instance mana yang akan dihentikan. Secara default, ada kemungkinan bahwa Amazon EC2 Auto Scaling dapat menghentikan instance yang 2,9 jam untuk memproses pekerjaan selama 3 jam, bukan instance yang saat ini tidak digunakan. Untuk menghindari masalah dengan penghentian yang tidak terduga saat menggunakan Auto Scaling Amazon EC2, Anda harus mendesain aplikasi untuk menanggapi skenario ini.

Anda dapat menggunakan fitur-fitur berikut untuk mencegah grup Auto Scaling menghentikan instans yang belum siap untuk dihentikan, atau menghentikan instans terlalu cepat bagi mereka untuk menyelesaikan pekerjaan yang ditetapkan. Ketiga fitur ini dapat digunakan dalam kombinasi atau secara terpisah.

Daftar Isi

- [Proteksi penurunan skala instance](#)
- [Kebijakan penghentian kustom](#)
- [Pengait siklus hidup penghentian](#)

Important

Saat mendesain aplikasi Anda di Amazon EC2 Auto Scaling untuk menangani penghentian instans dengan baik, ingatlah poin-poin ini.

- Jika instans tidak sehat, Auto Scaling Amazon EC2 akan menggantikannya terlepas dari fitur yang Anda gunakan (kecuali jika Anda menanggukkan `ReplaceUnhealthy` prosesnya). Anda dapat menggunakan kait siklus hidup untuk memungkinkan aplikasi dimatikan dengan anggun atau menyalin data apa pun yang perlu Anda pulihkan sebelum instance dihentikan.
- Pengait siklus hidup terminasi tidak dijamin akan berjalan atau selesai sebelum instance dihentikan. Jika sesuatu gagal, Amazon EC2 Auto Scaling masih menghentikan instance.

Proteksi penurunan skala instance

Anda dapat menggunakan perlindungan skala dalam instance dalam banyak situasi di mana penghentian instance adalah tindakan penting yang harus ditolak secara default, dan hanya diizinkan secara eksplisit untuk instance tertentu. Misalnya, saat menjalankan beban kerja kontainer, biasanya ingin melindungi semua instance dan menghapus perlindungan hanya untuk instance tanpa tugas

saat ini atau yang dijadwalkan. Layanan seperti Amazon ECS telah membangun integrasi dengan perlindungan skala instans ke dalam produk mereka.

Anda dapat mengaktifkan perlindungan scale-in pada grup Auto Scaling untuk menerapkan perlindungan scale-in ke instance saat dibuat dan mengaktifkannya untuk instance yang ada. Ketika sebuah instance tidak memiliki pekerjaan lagi yang harus dilakukan, itu dapat mematikan perlindungan. Instans dapat melanjutkan polling untuk pekerjaan baru dan mengaktifkan kembali perlindungan ketika ada pekerjaan baru yang ditugaskan.

Aplikasi dapat mengatur perlindungan baik dari bidang kontrol terpusat yang mengelola apakah sebuah instance dapat dihentikan atau tidak, atau dari instance itu sendiri. Namun, armada besar dapat mengalami masalah pelambatan jika sejumlah besar instance terus beralih perlindungan skala-in mereka.

Untuk informasi selengkapnya, lihat [Gunakan perlindungan skala dalam instance](#).

Kebijakan penghentian kustom

Seperti perlindungan penskalaan instance, kebijakan penghentian khusus membantu Anda mencegah grup Auto Scaling menghentikan instance tertentu.

Secara default, grup Auto Scaling Anda menggunakan kebijakan penghentian default untuk menentukan instance mana yang dihentikan terlebih dahulu. Jika ingin lebih mengontrol instance mana yang dihentikan terlebih dahulu, Anda dapat menerapkan kebijakan penghentian kustom Anda sendiri menggunakan fungsi Lambda. Auto Scaling Amazon EC2 memanggil fungsi kapan pun harus memutuskan instance mana yang akan dihentikan. Ini hanya akan mengakhiri sebuah instance yang dikembalikan oleh fungsi. Jika fungsi error, waktu habis, atau menghasilkan daftar kosong, Auto Scaling Amazon EC2 tidak menghentikan instans.

Kebijakan penghentian kustom berguna jika diketahui ketika sebuah instance cukup berlebihan atau kurang dimanfaatkan sehingga dapat dihentikan. Untuk mendukung hal ini, Anda perlu mengimplementasikan aplikasi Anda dengan bidang kontrol yang memantau beban kerja di seluruh grup. Dengan begitu, jika sebuah instance masih memproses pekerjaan, fungsi Lambda tahu untuk tidak memasukkannya.

Untuk informasi selengkapnya, lihat [Membuat kebijakan penghentian kustom dengan Lambda](#).

Pengait siklus hidup penghentian

Pengait siklus hidup terminasi memperpanjang umur instance yang sudah dipilih untuk penghentian. Ini memberikan waktu ekstra untuk menyelesaikan semua pesan atau permintaan yang saat ini

ditugaskan ke instance, atau untuk menyimpan kemajuan dan mentransfer pekerjaan ke instance lain.

Untuk banyak beban kerja, pengait siklus hidup mungkin cukup untuk mematikan aplikasi dengan anggun pada instance yang dipilih untuk penghentian. Ini adalah pendekatan upaya terbaik dan tidak dapat digunakan untuk mencegah penghentian jika ada kegagalan.

Untuk menggunakan kait siklus hidup, Anda perlu tahu kapan sebuah instance dipilih untuk dihentikan. Anda memiliki dua cara untuk mengetahui hal ini:

Opsi	Deskripsi	Paling baik digunakan untuk	Tautan ke dokumentasi
Di dalam instance	Layanan Metadata Instance (IMDS) adalah titik akhir aman yang dapat Anda polling untuk status instance langsung dari instance. Jika metadata kembali dengan <code>Terminate</code> , maka instance Anda dijadwalkan untuk dihentikan.	Aplikasi di mana Anda harus melakukan tindakan pada instance sebelum instance dihentikan.	Mengambil status siklus hidup target
Di luar contoh	Ketika sebuah instance berakhir, pemberitahuan acara dihasilkan. Anda dapat membuat aturan menggunakan Amazon EventBridge, Amazon SQS, atau Amazon SNS untuk menangkap peristiwa ini, dan memanggil respons seperti dengan fungsi Lambda.	Aplikasi yang perlu mengambil tindakan di luar instance.	Konfigurasi target notifikasi

Untuk menggunakan kait siklus hidup, Anda juga perlu tahu kapan instance Anda siap untuk dihentikan sepenuhnya. Auto Scaling Amazon EC2 tidak akan memberi tahu Amazon EC2 untuk menghentikan instans hingga menerima panggilan atau batas waktu berlalu, mana [CompleteLifecycleAction](#) yang terjadi lebih dulu.

Secara default, sebuah instance dapat terus berjalan selama satu jam (batas waktu detak jantung) karena pengait siklus hidup penghentian. Anda dapat mengonfigurasi batas waktu default jika satu jam tidak cukup waktu untuk menyelesaikan tindakan siklus hidup. Saat tindakan siklus hidup sedang berlangsung, Anda dapat memperpanjang batas waktu dengan [RecordLifecycleActionHeartbeat](#) panggilan API.

Untuk informasi selengkapnya, lihat [Hook siklus aktif Amazon EC2 Auto Scaling](#).

Menangguhkan dan melanjutkan proses Auto Scaling Amazon EC2

Topik ini menjelaskan cara menangguhkan dan kemudian melanjutkan satu atau beberapa proses untuk grup Auto Scaling Anda untuk menonaktifkan sementara operasi tertentu.

Proses penangguhan dapat berguna ketika Anda perlu menyelidiki atau memecahkan masalah tanpa gangguan dari kebijakan penskalaan atau tindakan terjadwal. Ini juga membantu mencegah Auto Scaling Amazon EC2 menandai instans yang tidak sehat dan menggantinya saat Anda membuat perubahan pada grup Auto Scaling Anda.

Topik

- [Jenis proses](#)
- [Pertimbangan](#)
- [Menangguhkan proses](#)
- [Lanjutkan proses](#)
- [Bagaimana proses yang ditangguhkan memengaruhi proses lain](#)

Note

Selain penangguhan yang Anda mulai, Amazon EC2 Auto Scaling juga dapat menangguhkan proses untuk grup Auto Scaling yang berulang kali gagal meluncurkan instance. Ini dikenal sebagai penangguhan administratif. Penangguhan administratif paling umum berlaku untuk grup Auto Scaling yang telah mencoba meluncurkan instance selama lebih dari 24 jam namun belum berhasil meluncurkan instance apa pun. Anda dapat melanjutkan proses yang ditangguhkan oleh Amazon EC2 Auto Scaling karena alasan administratif.

Jenis proses

Fitur suspend-resume mendukung proses berikut:

- **Launch**— Menambahkan instans ke grup Auto Scaling saat grup diskalakan, atau saat Amazon EC2 Auto Scaling memilih untuk meluncurkan instans karena alasan lain, seperti saat grup menambahkan instance ke kolam hangat.
- **Terminate**— Menghapus instans dari grup Auto Scaling saat grup melakukan penskalaan, atau saat Amazon EC2 Auto Scaling memilih untuk menghentikan instans karena alasan lain, seperti saat instans dihentikan karena melebihi durasi masa pakai maksimumnya atau gagal dalam pemeriksaan kesehatan.
- **AddToLoadBalancer**— Menambahkan instance ke grup target penyeimbang beban terlampir atau Classic Load Balancer saat diluncurkan. Untuk informasi selengkapnya, lihat [Gunakan Elastic Load Balancing untuk mendistribusikan lalu lintas di seluruh instans dalam grup Auto Scaling](#).
- **AlarmNotification**— Menerima pemberitahuan dari CloudWatch alarm yang terkait dengan kebijakan penskalaan dinamis. Untuk informasi selengkapnya, lihat [Penskalaan dinamis untuk Amazon EC2 Auto Scaling](#).
- **AZRebalance**— Menyeimbangkan jumlah instans EC2 dalam grup secara merata di semua Availability Zone yang ditentukan saat grup menjadi tidak seimbang, misalnya, ketika Availability Zone yang sebelumnya tidak tersedia kembali ke keadaan sehat. Untuk informasi selengkapnya, lihat [Aktivitas rebalancing](#).
- **HealthCheck**— Memeriksa kesehatan instans dan menandai instance sebagai tidak sehat jika Amazon EC2 atau Elastic Load Balancing memberi tahu Amazon EC2 Auto Scaling bahwa instans tersebut tidak sehat. Proses ini dapat mengesampingkan status kesehatan suatu instance yang Anda atur secara manual. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).
- **InstanceRefresh**— Mengakhiri dan mengganti instance menggunakan fitur penyegaran instans. Untuk informasi selengkapnya, lihat [Menggunakan penyegaran instans untuk memperbarui instance dalam grup Auto Scaling](#).
- **ReplaceUnhealthy**— Mengakhiri instance yang ditandai sebagai tidak sehat dan kemudian membuat instance baru untuk menggantikannya. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).
- **ScheduledActions**— Melakukan tindakan penskalaan terjadwal yang Anda buat atau yang dibuat untuk Anda saat Anda membuat rencana penskalaan dan mengaktifkan AWS Auto Scaling

penskalaan prediktif. Untuk informasi selengkapnya, lihat [Skala terjadwal untuk Amazon EC2 Auto Scaling](#).

Pertimbangan

Pertimbangkan hal berikut sebelum menanggukhan proses:

- Penanggukan AlarmNotification memungkinkan Anda menghentikan sementara kebijakan pelacakan target, langkah, dan penskalaan sederhana grup tanpa menghapus kebijakan penskalaan atau alarm terkait. CloudWatch Untuk menghentikan sementara kebijakan penskalaan individu, lihat [Menonaktifkan kebijakan penskalaan untuk grup Auto Scaling](#).
- Anda dapat memilih untuk menanggukhan HealthCheck dan ReplaceUnhealthy proses untuk me-reboot instans tanpa Auto Scaling Amazon EC2 menghentikan instans berdasarkan pemeriksaan kesehatannya. Namun, jika Anda memerlukan Auto Scaling Amazon EC2 untuk terus melakukan pemeriksaan kesehatan pada instans yang tersisa, gunakan fitur siaga sebagai gantinya. Untuk informasi selengkapnya, lihat [Hapus instans sementara dari grup Auto Scaling](#).
- Jika Anda menanggukhan Terminate proses Launch dan, atau AZRebalance, dan kemudian Anda membuat perubahan pada grup Auto Scaling, misalnya, dengan melepaskan instance atau mengubah Availability Zone yang ditentukan, grup Anda dapat menjadi tidak seimbang antara Availability Zone. Jika itu terjadi, setelah Anda melanjutkan proses yang ditanggukhan, Auto Scaling Amazon EC2 secara bertahap mendistribusikan ulang instans secara merata di antara Availability Zone.
- Jika Anda menanggukhan Terminate proses, Anda masih dapat memaksa instance untuk dihentikan dengan menggunakan [delete-auto-scaling-group](#) perintah dengan opsi hapus paksa.
- Menanggukhan Terminate proses hanya berlaku untuk contoh yang saat ini berada di negara bagian. InService Ini tidak mencegah penghentian contoh di negara bagian lain, seperti Pending, atau contoh yang gagal dilanjutkan dengan benar dari siaga.
- RemoveFromLoadBalancerLowPriorityProses ini dapat diabaikan ketika ada dalam panggilan untuk menggambarkan grup Auto Scaling menggunakan AWS CLI atau SDK. Proses ini tidak digunakan lagi dan dipertahankan hanya untuk kompatibilitas mundur.

Menanggukhan proses

Untuk menanggukhan proses grup Auto Scaling, gunakan salah satu metode berikut:

Console

Untuk menanggihkan proses

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Detail, pilih Konfigurasi lanjutan, Edit.
4. Untuk Proses yang ditanggihkan, pilih proses yang akan ditanggihkan.
5. Pilih Perbarui.

AWS CLI

Gunakan perintah [suspend-processes](#) berikut untuk menanggihkan proses individual.

```
aws autoscaling suspend-processes --auto-scaling-group-name my-asg --scaling-processes HealthCheck ReplaceUnhealthy
```

Untuk menanggihkan semua proses, hilangkan `--scaling-processes` opsi, sebagai berikut.

```
aws autoscaling suspend-processes --auto-scaling-group-name my-asg
```

Lanjutkan proses

Untuk melanjutkan proses yang ditanggihkan untuk grup Auto Scaling, gunakan salah satu metode berikut:

Console

Untuk melanjutkan proses penangguhan

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Detail, pilih Konfigurasi lanjutan, Edit.
4. Untuk proses yang ditangguhkan, hapus proses yang ditangguhkan.
5. Pilih Perbarui.

AWS CLI

Untuk melanjutkan proses yang ditangguhkan, gunakan perintah [resume-process](#) berikut.

```
aws autoscaling resume-processes --auto-scaling-group-name my-asg --scaling-processes HealthCheck
```

Untuk melanjutkan semua proses yang ditangguhkan, hilangkan `--scaling-processes` opsi, sebagai berikut.

```
aws autoscaling resume-processes --auto-scaling-group-name my-asg
```

Bagaimana proses yang ditangguhkan memengaruhi proses lain

Bagian berikut menjelaskan apa yang terjadi ketika proses yang berbeda ditangguhkan secara individual.

Topik

- [Launchditangguhkan](#)
- [Terminateditangguhkan](#)
- [AddToLoadBalancerditangguhkan](#)
- [AlarmNotificationditangguhkan](#)
- [AZRebalanceditangguhkan](#)
- [HealthCheckditangguhkan](#)
- [InstanceRefreshditangguhkan](#)
- [ReplaceUnhealthyditangguhkan](#)
- [ScheduledActionsditangguhkan](#)
- [Pertimbangan tambahan](#)

Launch ditangguhkan

- AlarmNotification masih aktif, tetapi grup Auto Scaling Anda tidak dapat memulai aktivitas scale-out untuk alarm yang dilanggar.
- ScheduledActions aktif, tetapi grup Auto Scaling Anda tidak dapat memulai aktivitas penskalaan untuk tindakan terjadwal apa pun yang terjadi.
- AZRebalance berhenti menyeimbangkan ulang grup.
- ReplaceUnhealthy terus menghentikan instance yang tidak sehat, tetapi tidak meluncurkan pengganti. Saat Anda melanjutkan Launch proses, Auto Scaling Amazon EC2 segera menggantikan instans yang dihentikan selama waktu yang ditangguhkan. Launch
- InstanceRefresh tidak menggantikan instance.

Terminated ditangguhkan

- AlarmNotification masih aktif, tetapi grup Auto Scaling Anda tidak dapat memulai aktivitas penskalaan untuk alarm yang dilanggar.
- ScheduledActions aktif, tetapi grup Auto Scaling Anda tidak dapat memulai aktivitas penskalaan untuk tindakan terjadwal apa pun yang terjadi.
- AZRebalance masih aktif tetapi tidak berfungsi dengan baik. Ini dapat meluncurkan instance baru tanpa menghentikan instance lama. Hal ini dapat menyebabkan grup Auto Scaling Anda tumbuh hingga 10 persen lebih besar dari ukuran maksimumnya, karena hal ini dimungkinkan sementara temporer selama aktivitas rebalancing. Grup Auto Scaling Anda dapat tetap di atas ukuran maksimumnya hingga Anda melanjutkan proses Terminate.
- ReplaceUnhealthy tidak aktif tetapi tidakHealthCheck. Ketika Terminate dilanjutkan, ReplaceUnhealthy proses segera mulai berjalan. Jika ada instance yang ditandai sebagai tidak sehat saat Terminate ditangguhkan, perangkat segera diganti.
- InstanceRefresh tidak menggantikan instance.

AddToLoadBalancer ditangguhkan

- Auto Scaling Amazon EC2 meluncurkan instans tetapi tidak menambahkannya ke grup target penyeimbang beban atau Classic Load Balancer. Saat Anda melanjutkan AddToLoadBalancer proses, itu melanjutkan penambahan instance ke penyeimbang beban saat diluncurkan. Namun demikian, hal tersebut tidak menambahkan instance yang diluncurkan saat proses ini ditangguhkan. Anda harus mendaftarkan instance tersebut secara manual.

AlarmNotification ditangguhkan

- Auto Scaling Amazon EC2 tidak menjalankan kebijakan penskalaan saat CloudWatch ambang alarm dilanggar. Saat Anda melanjutkan `AlarmNotification`, Amazon EC2 Auto Scaling mempertimbangkan kebijakan dengan ambang batas alarm yang saat ini ditembus.

AZRebalance ditangguhkan

- Auto Scaling Amazon EC2 tidak mencoba mendistribusikan ulang instans setelah kejadian tertentu. Namun, jika terjadi peristiwa peningkatan atau penurunan skala, proses penskalaan masih mencoba menyeimbangkan Zona Ketersediaan. Sebagai contoh, selama peningkatan skala, proses penskalaan ini meluncurkan instance di Zona Ketersediaan dengan instance paling sedikit. Jika grup menjadi tidak seimbang saat `AZRebalance` ditangguhkan dan Anda melanjutkannya, Amazon EC2 Auto Scaling mencoba menyeimbangkan kembali grup. Panggilan pertama adalah untuk `Launch` dan kemudian `Terminate`.

HealthCheck ditangguhkan

- Amazon EC2 Auto Scaling berhenti menandai instance yang tidak sehat sebagai hasil dari pemeriksaan kesehatan EC2 dan Elastic Load Balancing. Pemeriksaan kesehatan khusus Anda terus berfungsi dengan baik. Setelah Anda menangguhkan `HealthCheck`, jika perlu, Anda dapat mengatur kondisi kesehatan instance dalam grup secara manual dan `ReplaceUnhealthy` menggantinya.

InstanceRefresh ditangguhkan

- Auto Scaling Amazon EC2 berhenti mengganti instans sebagai hasil penyegaran instans. Jika ada penyegaran instance yang sedang berlangsung, ini menghentikan operasi tanpa membatalkannya.

ReplaceUnhealthy ditangguhkan

- Amazon EC2 Auto Scaling menghentikan penggantian instance yang ditandai sebagai tidak sehat. Instance yang gagal dalam pemeriksaan kesehatan EC2 atau Elastic Load Balancing masih ditandai sebagai tidak sehat. Segera setelah Anda melanjutkan proses `ReplaceUnhealthy`, Amazon EC2 Auto Scaling menggantikan instance yang ditandai tidak sehat sementara proses

ini ditangguhkan. `ReplaceUnhealthy` prosesnya memanggil `Terminate` terlebih dahulu dan kemudian `Launch`.

Scheduled Actions ditangguhkan

- Auto Scaling Amazon EC2 tidak menjalankan tindakan terjadwal yang dijadwalkan untuk berjalan selama periode penangguhan. Saat Anda melanjutkan `Scheduled Actions`, Amazon EC2 Auto Scaling hanya mempertimbangkan tindakan terjadwal yang waktu yang dijadwalkan belum berlalu.

Pertimbangan tambahan

Selain itu, ketika `Launch` atau `Terminate` ditangguhkan, fitur berikut mungkin tidak berfungsi dengan benar:

- Masa pakai instans maksimum — Saat `Launch` atau `Terminate` ditangguhkan, fitur masa pakai instans maksimum tidak dapat menggantikan instance apa pun.
- Interupsi Instans Spot — Jika `Terminate` ditangguhkan dan grup Auto Scaling Anda memiliki Instans Spot, mereka masih dapat dihentikan jika kapasitas Spot tidak lagi tersedia. Meskipun `Launch` ditangguhkan, Auto Scaling Amazon EC2 tidak dapat meluncurkan instans pengganti dari kumpulan Instans Spot lain atau dari kumpulan Instans Spot yang sama saat tersedia lagi.
- Penyeimbangan Kembali Kapasitas — Jika `Terminate` ditangguhkan dan Anda menggunakan Penyeimbangan Kembali Kapasitas untuk menangani interupsi Instans Spot, layanan Spot Amazon EC2 masih dapat menghentikan instans jika kapasitas Spot tidak lagi tersedia. Jika `Launch` ditangguhkan, Auto Scaling Amazon EC2 tidak dapat meluncurkan instans pengganti dari kumpulan Instans Spot lain atau dari kumpulan Instans Spot yang sama saat tersedia lagi.
- Melampirkan dan melepaskan instance — Saat **Launch** dan **Terminate** ditangguhkan, Anda dapat melepaskan instance yang dilampirkan ke grup Auto Scaling, tetapi saat `Launch` ditangguhkan, Anda tidak dapat melampirkan instance baru ke grup.
- Instans siaga — Ketika `Launch` dan `Terminate` ditangguhkan, Anda dapat menempatkan instance di `Standby` negara bagian, tetapi saat `Launch` ditangguhkan, Anda tidak dapat mengembalikan instance di `Standby` negara bagian ke layanan.

Pantau grup Penskalaan Otomatis

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja Amazon EC2 Auto Scaling dan AWS CloudSolusi. AWS menyediakan alat pemantauan berikut untuk menonton Amazon EC2 Auto Scaling, melaporkan ketika ada sesuatu yang salah, dan mengambil tindakan otomatis bila diperlukan:

Pemeriksaan kondisi

Amazon EC2 Auto Scaling secara berkala melakukan pemeriksaan kesehatan pada instans dalam grup Auto Scaling Anda. Jika instans tidak lulus pemeriksaan kesehatannya, instans ditandai tidak sehat dan akan dihentikan sementara Amazon EC2 Auto Scaling meluncurkan instans baru untuk menggantikannya. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

AWS Health Dashboard

Yang AWS Health Dashboard menampilkan informasi, dan juga menyediakan pemberitahuan yang dipanggil oleh perubahan kesehatan AWS sumber daya. Informasi ini disajikan dalam dua cara: di dasbor yang menampilkan peristiwa terbaru dan mendatang yang diatur berdasarkan kategori, dan dalam catatan peristiwa lengkap yang menampilkan semua peristiwa dari 90 hari terakhir. Untuk informasi selengkapnya, lihat [AWS Health Dashboard pemberitahuan untuk Auto Scaling Amazon EC2](#).

CloudTrail

Dengan AWS CloudTrail, Anda dapat melacak panggilan yang dilakukan ke Amazon EC2 Auto Scaling API oleh atau atas nama Anda Akun AWS. CloudTrail menyimpan informasi dalam file log di bucket Amazon S3 yang Anda tentukan. Anda dapat menggunakan file log ini untuk memantau aktivitas grup Auto Scaling Anda. Log termasuk permintaan yang dibuat, sumber alamat IP asal permintaan, siapa yang meminta, kapan permintaan dibuat, dan seterusnya. Untuk informasi selengkapnya, lihat [Log panggilan API Auto Scaling Amazon EC2 dengan AWS CloudTrail](#).

Pengumpulan log untuk instans Amazon EC2 Anda

Anda dapat menggunakan CloudWatch untuk mengumpulkan log dari sistem operasi untuk instans EC2 Anda. Untuk informasi lebih lanjut, lihat [Kumpulkan metrik dan log dari instans Amazon EC2 dan server lokal dengan CloudWatch](#) dan [Lihat data log yang dikirim ke CloudWatch Log](#) di dalam Amazon CloudWatch Panduan Pengguna.

Untuk informasi tentang AWS layanan yang dapat membantu Anda mencatat dan mengumpulkan data tentang beban kerja Anda, lihat [Panduan logging dan monitoring untuk pemilik aplikasi](#) membimbing AWS Panduan Preskriptif.

Amazon CloudWatch

Amazon CloudWatch membantu Anda menganalisis log dan, secara real time, memantau metrik Anda AWS sumber daya dan aplikasi host. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan. Misalnya, Anda dapat diberi tahu saat aktivitas jaringan tiba-tiba lebih tinggi atau lebih rendah dari nilai yang diharapkan metrik. Untuk informasi selengkapnya tentang penggunaan layanan ini untuk memantau metrik grup dan instans Penskalaan Otomatis, lihat [Pantau CloudWatch metrik untuk grup dan instans Auto Scaling](#).

CloudWatch juga melacak AWS Metrik penggunaan API untuk Amazon EC2 Auto Scaling. Anda dapat menggunakan metrik ini untuk mengonfigurasi alarm yang memberi tahu Anda saat volume panggilan API melanggar ambang batas yang Anda tentukan. Untuk informasi lebih lanjut, lihat [AWS metrik penggunaan](#) di dalam Amazon CloudWatch Panduan Pengguna.

AWS Compute Optimizer

Compute Optimizer menyediakan rekomendasi instans Amazon EC2 yang dapat membantu Anda memutuskan apakah akan pindah ke jenis instans baru. Ini menganalisis apakah jenis instans grup Auto Scaling optimal dan menghasilkan rekomendasi untuk mengurangi biaya dan meningkatkan kinerja beban kerja Anda. Untuk informasi selengkapnya, lihat [Gunakan AWS Compute Optimizer untuk mendapatkan rekomendasi untuk jenis instans untuk grup Auto Scaling](#).

Amazon EventBridge

Amazon EventBridge adalah layanan bus acara tanpa server yang memudahkan untuk menghubungkan aplikasi Anda dengan data dari berbagai sumber. EventBridge memberikan aliran data real-time dari aplikasi Anda sendiri, Software-as-a-Layanan (SaaS) aplikasi, dan AWS layanan dan rute yang data ke target seperti Lambda. Hal ini memungkinkan Anda memantau peristiwa yang terjadi di layanan, dan membangun arsitektur berbasis peristiwa. Untuk informasi selengkapnya, lihat [Gunakan EventBridge untuk menangani peristiwa Auto Scaling](#).

AWS Security Hub

Gunakan [AWS Security Hub](#) untuk memantau penggunaan Amazon EC2 Auto Scaling karena terkait dengan praktik terbaik keamanan. Hub Keamanan menggunakan detektif kontrol keamanan untuk mengevaluasi konfigurasi sumber daya dan standar keamanan untuk membantu Anda mematuhi berbagai kerangka kerja kepatuhan. Untuk informasi selengkapnya tentang penggunaan Hub Keamanan untuk mengevaluasi sumber daya Amazon EC2 Auto Scaling, lihat [Kontrol Penskalaan Otomatis Amazon EC2](#) di dalam AWS Security Hub Panduan Pengguna.

Amazon Simple Notification Service

Anda dapat mengonfigurasi grup Auto Scaling untuk mengirim notifikasi Amazon SNS ketika Amazon EC2 Auto Scaling meluncurkan atau mengakhiri instans. Untuk informasi selengkapnya, lihat [Opsis pemberitahuan Amazon SNS untuk Auto Scaling Amazon EC2](#).

Health memeriksa instans dalam grup Auto Scaling

Auto Scaling Amazon EC2 terus memantau status kesehatan instans dalam grup Auto Scaling untuk mempertahankan kapasitas yang diinginkan.

Semua instance dalam grup Auto Scaling dimulai dengan Healthy status. Instance diasumsikan sehat, kecuali Amazon EC2 Auto Scaling menerima notifikasi bahwa instance tersebut tidak sehat. Ini dapat menerima pemberitahuan dari berbagai sumber ketika sebuah instance menjadi tidak sehat dan perlu diganti. Sumber-sumber ini meliputi:

- Amazon EC2
- Penyeimbang Beban Elastis
- Kisi VPC
- Pemeriksaan kesehatan khusus yang Anda tentukan

Jika Auto Scaling Amazon EC2 menentukan bahwa InService instans tidak sehat, instans menggantinya dengan instans baru untuk mempertahankan kapasitas grup yang diinginkan. Instans baru diluncurkan menggunakan pengaturan grup Auto Scaling saat ini dan templat peluncuran atau konfigurasi peluncuran terkait.

Instance yang tidak sehat juga dapat terjadi ketika sebuah instance dihentikan secara tak terduga, seperti dari interupsi Instans Spot atau penghentian manual oleh pengguna. Sekali lagi, Auto

Scaling Amazon EC2 akan secara otomatis meluncurkan instance pengganti dalam kasus ini untuk mempertahankan kapasitas yang diinginkan.

Daftar Isi

- [Tentang pemeriksaan kesehatan untuk grup Auto Scaling Anda](#)
- [Lihat alasan kegagalan pemeriksaan kesehatan](#)
- [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#)

Tentang pemeriksaan kesehatan untuk grup Auto Scaling Anda

Topik ini memberikan gambaran umum tentang jenis pemeriksaan kesehatan default dan tersedia dan menjelaskan cara kerjanya.

Daftar Isi

- [Jenis pemeriksaan kesehatan](#)
- [Pemeriksaan kesehatan Amazon EC2](#)
- [Pemeriksaan kondisi Elastic Load Balancing](#)
- [Pemeriksaan kesehatan VPC Lattice](#)
- [Bagaimana Auto Scaling Amazon EC2 meminimalkan waktu henti](#)
- [Pertimbangan pemeriksaan kesehatan](#)
- [Pemeriksaan kesehatan khusus](#)
- [Sumber daya terkait](#)

Jenis pemeriksaan kesehatan

Auto Scaling Amazon EC2 dapat menentukan status kesehatan instans dengan menggunakan satu atau beberapa pemeriksaan kesehatan berikut:

Jenis pemeriksaan kesehatan	Apa yang diperiksa
Pemeriksaan status Amazon EC2 dan acara terjadwal	<ul style="list-style-type: none"> • Memeriksa bahwa instance sedang berjalan • Memeriksa masalah perangkat keras atau perangkat lunak yang mendasari yang mungkin mengganggu instance

Jenis pemeriksaan kesehatan	Apa yang diperiksa
	Ini adalah jenis pemeriksaan kesehatan default untuk grup Auto Scaling.
Pemeriksaan kondisi Elastic Load Balancing	<ul style="list-style-type: none"> • Memeriksa apakah penyeimbang beban melaporkan instance sebagai sehat, mengonfirmasi apakah instance tersedia untuk menangani permintaan <p>Untuk menjalankan jenis pemeriksaan kesehatan ini, Anda harus mengaktifkannya untuk grup Auto Scaling Anda.</p>
Pemeriksaan kesehatan VPC Lattice	<ul style="list-style-type: none"> • Memeriksa apakah VPC Lattice melaporkan instance sebagai sehat, mengonfirmasi apakah instance tersedia untuk menangani permintaan <p>Untuk menjalankan jenis pemeriksaan kesehatan ini, Anda harus mengaktifkannya untuk grup Auto Scaling Anda.</p>
Pemeriksaan kesehatan khusus	<ul style="list-style-type: none"> • Memeriksa masalah lain yang mungkin mengindikasikan masalah kesehatan misalnya, menurut pemeriksaan kesehatan khusus Anda

Pemeriksaan kesehatan Amazon EC2

Setelah instance diluncurkan, itu dilampirkan ke grup Auto Scaling dan memasuki InService status. Untuk informasi selengkapnya tentang status siklus hidup yang berbeda untuk instance dalam grup Auto Scaling, lihat. [Siklus aktif instance Amazon EC2 Auto Scaling](#)

Auto Scaling Amazon EC2 secara berkala memeriksa status kesehatan semua instans dalam grup Auto Scaling untuk memastikan bahwa mereka berjalan dan dalam kondisi baik.

Pemeriksaan status

Auto Scaling Amazon EC2 menggunakan hasil pemeriksaan status instans Amazon EC2 dan pemeriksaan status sistem untuk menentukan status kesehatan suatu instans. Jika instans berada dalam status Amazon EC2 selain `running`, atau jika statusnya untuk pemeriksaan

status menjadi `impaired`, Auto Scaling Amazon EC2 menganggap instans tidak sehat dan menggantikannya. Hal ini termasuk ketika instance tersebut memiliki salah satu status berikut:

- `stopping`
- `stopped`
- `shutting-down`
- `terminated`

Pemeriksaan status Amazon EC2 tidak memerlukan konfigurasi khusus dan selalu diaktifkan. Untuk informasi lebih lanjut, lihat [Jenis pemeriksaan status](#) di Panduan Pengguna Amazon EC2 untuk Instance Linux.

Important

Auto Scaling Amazon EC2 memungkinkan pemeriksaan status gagal sesekali, tanpa mengambil tindakan apa pun. Ketika pemeriksaan status gagal, Auto Scaling Amazon EC2 menunggu beberapa menit AWS untuk memperbaiki masalah. Itu tidak segera menandai instance `Unhealthy` ketika statusnya untuk pemeriksaan status menjadi `impaired`. Namun, jika Auto Scaling Amazon EC2 mendeteksi bahwa instans tidak lagi dalam `running` keadaan, situasi ini diperlakukan sebagai kegagalan langsung. Dalam hal ini, segera menandai instance `Unhealthy` dan menggantikannya.

Peristiwa terjadwal

Amazon EC2 kadang-kadang dapat menjadwalkan acara pada instans Anda untuk dijalankan setelah stempel waktu tertentu. Untuk informasi selengkapnya, lihat [Acara terjadwal untuk instans Anda](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Jika salah satu instans Anda terpengaruh oleh peristiwa terjadwal, Auto Scaling Amazon EC2 menganggap instans tersebut tidak sehat dan menggantikannya. Instance tidak mulai berakhir sampai tanggal dan waktu yang ditentukan dalam stempel waktu tercapai.

Pemeriksaan kondisi Elastic Load Balancing

Saat Anda mengaktifkan pemeriksaan kesehatan Elastic Load Balancing untuk grup Auto Scaling, Auto Scaling Amazon EC2 dapat menggunakan hasil pemeriksaan kesehatan tersebut untuk menentukan status kesehatan suatu instans.

Sebelum Anda dapat mengaktifkan pemeriksaan kesehatan Elastic Load Balancing untuk grup Auto Scaling Anda, Anda harus melakukan hal berikut:

- Siapkan penyeimbang beban Elastic Load Balancing dan konfigurasi pemeriksaan kesehatan untuk digunakan untuk menentukan apakah instans Anda sehat.
- Pasang penyeimbang beban ke grup Auto Scaling Anda.

Setelah Anda menyelesaikan tindakan sebelumnya, hal berikut terjadi:

- Auto Scaling Amazon EC2 mendaftarkan instans dalam grup Auto Scaling dengan penyeimbang beban.
- Setelah sebuah instance selesai mendaftar, ia memasuki InService status dan menjadi tersedia untuk digunakan dengan penyeimbang beban.

Secara default, Auto Scaling Amazon EC2 mengabaikan hasil pemeriksaan kesehatan Elastic Load Balancing. Namun, Anda dapat mengaktifkan pemeriksaan kesehatan ini untuk grup Auto Scaling Anda. Setelah Anda melakukannya, saat Elastic Load Balancing melaporkan instans terdaftar sebagai `Unhealthy`, Auto Scaling Amazon EC2 menandai instans `Unhealthy` tersebut pada pemeriksaan kesehatan berkala berikutnya dan menggantinya.

Jika pengurusan koneksi (penundaan deregistrasi) diaktifkan untuk penyeimbang beban Anda, Auto Scaling Amazon EC2 menunggu permintaan dalam penerbangan selesai atau batas waktu maksimum berakhir sebelum menghentikan instans yang tidak sehat.

Untuk mempelajari cara mengaktifkan pemeriksaan kesehatan Elastic Load Balancing untuk grup Auto Scaling Anda, lihat [Pasang penyeimbang beban Elastic Load Balancing ke grup Auto Scaling Anda](#)

Note

Saat Anda mengaktifkan pemeriksaan kesehatan Elastic Load Balancing untuk grup, Auto Scaling Amazon EC2 dapat menggantikan instans yang dilaporkan Elastic Load Balancing sebagai tidak sehat, tetapi hanya setelah penyeimbang beban dalam status `InService`. Untuk informasi selengkapnya, lihat [Verifikasi status lampiran penyeimbang beban Anda](#).

Pemeriksaan kesehatan VPC Lattice

Secara default, Amazon EC2 Auto Scaling mengabaikan hasil pemeriksaan kesehatan VPC Lattice. Anda dapat mengaktifkan pemeriksaan kesehatan ini secara opsional untuk grup Auto Scaling Anda. Setelah Anda melakukan ini, ketika VPC Lattice melaporkan instance terdaftar sebagai Amazon Unhealthy EC2 Auto Scaling menandai Unhealthy instans pada pemeriksaan kesehatan berkala berikutnya dan menggantikannya. Proses mendaftarkan instance dan kemudian memeriksa kesehatannya sama dengan cara kerja pemeriksaan kesehatan Elastic Load Balancing.

Untuk mempelajari cara mengaktifkan pemeriksaan kesehatan VPC Lattice untuk grup Auto Scaling Anda, lihat [Lampirkan grup target VPC Lattice ke grup Auto Scaling](#)

Note

Saat Anda mengaktifkan pemeriksaan kesehatan VPC Lattice untuk grup, Auto Scaling Amazon EC2 dapat menggantikan instance yang dilaporkan VPC Lattice sebagai tidak sehat, tetapi hanya setelah grup target dalam status. InService Untuk informasi selengkapnya, lihat [Verifikasi status lampiran grup target VPC Lattice](#).

Bagaimana Auto Scaling Amazon EC2 meminimalkan waktu henti

Secara default, penggantian pemeriksaan kesehatan memerlukan instance untuk dihentikan terlebih dahulu, yang mungkin mencegah permintaan baru diterima hingga instance baru diluncurkan.

Jika Auto Scaling Amazon EC2 menentukan bahwa instans apa pun tidak lagi berjalan (atau Unhealthy ditandai dengan perintah), itu segera [set-instance-health](#) menggantikannya. Namun, jika instans lain ternyata tidak sehat, Auto Scaling Amazon EC2 menggunakan pendekatan berikut untuk memulihkan dari kegagalan. Pendekatan ini meminimalkan waktu henti yang mungkin terjadi karena masalah sementara atau pemeriksaan kesehatan yang salah konfigurasi.

- Jika aktivitas penskalaan sedang berlangsung dan grup Auto Scaling Anda berada di bawah kapasitas yang diinginkan sebesar 10 persen atau lebih, Auto Scaling Amazon EC2 menunggu aktivitas penskalaan yang sedang berlangsung sebelum mengganti instans yang tidak sehat.
- Saat melakukan scaling, Amazon EC2 Auto Scaling menunggu instans lulus pemeriksaan kesehatan awal. Itu juga menunggu pemanasan instance default selesai untuk memastikan bahwa instance baru sudah siap.

- Setelah instans selesai pemanasan dan grup telah meningkat hingga di atas 90 persen dari kapasitas yang diinginkan, Amazon EC2 Auto Scaling menggantikan contoh yang tidak sehat sebagai berikut:
 - Auto Scaling Amazon EC2 hanya menggantikan hingga 10 persen dari kapasitas yang diinginkan grup sekaligus. Hal ini dilakukan sampai semua contoh yang tidak sehat diganti.
 - Saat mengganti instance, ia menunggu instance baru lulus pemeriksaan kesehatan awal. Itu juga menunggu pemanasan instance default selesai sebelum melanjutkan.

Note

Jika ukuran grup Auto Scaling cukup kecil sehingga nilai yang dihasilkan 10 persen kurang dari satu, Amazon EC2 Auto Scaling menggantikan instans yang tidak sehat satu per satu. Ini mungkin mengakibatkan beberapa downtime untuk grup.

Selain itu, jika semua instance dalam grup Auto Scaling dilaporkan tidak sehat oleh pemeriksaan kesehatan Elastic Load Balancing dan penyeimbang beban berada dalam InService status, Amazon EC2 Auto Scaling mungkin menandai lebih sedikit instans yang tidak sehat pada suatu waktu. Ini dapat menghasilkan lebih sedikit contoh yang diganti pada satu waktu daripada 10 persen yang diterapkan dalam skenario lain. Ini memberi Anda waktu untuk memperbaiki masalah tanpa Auto Scaling Amazon EC2 secara otomatis menghentikan seluruh grup.

Pertimbangan pemeriksaan kesehatan

Bagian ini berisi pertimbangan untuk pemeriksaan kesehatan Auto Scaling Amazon EC2.

- Jika Anda memerlukan sesuatu untuk terjadi pada instance yang sedang dihentikan, atau pada instance yang dimulai, Anda dapat menggunakan kait siklus hidup. Hook ini memungkinkan Anda melakukan tindakan kustom saat Amazon EC2 Auto Scaling meluncurkan atau menghentikan instans. Untuk informasi selengkapnya, lihat [Hook siklus aktif Amazon EC2 Auto Scaling](#).
- Auto Scaling Amazon EC2 tidak menyediakan cara untuk menghapus pemeriksaan status Amazon EC2 dan acara terjadwal dari pemeriksaan kesehatannya. Jika Anda tidak ingin instance diganti, kami sarankan Anda menanggihkan `ReplaceUnhealthy` dan `HealthCheck` memproses untuk grup Auto Scaling individual. Untuk informasi selengkapnya, lihat [Menanggihkan dan melanjutkan proses Auto Scaling Amazon EC2](#).

- Untuk mengatur kembali status kesehatan instans yang tidak sehat secara manual `Healthy`, Anda dapat mencoba menggunakan [set-instance-health](#) perintah tersebut. Jika Anda mendapatkan kesalahan, ini mungkin karena instance sudah berakhir. Umumnya, menyetel status kesehatan instance kembali `Healthy` dengan [set-instance-health](#) perintah hanya berguna dalam kasus di mana `ReplaceUnhealthy` proses atau `Terminate` proses ditangguhkan.
- Auto Scaling Amazon EC2 tidak melakukan pemeriksaan kesehatan pada instans yang ada di negara bagian. Standby Untuk informasi selengkapnya, lihat [Hapus instans sementara dari grup Auto Scaling](#).
- Ketika instance Anda dihentikan, setiap alamat Elastic IP terkait akan dilepaskan kaitannya dan tidak secara otomatis terkait dengan instance baru tersebut. Anda harus mengaitkan alamat IP Elastis secara manual dengan instans baru, atau melakukannya secara otomatis dengan solusi berbasis kait siklus hidup. Untuk informasi selengkapnya, lihat [Alamat IP Elastis](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Demikian pula, ketika instance Anda dihentikan, volume EBS terlampirnya terlepas (atau dihapus tergantung pada atribut volume). `DeleteOnTermination` Anda harus melampirkan volume EBS ini secara manual ke instans baru, atau melakukannya secara otomatis dengan solusi berbasis kait siklus hidup. Untuk informasi selengkapnya, lihat [Melampirkan volume Amazon EBS ke instans](#) di Panduan Pengguna Amazon EBS.

Pemeriksaan kesehatan khusus

Anda dapat menjalankan tugas deteksi kesehatan khusus secara opsional pada instans di grup Auto Scaling dan menyetel status kesehatan instans jika tugas `Unhealthy` gagal. Ini memperluas pemeriksaan kesehatan Anda dengan menggunakan kombinasi pemeriksaan kesehatan khusus, pemeriksaan status Amazon EC2, dan pemeriksaan kesehatan Elastic Load Balancing, jika diaktifkan.

Anda dapat mengirim informasi kesehatan instans langsung ke Amazon EC2 Auto Scaling menggunakan AWS CLI atau SDK. Contoh berikut menunjukkan cara menggunakan AWS CLI untuk mengonfigurasi status kesehatan suatu instans dan kemudian memverifikasi status kesehatan instans.

Gunakan [set-instance-health](#) perintah berikut untuk mengatur status kesehatan dari instance yang ditentukan ke `Unhealthy`.

```
aws autoscaling set-instance-health --instance-id i-1234567890abcdef0 --health-status Unhealthy
```

Secara default, perintah ini menghormati masa tenggang pemeriksaan kesehatan. Namun, Anda dapat mengesampingkan perilaku ini dan tidak menghormati masa tenggang dengan memasukkan `--no-should-respect-grace-period` opsi.

Gunakan [describe-auto-scaling-groups](#) perintah berikut untuk memverifikasi bahwa status kesehatan instans adalah `Unhealthy`.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names my-asg
```

Berikut ini adalah contoh tanggapan yang menunjukkan bahwa status kesehatan instance adalah `Unhealthy` dan bahwa instance tersebut akan berakhir.

```
{
  "AutoScalingGroups": [
    {
      ....
      "Instances": [
        {
          "ProtectedFromScaleIn": false,
          "AvailabilityZone": "us-west-2a",
          "LaunchTemplate": {
            "LaunchTemplateName": "my-launch-template",
            "Version": "1",
            "LaunchTemplateId": "lt-1234567890abcdef0"
          },
          "InstanceId": "i-1234567890abcdef0",
          "InstanceType": "t2.micro",
          "HealthStatus": "Unhealthy",
          "LifecycleState": "Terminating"
        },
        ...
      ]
    }
  ]
}
```

Sumber daya terkait

Untuk informasi tentang pemecahan masalah pemeriksaan kesehatan, lihat. [Memecahkan Masalah Auto Scaling Amazon EC2: Pemeriksaan Kesehatan](#) Jika pemeriksaan kesehatan Anda gagal, periksa topik ini untuk langkah-langkah pemecahan masalah. Topik ini akan membantu Anda

mengetahui apa yang salah dalam grup Auto Scaling Anda dan memberi Anda saran tentang cara memperbaikinya.

Auto Scaling Amazon EC2 juga memantau kesehatan instans yang Anda luncurkan ke kolam hangat menggunakan Amazon EC2, Amazon EBS, atau pemeriksaan kesehatan khusus. Untuk informasi selengkapnya, lihat [Lihat status pemeriksaan kesehatan dan alasan kegagalan pemeriksaan kesehatan](#).

Lihat alasan kegagalan pemeriksaan kesehatan

Dengan menggunakan prosedur berikut, Anda dapat melihat informasi tentang setiap contoh yang diganti karena pemeriksaan kesehatan.

Untuk melihat alasan kegagalan pemeriksaan kesehatan (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Pada tab Aktivitas, di Riwayat aktivitas, kolom Status menunjukkan apakah grup Auto Scaling Anda telah berhasil meluncurkan atau menghentikan instance.

Jika menghentikan kasus yang tidak sehat, kolom Penyebab menunjukkan tanggal dan waktu penghentian dan alasan kegagalan pemeriksaan kesehatan. Misalnya, At 2022-05-14T20:11:53Z an instance was taken out of service in response to an ELB system health check failure.

Untuk informasi tentang jenis kesalahan yang mungkin Anda temui dan cara menanganinya, lihat [Memecahkan Masalah Auto Scaling Amazon EC2: Pemeriksaan Kesehatan](#).

Note

Secara default, Auto Scaling Amazon EC2 membuat aktivitas penskalaan baru untuk menghentikan instance yang tidak sehat, lalu menghentikannya. Saat instance dihentikan, aktivitas penskalaan lain meluncurkan instance baru.

Anda dapat mengubah perilaku ini untuk meluncurkan instance baru terlebih dahulu dengan menggunakan kebijakan pemeliharaan instans. Dengan kebijakan pemeliharaan instans, Anda dapat menetapkan ambang batas untuk grup Auto Scaling ketika ada peristiwa yang

mengarah ke penggantian instance, dan grup Auto Scaling Anda hanya dapat mengganti instance dalam rentang ambang tersebut. Namun, karena Auto Scaling Amazon EC2 segera menghentikan instans yang tidak lagi berjalan, ambang batas bawah kebijakan pemeliharaan instans Anda dapat dilampaui jika instans berakhir secara tidak terduga atau Anda menghentikan atau mem-boot ulang instans secara manual. Untuk informasi selengkapnya, lihat [Kebijakan pemeliharaan instans](#).

Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling

Ketika pemeriksaan kesehatan Auto Scaling Amazon EC2 menentukan bahwa InService instans tidak sehat, instans menggantinya dengan instance baru. Masa tenggang pemeriksaan kesehatan menentukan jumlah waktu minimum (dalam hitungan detik) untuk menyimpan contoh baru dalam layanan sebelum menghentikannya jika ternyata tidak sehat.

Contoh kasus penggunaan mungkin merupakan persyaratan untuk Auto Scaling Amazon EC2 untuk menghindari tindakan jika pemeriksaan kesehatan Elastic Load Balancing gagal dan penyebabnya adalah instans masih diinisialisasi. Pemeriksaan kesehatan Elastic Load Balancing berjalan secara paralel, dimulai saat instance terdaftar dengan load balancer. Masa tenggang mencegah Amazon EC2 Auto Scaling menandai Unhealthy instans Anda yang baru diluncurkan dan menghentikannya secara tidak perlu jika mereka tidak segera lulus pemeriksaan kesehatan ini setelah memasuki negara bagian. InService

Di konsol, secara default, masa tenggang pemeriksaan kesehatan adalah 300 detik saat Anda membuat grup Auto Scaling. Nilai defaultnya adalah 0 detik saat Anda membuat grup Auto Scaling menggunakan AWS CLI atau SDK. Nilai 0 mematikan masa tenggang pemeriksaan kesehatan.

Menyetel nilai ini terlalu tinggi mengurangi efektivitas pemeriksaan kesehatan Auto Scaling Amazon EC2. Jika Anda menggunakan kait siklus hidup misalnya peluncuran, Anda dapat mengatur masa tenggang pemeriksaan kesehatan ke 0. Dengan kait siklus hidup, Auto Scaling Amazon EC2 menyediakan cara untuk memastikan bahwa instans selalu diinisialisasi sebelum memasuki status. InService Untuk informasi selengkapnya, lihat [Hook siklus aktif Amazon EC2 Auto Scaling](#).

Masa tenggang berlaku untuk contoh berikut:

- Instans yang baru diluncurkan
- Contoh yang dimasukkan kembali ke layanan setelah siaga
- Contoh yang Anda lampirkan secara manual ke grup

Important

Selama masa tenggang pemeriksaan kesehatan, jika Amazon EC2 Auto Scaling mendeteksi bahwa suatu instans tidak lagi dalam status Amazon EC2, instans segera menandai instans dan menggantikannya. `running Unhealthy` Misalnya, jika Anda menghentikan instance dalam grup Auto Scaling, itu ditandai `Unhealthy` dan diganti.

Tetapkan masa tenggang pemeriksaan kesehatan untuk grup

Anda dapat mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling baru dan yang sudah ada.

Console

Untuk mengubah masa tenggang pemeriksaan kesehatan untuk grup baru (konsol)

Saat Anda membuat grup Auto Scaling, pada halaman Konfigurasi opsi lanjutan, untuk pemeriksaan Kesehatan, Masa tenggang pemeriksaan Kesehatan, masukkan jumlah waktu, dalam hitungan detik. Ini adalah berapa lama Amazon EC2 Auto Scaling perlu menunggu sebelum memeriksa status kesehatan suatu instans setelah memasuki status `InService`

AWS CLI

Untuk memodifikasi masa tenggang pemeriksaan kesehatan untuk grup baru (AWS CLI)

Tambahkan `--health-check-grace-period` opsi ke [create-auto-scaling-group](#) perintah. Contoh berikut mengonfigurasi masa tenggang pemeriksaan kesehatan dengan nilai `60` detik untuk grup Auto Scaling baru bernama `my-asg`

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg --health-check-grace-period 60 ...
```

Console

Untuk mengubah masa tenggang pemeriksaan kesehatan untuk grup yang ada (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.

2. Pada bilah navigasi di bagian atas layar, pilih tempat Wilayah AWS Anda membuat grup Auto Scaling.
3. Pilih kotak centang di samping grup Auto Scaling.

Panel split terbuka di bagian bawah halaman.

4. Pada tab Detail, pilih Pemeriksaan kesehatan, Edit.
5. Di bawah Health check masa tenggang, masukkan jumlah waktu, dalam hitungan detik. Ini adalah berapa lama Amazon EC2 Auto Scaling perlu menunggu sebelum memeriksa status kesehatan suatu instans setelah memasuki status. InService
6. Pilih Perbarui.

AWS CLI

Untuk memodifikasi masa tenggang pemeriksaan kesehatan untuk grup yang ada (AWS CLI)

Tambahkan `--health-check-grace-period` opsi ke [update-auto-scaling-group](#) perintah. Contoh berikut mengonfigurasi masa tenggang pemeriksaan kesehatan dengan nilai `120` detik untuk grup Auto Scaling yang ada bernama `my-asg`

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg --health-check-grace-period 120
```

Note

Kami sangat menyarankan juga mengatur waktu pemanasan instans default untuk grup Auto Scaling Anda. Untuk informasi selengkapnya, lihat [Mengatur pemanasan instance default untuk grup Auto Scaling](#).

AWS Health Dashboard pemberitahuan untuk Auto Scaling Amazon EC2

Anda AWS Health Dashboard memberikan dukungan untuk notifikasi yang berasal dari Amazon EC2 Auto Scaling. Notifikasi ini memberikan panduan kesadaran dan remediasi untuk kinerja sumber daya atau masalah ketersediaan yang dapat memengaruhi aplikasi Anda. Hanya peristiwa khusus untuk grup keamanan yang hilang dan peluncuran templat yang saat ini tersedia.

AWS Health Dashboard Ini adalah bagian dari AWS Health layanan. Tidak perlu penyiapan dan dapat dilihat oleh pengguna yang diautentikasi di akun Anda. Untuk informasi selengkapnya, lihat [Memulai dengan AWS Health Dasbor Anda](#).

Jika Anda menerima pesan yang serupa dengan pesan berikut, pesan tersebut harus diperlakukan sebagai alarm untuk mengambil tindakan.

Contoh: Grup Auto Scaling tidak ditingkatkan karena grup keamanan yang hilang

Hello,

At 2020-01-11 04:00 UTC, we detected an issue with your Auto Scaling group [ARN] in Akun AWS 123456789012.

A security group associated with this Auto Scaling group cannot be found. Each time a scale out operation is performed, it will be prevented until you make a change that fixes the issue.

We recommend that you review and update your Auto Scaling group configuration to change the launch template or launch configuration that depends on the unavailable security group.

Sincerely,
Amazon Web Services

Contoh: Grup Auto Scaling tidak ditingkatkan karena template peluncuran yang hilang

Hello,

At 2021-05-11 04:00 UTC, we detected an issue with your Auto Scaling group [ARN] in Akun AWS 123456789012.

The launch template associated with this Auto Scaling group cannot be found. Each time a scale out operation is performed, it will be prevented until you make a change that fixes the issue.

We recommend that you review and update your Auto Scaling group configuration and specify an existing launch template to use.

Sincerely,
Amazon Web Services

Pantau CloudWatch metrik untuk grup dan instans Auto Scaling

Metrik adalah konsep dasar di Amazon CloudWatch. Metrik mewakili kumpulan titik data yang diurutkan waktu yang dipublikasikan ke CloudWatch. Pikirkan metrik sebagai variabel untuk memantau, dan titik data sebagai representasi nilai-nilai variabel tersebut dari waktu ke waktu. Anda dapat menggunakan metrik ini untuk memverifikasi bahwa sistem Anda bekerja sesuai harapan.

Metrik Auto Scaling Amazon EC2 yang mengumpulkan informasi tentang grup Auto Scaling ada di namespace. `AWS/AutoScaling` Metrik instans Amazon EC2 yang mengumpulkan CPU dan data penggunaan lainnya dari instans Auto Scaling ada di namespace. `AWS/EC2`

Konsol Auto Scaling Amazon EC2 menampilkan serangkaian grafik untuk metrik grup dan metrik instans gabungan untuk grup. Bergantung pada kebutuhan Anda, Anda mungkin lebih suka mengakses data untuk grup dan instans Auto Scaling dari Amazon, CloudWatch bukan konsol Auto Scaling Amazon EC2.

Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).

Daftar Isi

- [Lihat grafik pemantauan di konsol Auto Scaling Amazon EC2](#)
- [CloudWatch Metrik Amazon untuk Auto Scaling Amazon EC2](#)
- [Konfigurasi pemantauan untuk instans Auto Scaling](#)

Lihat grafik pemantauan di konsol Auto Scaling Amazon EC2

Di bagian Auto Scaling Amazon EC2 di konsol Amazon EC2, Anda dapat memantau kemajuan grup Auto Scaling individual minute-by-minute menggunakan metrik. CloudWatch

Anda dapat memantau jenis metrik berikut:

- Metrik Auto Scaling — Metrik Auto Scaling diaktifkan hanya jika Anda mengaktifkannya. Untuk informasi selengkapnya, lihat [Aktifkan metrik grup Auto Scaling \(konsol\)](#). Saat metrik Auto Scaling diaktifkan, grafik pemantauan menampilkan data yang dipublikasikan pada perincian satu menit untuk metrik Auto Scaling.

- Metrik EC2 — Metrik instans Amazon EC2 selalu diaktifkan. Ketika pemantauan terperinci diaktifkan, grafik pemantauan menunjukkan data yang dipublikasikan pada granularitas satu menit misalnya metrik. Untuk informasi selengkapnya, lihat [Konfigurasi pemantauan untuk instans Auto Scaling](#).

Untuk melihat grafik pemantauan menggunakan konsol Auto Scaling Amazon EC2

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling yang ingin Anda lihat metriknya.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Pilih tab Pemantauan.

Auto Scaling Amazon EC2 menampilkan grafik pemantauan untuk metrik Auto Scaling.

4. Untuk melihat grafik pemantauan metrik instans agregat untuk grup, pilih EC2.

Tindakan grafik

- Arahkan kursor ke titik data untuk melihat pop-up data untuk waktu tertentu di UTC.
- Untuk memperbesar grafik, pilih Perbesar dari alat menu (tiga titik vertikal) di kanan atas grafik. Atau, pilih ikon maksimalkan di bagian atas grafik.
- Sesuaikan periode waktu untuk data yang ditampilkan dalam grafik dengan memilih salah satu nilai periode waktu yang telah ditentukan. Jika grafik diperbesar, Anda dapat memilih Kustom untuk menentukan periode waktu Anda sendiri.
- Pilih Refresh dari alat menu untuk memperbarui data dalam grafik.
- Seret kursor Anda ke atas data grafik untuk memilih rentang tertentu. Anda kemudian dapat memilih Terapkan rentang waktu di alat menu.
- Pilih Lihat log dari alat menu untuk melihat aliran log terkait (jika ada) di CloudWatch konsol.
- Untuk melihat grafik CloudWatch, pilih Lihat dalam metrik dari alat menu. Ini membawa Anda ke CloudWatch halaman untuk grafik itu. Di sana, Anda dapat melihat lebih banyak informasi atau mengakses informasi historis untuk mendapatkan pemahaman yang lebih baik tentang bagaimana grup Auto Scaling Anda berubah selama periode yang diperpanjang.

Metrik grafik untuk grup Auto Scaling Anda

Setelah membuat grup Auto Scaling, Anda dapat membuka konsol Auto Scaling Amazon EC2 dan melihat grafik pemantauan untuk grup di tab Monitoring.

Di bagian Auto Scaling, metrik grafik menyertakan metrik berikut. Metrik ini memberikan pengukuran yang dapat menjadi indikator potensi masalah, seperti jumlah instance penghentian atau jumlah instance yang tertunda. Anda dapat menemukan definisi untuk metrik ini di [CloudWatch Metrik Amazon untuk Auto Scaling Amazon EC2](#).

Nama tampilan	CloudWatch nama metrik
Ukuran Grup Minimum	GroupMinSize
Ukuran Grup Maksimum	GroupMaxSize
Kapasitas yang Diinginkan	GroupDesiredCapacity
Dalam Instans Layanan	GroupInServiceInstances
Contoh Tertunda	GroupPendingInstances
Instans Siaga	GroupStandbyInstances
Mengakhiri Instans	GroupTerminatingInstances
Jumlah Contoh	GroupTotalInstances

Di bagian EC2, Anda dapat menemukan metrik grafik berikut berdasarkan metrik kinerja utama untuk instans Amazon EC2 Anda. Metrik EC2 ini adalah agregat metrik untuk semua instance dalam grup. Anda dapat menemukan definisi untuk metrik ini di [Daftar CloudWatch metrik yang tersedia untuk instans Anda](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Nama tampilan	CloudWatch nama metrik
Pemanfaatan CPU	CPUUtilization
Disk Membaca	DiskReadBytes

Nama tampilan	CloudWatch nama metrik
Operasi Baca Disk	DiskReadOps
Disk Menulis	DiskWriteBytes
Operasi Tulis Disk	DiskWriteOps
Jaringan Dalam	NetworkIn
Jaringan Keluar	NetworkOut
Pemeriksaan Status Gagal (Apa Saja)	StatusCheckFailed
Pemeriksaan Status Gagal (Instance)	StatusCheckFailed_Instance
Pemeriksaan Status Gagal (Sistem)	StatusCheckFailed_System

Selain itu, beberapa metrik tersedia untuk kasus penggunaan tertentu dalam metrik grafik Auto Scaling.

Metrik berikut berguna jika grup Anda menggunakan bobot yang menentukan berapa banyak unit yang disumbangkan setiap instance terhadap kapasitas grup yang diinginkan. Anda dapat menemukan definisi untuk metrik ini di [CloudWatch Metrik Amazon untuk Auto Scaling Amazon EC2](#).

Nama tampilan	CloudWatch nama metrik
Dalam Unit Kapasitas Layanan	GroupInServiceCapacity
Unit Kapasitas Tertunda	GroupPendingCapacity
Unit Kapasitas Siaga	GroupStandbyCapacity
Mengakhiri Unit Kapasitas	GroupTerminatingCapacity

Nama tampilan	CloudWatch nama metrik
Total Unit Kapasitas	GroupTotalCapacity

Metrik berikut berguna jika grup Anda menggunakan fitur [kolam hangat](#). Anda dapat menemukan definisi untuk metrik ini di [CloudWatch Metrik Amazon untuk Auto Scaling Amazon EC2](#).

Nama tampilan	CloudWatch nama metrik
Ukuran Minimum Kolam Hangat	WarmPoolMinSize
Kapasitas Kolam Hangat yang Diinginkan	WarmPoolDesiredCapacity
Unit Kapasitas Tertunda Kolam Hangat	WarmPoolPendingCapacity
Unit Kapasitas Terminasi Kolam Hangat	WarmPoolTerminatingCapacity
Unit Kapasitas Hangat Kolam Hangat	WarmPoolWarmedCapacity
Unit Kapasitas Total Kolam Hangat Diluncurkan	WarmPoolTotalCapacity
Kapasitas yang Diinginkan Grup dan Kolam Hangat	GroupAndWarmPoolDesiredCapacity
Unit Total Kapasitas Grup dan Kolam Hangat Diluncurkan	GroupAndWarmPoolTotalCapacity

Sumber daya terkait

- Untuk memantau metrik per instans, lihat Metrik [grafik untuk instans Anda di](#) Panduan Pengguna Amazon EC2 untuk Instans Linux.

- CloudWatch dasbor adalah halaman beranda yang dapat disesuaikan di konsol. CloudWatch Anda dapat menggunakan halaman ini untuk memantau sumber daya Anda dalam satu tampilan, bahkan termasuk sumber daya yang tersebar di berbagai Wilayah. Anda dapat menggunakan CloudWatch dasbor untuk membuat tampilan metrik dan alarm yang disesuaikan untuk sumber daya Anda. AWS Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#).

CloudWatch Metrik Amazon untuk Auto Scaling Amazon EC2

Auto Scaling Amazon EC2 menerbitkan metrik berikut di namespace. AWS/AutoScaling Metrik grup Auto Scaling aktual yang tersedia akan bergantung pada apakah metrik grup Anda diaktifkan, dan metrik grup mana yang Anda aktifkan. Metrik grup tersedia dengan perincian satu menit tanpa biaya tambahan, tetapi Anda harus mengaktifkannya.

Saat Anda mengaktifkan metrik grup Auto Scaling, Auto Scaling Amazon EC2 mengirimkan data CloudWatch sampel ke setiap menit dengan upaya terbaik. Dalam kasus yang jarang terjadi ketika CloudWatch mengalami gangguan layanan, data tidak diisi ulang untuk mengisi celah dalam riwayat metrik grup.

Daftar Isi

- [Metrik grup Auto Scaling](#)
- [Dimensi untuk metrik grup Auto Scaling](#)
- [Metrik dan dimensi penskalaan prediktif](#)
- [Aktifkan metrik grup Auto Scaling \(konsol\)](#)
- [Aktifkan metrik grup Auto Scaling \(\) AWS CLI](#)

Metrik grup Auto Scaling

Dengan metrik ini, Anda mendapatkan visibilitas yang hampir terus menerus ke dalam riwayat grup Auto Scaling Anda, seperti perubahan ukuran grup dari waktu ke waktu.

Metrik	Penjelasan
GroupMinSize	Ukuran minimum grup Auto Scaling. Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.

Metrik	Penjelasan
GroupMaxSize	<p>Ukuran maksimum grup Auto Scaling.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
GroupDesiredCapacity	<p>Jumlah instance yang diupayakan untuk dijaga oleh grup Auto Scaling.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
GroupInServiceInstances	<p>Jumlah instance yang berjalan sebagai bagian dari grup Auto Scaling. Metrik ini tidak mencakup instance yang tertunda atau berakhir.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
GroupPendingInstances	<p>Jumlah instance yang tertunda. Instance yang tertunda belum dalam layanan. Metrik ini tidak mencakup instance yang sedang dalam layanan atau berakhir.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
GroupStandbyInstances	<p>Jumlah instance yang berada dalam status Standby. Instance dalam status ini masih berjalan tetapi tidak aktif dalam layanan.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
GroupTerminatingInstances	<p>Jumlah instance yang sedang dalam proses penghentian. Metrik ini tidak mencakup instance yang sedang dalam layanan atau tertunda.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>

Metrik	Penjelasan
GroupTotalInstances	<p>Total jumlah instance dalam grup Auto Scaling. Metrik ini mengidentifikasi jumlah instance yang sedang dalam layanan, tertunda, dan diakhiri.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>

Saat Anda mengonfigurasi grup instans campuran untuk mengukur kapasitas yang diinginkan dalam unit yang berbeda, seperti dengan menetapkan bobot berdasarkan jumlah vCPU dari setiap jenis instans, metrik berikut menghitung jumlah unit yang digunakan oleh grup Auto Scaling Anda. Jika Anda tidak mengonfigurasi grup instance campuran untuk mengukur kapasitas yang diinginkan dalam unit yang berbeda, maka metrik berikut akan diisi, tetapi sama dengan metrik yang ditentukan dalam tabel sebelumnya. Untuk informasi lebih lanjut, lihat [Gambaran umum pengaturan](#).

Metrik	Penjelasan
GroupInServiceCapacity	<p>Jumlah unit kapasitas yang berjalan sebagai bagian dari grup Auto Scaling.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
GroupPendingCapacity	<p>Jumlah unit kapasitas yang tertunda.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
GroupStandbyCapacity	<p>Jumlah unit kapasitas dalam status Standby.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
GroupTerminatingCapacity	<p>Jumlah unit kapasitas yang sedang dalam proses penghentian.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>

Metrik	Penjelasan
GroupTotalCapacity	Total jumlah unit kapasitas dalam grup Auto Scaling. Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.

Auto Scaling Amazon EC2 juga melaporkan metrik berikut untuk grup Auto Scaling yang memiliki kolam hangat. Untuk informasi lebih lanjut, lihat [Kolam hangat untuk Auto Scaling Amazon EC2](#).

Metrik	Deskripsi
WarmPoolMinSize	Ukuran minimum kolam hangat. Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.
WarmPoolDesiredCapacity	Jumlah kapasitas yang coba dipertahankan oleh Amazon EC2 Auto Scaling di kolam hangat. Ini setara dengan ukuran maksimum grup Auto Scaling dikurangi kapasitas yang diinginkan, atau, jika diatur, sebagai kapasitas maksimum yang disiapkan dari grup Auto Scaling dikurangi kapasitas yang diinginkan. Namun, ketika ukuran minimum kolam hangat sama dengan atau lebih besar dari perbedaan antara ukuran maksimum (atau, jika diatur, kapasitas maksimum yang disiapkan) dan kapasitas yang diinginkan dari grup Auto Scaling, maka kapasitas kolam hangat yang diinginkan akan setara dengan <code>WarmPoolMinSize</code> Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.
WarmPoolPendingCapacity	Jumlah kapasitas di kolam hangat yang tertunda. Metrik ini tidak termasuk instance yang sedang berjalan, dihentikan, atau dihentikan.

Metrik	Deskripsi
	<p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
<p>WarmPoolTerminatingCapacity</p>	<p>Jumlah kapasitas di kolam hangat yang sedang dalam proses penghentian. Metrik ini tidak termasuk instance yang sedang berjalan, berhenti, atau tertunda.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
<p>WarmPoolWarmCapacity</p>	<p>Jumlah kapasitas yang tersedia untuk masuk ke grup Auto Scaling selama skala keluar. Metrik ini tidak mencakup instance yang tertunda atau berakhir.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
<p>WarmPoolTotalCapacity</p>	<p>Total kapasitas kolam hangat, termasuk contoh yang berjalan, berhenti, menunggu, atau berhenti.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
<p>GroupAndWarmPoolDesiredCapacity</p>	<p>Kapasitas yang diinginkan dari grup Auto Scaling dan kolam hangat digabungkan.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>
<p>GroupAndWarmPoolTotalCapacity</p>	<p>Total kapasitas grup Auto Scaling dan kolam hangat digabungkan. Ini termasuk instance yang berjalan, berhenti, tertunda, penghentian, atau dalam layanan.</p> <p>Kriteria pelaporan: Dilaporkan jika pengumpulan metrik diaktifkan.</p>

Dimensi untuk metrik grup Auto Scaling

Anda dapat menggunakan dimensi berikut untuk menyaring metrik yang tercantum di tabel sebelumnya.

Dimensi	Deskripsi
AutoScalingGroupName	Filter pada nama grup Auto Scaling.

Metrik dan dimensi penskalaan prediktif

AWS/AutoScalingNamespace menyertakan metrik berikut untuk penskalaan prediktif.

Metrik tersedia dengan resolusi satu jam.

Anda dapat mengevaluasi akurasi perkiraan dengan membandingkan nilai yang diperkirakan dengan nilai aktual. Untuk informasi selengkapnya tentang mengevaluasi akurasi perkiraan menggunakan metrik ini, lihat [Pantau metrik penskalaan prediktif dengan CloudWatch](#)

Metrik	Deskripsi	Dimensi
PredictiveScalingLoadForecast	<p>Jumlah beban yang diantisipasi akan dihasilkan oleh aplikasi Anda.</p> <p>Statistik AverageMinimum, dan Maximum statistik berguna, tetapi Sum statistiknya tidak.</p> <p>Kriteria pelaporan: Dilaporkan setelah perkiraan awal dibuat.</p>	AutoScalingGroupName , PolicyName , PairIndex
PredictiveScalingCapacityForecast	<p>Jumlah kapasitas yang diantisipasi yang dibutuhkan untuk memenuhi permintaan aplikasi. Ini didasarkan pada perkiraan beban dan tingkat pemanfaatan target di mana Anda ingin mempertahankan instance Auto Scaling Anda.</p> <p>Statistik AverageMinimum, dan Maximum statistik berguna, tetapi Sum statistiknya tidak.</p>	AutoScalingGroupName , PolicyName

Metrik	Deskripsi	Dimensi
	Kriteria pelaporan: Dilaporkan setelah perkiraan awal dibuat.	
PredictiveScalingMetricPairCorrelation	<p>Korelasi antara metrik penskalaan dan rata-rata per instance dari metrik beban. Penskalaan prediktif mengasumsikan korelasi tinggi. Oleh karena itu, jika Anda mengamati nilai rendah untuk metrik ini, lebih baik tidak menggunakan pasangan metrik.</p> <p>Statistik AverageMinimum, dan Maximum statistik berguna, tetapi Sum statistiknya tidak.</p> <p>Kriteria pelaporan: Dilaporkan setelah perkiraan awal dibuat.</p>	AutoScalingGroupName , PolicyName , PairIndex

Note

`PairIndex`Dimensi mengembalikan informasi yang terkait dengan indeks pasangan metrik penskalaan beban seperti yang ditetapkan oleh Auto Scaling Amazon EC2. Saat ini, satu-satunya nilai yang valid adalah `0`.

Aktifkan metrik grup Auto Scaling (konsol)

Untuk mengaktifkan metrik grup

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Monitoring, pilih Koleksi metrik grup Auto Scaling, Aktifkan kotak centang yang terletak di bagian atas halaman di bawah Auto Scaling.

Untuk menonaktifkan metrik grup

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih grup Auto Scaling Anda.
3. Pada tab Monitoring, hapus Koleksi metrik grup Auto Scaling, kotak centang Aktifkan.

Aktifkan metrik grup Auto Scaling () AWS CLI

Untuk mengaktifkan metrik grup Auto Scaling

Aktifkan satu atau beberapa metrik grup dengan menggunakan [enable-metrics-collection](#) perintah. Misalnya, perintah berikut memungkinkan metrik tunggal untuk grup Auto Scaling yang ditentukan.

```
aws autoscaling enable-metrics-collection --auto-scaling-group-name my-asg \  
--metrics GroupDesiredCapacity --granularity "1Minute"
```

Jika Anda menghilangkan opsi `--metrics`, semua metrik akan diaktifkan.

```
aws autoscaling enable-metrics-collection --auto-scaling-group-name my-asg \  
--granularity "1Minute"
```

Untuk menonaktifkan metrik grup Auto Scaling

Gunakan [disable-metrics-collection](#) perintah untuk menonaktifkan semua metrik grup.

```
aws autoscaling disable-metrics-collection --auto-scaling-group-name my-asg
```

Konfigurasi pemantauan untuk instans Auto Scaling

Amazon EC2 mengumpulkan dan memproses data mentah dari instans menjadi metrik hampir real-time yang dapat dibaca yang menjelaskan CPU dan data penggunaan lainnya untuk grup Auto Scaling Anda. Anda dapat mengonfigurasi interval untuk memantau metrik ini dengan memilih perincian satu menit atau lima menit.

Pemantauan instans diaktifkan setiap kali instance diluncurkan, menggunakan pemantauan dasar (granularitas lima menit) atau pemantauan terperinci (perincian satu menit). Berlaku biaya untuk monitoring terperinci. Untuk informasi selengkapnya, lihat [CloudWatch harga Amazon](#) dan

[Memantau instans yang Anda gunakan CloudWatch](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Sebelum membuat grup Auto Scaling, Anda harus membuat template peluncuran atau konfigurasi peluncuran yang memungkinkan jenis pemantauan yang sesuai dengan aplikasi Anda. Jika Anda menambahkan kebijakan penskalaan ke grup Anda, kami sangat menyarankan Anda menggunakan pemantauan terperinci untuk mendapatkan data metrik untuk instans EC2 dengan perincian satu menit, karena hal itu menghasilkan respons yang lebih cepat terhadap perubahan beban.

Daftar Isi

- [Aktifkan pemantauan terperinci \(konsol\)](#)
- [Aktifkan pemantauan terperinci \(AWS CLI\)](#)
- [Beralih antara pemantauan dasar dan terperinci](#)
- [Kumpulkan metrik tambahan menggunakan agen CloudWatch](#)

Aktifkan pemantauan terperinci (konsol)

Secara default, pemantauan dasar diaktifkan ketika Anda menggunakan AWS Management Console untuk membuat template peluncuran atau konfigurasi peluncuran.

Untuk mengaktifkan pemantauan terperinci dalam template peluncuran

Saat Anda membuat templat peluncuran menggunakan AWS Management Console, di bagian Detail lanjutan, untuk CloudWatch Pemantauan terperinci, pilih Aktifkan. Jika tidak, monitoring dasar akan diaktifkan. Untuk informasi selengkapnya, lihat [Buat template peluncuran menggunakan pengaturan lanjutan](#).

Untuk mengaktifkan pemantauan terperinci dalam konfigurasi peluncuran

Saat Anda membuat konfigurasi peluncuran menggunakan AWS Management Console, di bagian Konfigurasi tambahan, pilih Aktifkan pemantauan terperinci instans EC2 di dalamnya CloudWatch. Jika tidak, monitoring dasar akan diaktifkan. Untuk informasi selengkapnya, lihat [Buat konfigurasi peluncuran](#).

Aktifkan pemantauan terperinci (AWS CLI)

Secara default, pemantauan dasar diaktifkan saat Anda membuat template peluncuran menggunakan AWS CLI. Pemantauan terperinci diaktifkan secara default saat Anda membuat konfigurasi peluncuran menggunakan AWS CLI.

Untuk mengaktifkan pemantauan terperinci dalam template peluncuran

Untuk templat peluncuran, gunakan perintah [create-launch-template](#) dan lewati file JSON yang berisi informasi untuk membuat templat peluncuran. Atur atribut monitoring ke "Monitoring": {"Enabled":true} untuk memungkinkan monitoring terperinci atau "Monitoring": {"Enabled":false} untuk memungkinkan monitoring dasar.

Untuk mengaktifkan pemantauan terperinci dalam konfigurasi peluncuran

Untuk konfigurasi peluncuran, gunakan perintah [create-launch-configuration](#) dengan opsi `--instance-monitoring`. Atur opsi ini ke `true` untuk memungkinkan monitoring terperinci atau `false` untuk memungkinkan monitoring dasar.

```
--instance-monitoring Enabled=true
```

Beralih antara pemantauan dasar dan terperinci

Untuk mengubah jenis pemantauan yang diaktifkan pada instans EC2 baru, perbarui template peluncuran atau perbarui grup Auto Scaling untuk menggunakan templat peluncuran baru atau konfigurasi peluncuran. Proses yang ada lanjut menggunakan jenis monitoring yang sebelumnya diaktifkan. [Untuk memperbarui semua instance, hentikan sehingga diganti oleh grup Auto Scaling Anda atau perbarui instans satu per satu menggunakan instance monitor dan instans unmonitor-](#).

Note

Dengan fitur penyegaran instans dan masa pakai instans maksimum, Anda juga dapat mengganti semua instance di grup Auto Scaling untuk meluncurkan instance baru yang menggunakan pengaturan baru. Untuk informasi selengkapnya, lihat [Daur ulang instans di grup Auto Scaling Anda](#).

Saat Anda beralih antara pemantauan dasar dan terperinci:

Jika Anda memiliki CloudWatch alarm yang terkait dengan kebijakan penskalaan langkah atau kebijakan penskalaan sederhana untuk grup Auto Scaling, gunakan [put-metric-alarm](#) perintah untuk memperbarui setiap alarm. Jadikan setiap periode sesuai dengan jenis monitoring (300 detik untuk monitoring dasar dan 60 detik untuk monitoring terperinci). Jika Anda berubah dari monitoring terperinci menjadi monitoring dasar tetapi tidak memperbarui alarm Anda untuk menyesuaikan

periode lima menit, maka akan terus memeriksa statistik setiap menit. Mereka mungkin tidak menemukan data tersedia sebanyak empat dari setiap lima periode.

Kumpulkan metrik tambahan menggunakan agen CloudWatch

Untuk mengumpulkan metrik tingkat sistem operasi seperti memori yang tersedia dan yang digunakan, Anda harus menginstal agen. CloudWatch Biaya tambahan mungkin berlaku. Anda dapat menggunakan CloudWatch agen untuk mengumpulkan metrik sistem dan file log dari instans Amazon EC2. Untuk informasi selengkapnya, lihat [Metrik yang dikumpulkan oleh CloudWatch agen](#) di Panduan CloudWatch Pengguna Amazon.

Log panggilan API Auto Scaling Amazon EC2 dengan AWS CloudTrail

Auto Scaling Amazon EC2 terintegrasi AWS CloudTrail dengan, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau layanan menggunakan Amazon EC2 Auto Scaling. CloudTrail menangkap semua panggilan API untuk Amazon EC2 Auto Scaling sebagai peristiwa. Panggilan yang direkam mencakup panggilan dari konsol Amazon EC2 Auto Scaling dan panggilan kode ke Amazon EC2 Auto Scaling API.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk acara untuk Auto Scaling Amazon EC2. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk Amazon EC2 Auto Scaling, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

Informasi Auto Scaling Amazon EC2 di CloudTrail

CloudTrail diaktifkan di akun Amazon Web Services Anda saat Anda membuat akun. Saat aktivitas terjadi di Auto Scaling Amazon EC2, aktivitas tersebut direkam dalam CloudTrail suatu peristiwa bersama dengan peristiwa Amazon Web Services lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh kejadian terbaru di akun Amazon Web Services Anda. Untuk informasi selengkapnya, lihat [Melihat peristiwa dengan riwayat CloudTrail acara](#).

Untuk catatan peristiwa yang sedang berlangsung di akun Amazon Web Services Anda, termasuk peristiwa untuk Auto Scaling Amazon EC2, buat jejak. Jejak memungkinkan CloudTrail untuk

mengirimkan file log ke bucket Amazon S3. Secara default, ketika Anda membuat jejak di konsol tersebut, jejak diterapkan ke semua Wilayah. Trail mencatat peristiwa dari semua Wilayah di partisi Amazon Web Services dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi Amazon Web Services lainnya untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

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

Semua tindakan Auto Scaling Amazon EC2 dicatat CloudTrail oleh dan didokumentasikan dalam Referensi API [Auto Scaling Amazon EC2](#). Misalnya, panggilan ke `CreateLaunchConfiguration`, `DescribeAutoScalingGroup`, dan `UpdateAutoScalingGroup` tindakan menghasilkan entri dalam file CloudTrail log.

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

- Apakah permintaan itu dibuat dengan kredensial pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan tersebut dibuat oleh layanan lain.

Untuk informasi lebih lanjut, lihat [CloudTrailuserIdentityelemen](#).

Memahami entri file log Amazon EC2 Auto Scaling

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

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan CreateLaunchConfiguration tindakan.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "Root",
    "principalId": "123456789012",
    "arn": "arn:aws:iam::123456789012:root",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-08-21T17:05:42Z"
      }
    }
  },
  "eventTime": "2018-08-21T17:07:49Z",
  "eventSource": "autoscaling.amazonaws.com",
  "eventName": "CreateLaunchConfiguration",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "Coral/Jakarta",
  "requestParameters": {
    "ebsOptimized": false,
    "instanceMonitoring": {
      "enabled": false
    }
  },
  "instanceType": "t2.micro",
  "keyName": "EC2-key-pair-oregon",
  "blockDeviceMappings": [
    {
      "deviceName": "/dev/xvda",
      "ebs": {
        "deleteOnTermination": true,
        "volumeSize": 8,
        "snapshotId": "snap-01676e0a2c3c7de9e",
        "volumeType": "gp2"
      }
    }
  ],
  "launchConfigurationName": "launch_configuration_1",
```

```
    "imageId": "ami-6cd6f714d79675a5",
    "securityGroups": [
      "sg-00c429965fd921483"
    ],
    "responseElements": null,
    "requestID": "0737e2ea-fb2d-11e3-bfd8-99133058e7bb",
    "eventID": "3fcfb182-98f8-4744-bd45-b38835ab61cb",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
  }
```

Sumber daya terkait

Dengan CloudWatch Log, Anda dapat memantau dan menerima peringatan untuk peristiwa tertentu yang ditangkap oleh CloudTrail. Peristiwa yang dikirim ke CloudWatch Log adalah peristiwa yang dikonfigurasi untuk dicatat oleh jejak Anda, jadi pastikan Anda telah mengonfigurasi jejak atau jejak Anda untuk mencatat jenis acara yang ingin Anda pantau. CloudWatch Log dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang batas tertentu terpenuhi. Anda juga dapat mengarsipkan data log dalam penyimpanan yang sangat durabel. Untuk informasi selengkapnya, lihat [Panduan Pengguna Amazon CloudWatch Logs](#) dan [Memantau file CloudTrail log dengan topik Amazon CloudWatch Logs](#) di Panduan AWS CloudTrail Pengguna.

Opsi pemberitahuan Amazon SNS untuk Auto Scaling Amazon EC2

Anda dapat mengonfigurasi grup Auto Scaling untuk memberi tahu Anda tentang peristiwa penting yang memengaruhi aplikasi Anda. Dengan notifikasi, Anda juga dapat menghilangkan polling, dan Anda tidak akan menemukan RequestLimitExceeded kesalahan yang terkadang dihasilkan dari polling.

Ada dua cara untuk menerima pemberitahuan tentang Auto Scaling Amazon EC2:

- Layanan Pemberitahuan Sederhana Amazon — Amazon SNS dapat memberi tahu Anda saat grup Auto Scaling meluncurkan atau menghentikan instans. Anda hanya dapat mengaktifkan atau menonaktifkan pemberitahuan Amazon SNS. Untuk informasi selengkapnya, lihat [Amazon SNS dan Amazon EC2 Auto Scaling](#).
- Amazon EventBridge - EventBridge menyediakan notifikasi yang lebih canggih dan didorong oleh peristiwa yang disesuaikan dengan kriteria yang ditentukan dan dikirim ke berbagai target,

termasuk Amazon SNS. EventBridge juga dapat memantau berbagai peristiwa Auto Scaling yang lebih luas untuk pemantauan yang lebih tepat. Untuk informasi selengkapnya, lihat [Gunakan EventBridge untuk menangani peristiwa Auto Scaling](#).

Anda juga dapat melakukan tindakan kustom saat instance memasuki status tertunda selama peluncuran atau penghentian dengan menggunakan kait siklus hidup dan layanan seperti, Amazon EventBridge SNS, dan Amazon SQS. Pengait siklus hidup juga dapat memberikan waktu tambahan bagi instans baru untuk menyelesaikan skrip yang ditentukan dalam data pengguna sebelum Auto Scaling Amazon EC2 menambahkan instance ke grup. Untuk informasi selengkapnya, lihat [Hook siklus aktif Amazon EC2 Auto Scaling](#).

Amazon SNS dan Amazon EC2 Auto Scaling

Bagian ini menunjukkan cara menggunakan Amazon SNS untuk memantau saat grup Auto Scaling Anda meluncurkan atau menghentikan instans.

Misalnya, jika Anda mengonfigurasi grup Auto Scaling untuk menggunakan jenis notifikasi `autoscaling: EC2_INSTANCE_TERMINATE`, dan grup Auto Scaling Anda akan menghentikan suatu instance, notifikasi email akan dikirimkan. Email ini berisi detail instance yang dihentikan, seperti ID instance dan penyebab instance tersebut dihentikan.

Perhatikan bahwa saat Auto Scaling Amazon EC2 menambahkan atau menghapus instans dari grup, pemberitahuan tentang perubahan ini akan dikirimkan kepada Anda, dengan satu pemberitahuan dikirim per instans. Namun, pengiriman notifikasi ini dilakukan dengan upaya terbaik, dan instans Anda masih bisa gagal setelah pemberitahuan awal, misalnya, jika pemeriksaan kesehatan nanti gagal. Jadi, meskipun Auto Scaling Amazon EC2 memberi tahu Anda pada awalnya, sebuah instance masih bisa gagal di kemudian hari. Perhatikan bahwa Anda dapat mengonfigurasi berapa lama setelah meluncurkan instans Auto Scaling Amazon EC2 menunggu sebelum melakukan pemeriksaan kesehatan pertama. Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).

Untuk informasi selengkapnya tentang Amazon SNS secara umum, lihat Panduan [Pengembang Layanan Pemberitahuan Sederhana Amazon](#).

Daftar Isi

- [Notifikasi SNS](#)
- [Konfigurasi notifikasi Amazon SNS untuk Auto Scaling Amazon EC2](#)
 - [Membuat topik Amazon SNS](#)

- [Berlangganan topik Amazon SNS](#)
- [Mengonfirmasi langganan Amazon SNS Anda](#)
- [Mengonfigurasi grup Auto Scaling Anda untuk mengirim notifikasi](#)
- [Menguji notifikasi](#)
- [Menghapus konfigurasi notifikasi](#)
- [Kebijakan utama untuk topik Amazon SNS terenkripsi](#)

Notifikasi SNS

Amazon EC2 Auto Scaling mendukung pengiriman notifikasi Amazon SNS saat peristiwa berikut terjadi.

Peristiwa	Penjelasan
<code>autoscaling:EC2_INSTANCE_LAUNCH</code>	Peluncuran instance berhasil
<code>autoscaling:EC2_INSTANCE_LAUNCH_ERROR</code>	Peluncuran instance gagal
<code>autoscaling:EC2_INSTANCE_TERMINATE</code>	Penghentian instance berhasil
<code>autoscaling:EC2_INSTANCE_TERMINATE_ERROR</code>	Penghentian instance gagal

Pesan ini mencakup informasi berikut:

- `Event` — Peristiwa.
- `AccountId`— ID akun Amazon Web Services.
- `AutoScalingGroupName` — Nama grup Auto Scaling.
- `AutoScalingGroupARN` — ARN grup Auto Scaling.
- `EC2InstanceId` — ID instance EC2.

Sebagai contoh:

```
Service: AWS Auto Scaling
Time: 2016-09-30T19:00:36.414Z
RequestId: 4e6156f4-a9e2-4bda-a7fd-33f2ae528958
Event: autoscaling:EC2_INSTANCE_LAUNCH
AccountId: 123456789012
AutoScalingGroupName: my-asg
AutoScalingGroupARN: arn:aws:autoscaling:region:123456789012:autoScalingGroup...
ActivityId: 4e6156f4-a9e2-4bda-a7fd-33f2ae528958
Description: Launching a new EC2 instance: i-0598c7d356eba48d7
Cause: At 2016-09-30T18:59:38Z a user request update of AutoScalingGroup constraints
to ...
StartTime: 2016-09-30T19:00:04.445Z
EndTime: 2016-09-30T19:00:36.414Z
StatusCode: InProgress
StatusMessage:
Progress: 50
EC2InstanceId: i-0598c7d356eba48d7
Details: {"Subnet ID":"subnet-id","Availability Zone":"zone"}
Origin: AutoScalingGroup
Destination: EC2
```

Konfigurasi notifikasi Amazon SNS untuk Auto Scaling Amazon EC2

Untuk menggunakan Amazon SNS guna mengirim notifikasi email, Anda harus membuat topik, lalu berlangganan alamat email Anda sesuai topik.

Membuat topik Amazon SNS

Topik SNS adalah titik akses logis, saluran komunikasi yang digunakan grup Auto Scaling Anda untuk mengirim notifikasi. Anda membuat topik dengan menentukan nama untuk topik Anda.

Saat Anda membuat nama topik, nama harus memenuhi persyaratan berikut:

- Antara 1 hingga 256 karakter
- Memuat huruf besar dan huruf kecil ASCII, angka, garis bawah, atau tanda hubung

Untuk informasi lebih lanjut, lihat [Membuat topik Amazon SNS](#) dalam Panduan Developer Amazon Simple Notification Service.

Berlangganan topik Amazon SNS

Untuk menerima notifikasi yang dikirimkan grup Auto Scaling Anda ke topik, Anda harus melangganan endpoint ke topik tersebut. Dalam prosedur ini, untuk Endpoint, tentukan alamat email untuk menerima notifikasi dari Amazon EC2 Auto Scaling.

Untuk informasi selengkapnya, lihat [Berlangganan topik Amazon SNS](#) di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

Mengonfirmasi langganan Amazon SNS Anda

Amazon SNS mengirimkan email konfirmasi ke alamat email yang Anda tentukan pada langkah sebelumnya.

Pastikan Anda membuka email dari AWS Notifikasi dan pilih tautan untuk mengonfirmasi langganan sebelum melanjutkan langkah berikutnya.

Anda akan menerima pesan pengakuan dari AWS Amazon SNS kini dikonfigurasi untuk menerima notifikasi dan mengirimkan notifikasi sebagai email ke alamat email yang Anda tentukan.

Mengonfigurasi grup Auto Scaling Anda untuk mengirim notifikasi

Anda dapat mengonfigurasi grup Auto Scaling Anda untuk mengirim notifikasi ke Amazon SNS ketika peristiwa penskalaan, seperti terjadinya peluncuran instance atau penghentian instance. Amazon SNS mengirimkan notifikasi dengan informasi tentang instance ke alamat email yang telah Anda tentukan.

Guna mengonfigurasi notifikasi Amazon SNS untuk grup Auto Scaling (konsol) Anda

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel terpisah terbuka di bagian bawah halaman, menunjukkan informasi tentang grup yang dipilih.

3. Pada tab Aktivitas, pilih Notifikasi aktivitas, Buat notifikasi.
4. Pada panel Buat notifikasi, lakukan hal berikut:
 - a. Untuk Topik SNS, pilih topik SNS Anda.
 - b. Untuk Jenis peristiwa, pilih peristiwa untuk dikirim notifikasi.

c. Pilih Buat.

Untuk mengonfigurasi notifikasi Amazon SNS untuk grup Auto Scaling (AWS CLI)

Gunakan perintah [put-notification-configuration](#) berikut ini.

```
aws autoscaling put-notification-configuration --auto-scaling-group-name my-  
asg --topic-arn arn --notification-types "autoscaling:EC2_INSTANCE_LAUNCH"  
"autoscaling:EC2_INSTANCE_TERMINATE"
```

Menguji notifikasi

Untuk membuat notifikasi tentang peristiwa peluncuran, perbarui grup Auto Scaling dengan meningkatkan kapasitas yang diinginkan dari grup Auto Scaling sebesar 1. Anda menerima notifikasi dalam beberapa menit setelah peluncuran instance.

Untuk mengubah kapasitas yang diinginkan (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel terpisah akan terbuka di bagian bawah halaman Grup Auto Scaling, yang menunjukkan informasi tentang grup yang dipilih.

3. Pada tab Detail, pilih Detail grup, Edit.
4. Untuk Kapasitas yang diinginkan, naikkan nilai saat ini sebesar 1. Jika nilai ini melebihi Kapasitas maksimum, Anda juga harus meningkatkan nilai Kapasitas maksimum sebesar 1.
5. Pilih Perbarui.
6. Setelah beberapa menit, Anda akan menerima notifikasi untuk peristiwa tersebut. Jika Anda tidak memerlukan instance tambahan yang Anda luncurkan untuk uji ini, Anda dapat mengurangi Kapasitas yang diinginkan sebesar 1. Setelah beberapa menit, Anda akan menerima notifikasi untuk peristiwa tersebut.

Menghapus konfigurasi notifikasi

Anda dapat menghapus konfigurasi notifikasi Auto Scaling Amazon EC2 Anda jika sudah tidak digunakan lagi.

Untuk menghapus konfigurasi notifikasi Auto Scaling Amazon EC2 (konsol)

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih grup Auto Scaling Anda.
3. Pada tab Aktivitas, pilih kotak centang di samping notifikasi yang ingin Anda hapus, lalu pilih Tindakan, Hapus.

Untuk menghapus konfigurasi notifikasi Auto Scaling Amazon EC2 (AWS CLI)

Gunakan perintah berikut delete-notification-configuration.

```
aws autoscaling delete-notification-configuration --auto-scaling-group-name my-asg --  
topic-arn arn
```

Untuk informasi tentang menghapus topik Amazon SNS dan semua langganan yang terkait dengan grup Auto Scaling, lihat Menghapus [langganan dan topik Amazon SNS](#) di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

Kebijakan utama untuk topik Amazon SNS terenkripsi

Topik Amazon SNS yang Anda tentukan mungkin dienkripsi dengan kunci terkelola pelanggan yang dibuat dengan AWS Key Management Service Untuk memberikan izin Auto Scaling Amazon EC2 untuk mempublikasikan ke topik terenkripsi, Anda harus terlebih dahulu membuat kunci KMS Anda dan kemudian menambahkan pernyataan berikut ke kebijakan kunci KMS. Ganti contoh ARN dengan ARN dari peran terkait layanan yang sesuai yang diizinkan akses ke kunci. Untuk informasi selengkapnya, lihat [Mengonfigurasi AWS KMS izin di Panduan](#) Pengembang Layanan Pemberitahuan Sederhana Amazon.

Dalam contoh ini, pernyataan kebijakan memberikan AWSServiceRoleForAutoScalingizin bernama peran terkait layanan untuk menggunakan kunci terkelola pelanggan. Untuk mempelajari selengkapnya tentang peran terkait layanan Auto Scaling Amazon EC2, lihat. [Peran terkait layanan untuk Amazon EC2 Auto Scaling](#)

```
{  
  "Sid": "Allow service-linked role use of the customer managed key",  
  "Effect": "Allow",  
  "Principal": {
```



```
"AWS": "arn:aws:iam::123456789012:role/aws-service-role/autoscaling.amazonaws.com/  
AWSServiceRoleForAutoScaling"  
},  
"Action": [  
  "kms:GenerateDataKey*",  
  "kms:Decrypt"  
],  
"Resource": "*" ]
```

Kunci `aws:SourceAccount` kondisi `aws:SourceArn` dan tidak didukung dalam kebijakan utama yang memungkinkan Auto Scaling Amazon EC2 dipublikasikan ke topik terenkripsi.

AWS layanan terintegrasi dengan Amazon EC2 Auto Scaling

Auto Scaling Amazon EC2 dapat diintegrasikan dengan layanan lain. AWS Tinjau opsi integrasi berikut untuk mempelajari lebih lanjut tentang cara kerja setiap layanan dengan Auto Scaling Amazon EC2.

Topik

- [Gunakan Rebalancing Kapasitas untuk menangani interupsi Spot Amazon EC2](#)
- [Gunakan Reservasi Kapasitas Sesuai Permintaan untuk memesan kapasitas di Availability Zone tertentu](#)
- [Buat grup Auto Scaling dari baris perintah menggunakan AWS CloudShell](#)
- [Buat grup Auto Scaling dengan AWS CloudFormation](#)
- [Gunakan AWS Compute Optimizer untuk mendapatkan rekomendasi untuk jenis instans untuk grup Auto Scaling](#)
- [Gunakan Elastic Load Balancing untuk mendistribusikan lalu lintas di seluruh instans dalam grup Auto Scaling](#)
- [Rutekan lalu lintas ke grup Auto Scaling Anda dengan grup target VPC Lattice](#)
- [Gunakan EventBridge untuk menangani peristiwa Auto Scaling](#)
- [Menyediakan konektivitas jaringan untuk instans Auto Scaling menggunakan Amazon VPC](#)

Gunakan Rebalancing Kapasitas untuk menangani interupsi Spot Amazon EC2

Anda dapat mengonfigurasi Amazon EC2 Auto Scaling untuk memonitor dan secara otomatis merespons perubahan yang memengaruhi ketersediaan Instance Spot Anda. Penyeimbangan Ulang Kapasitas membantu Anda mempertahankan ketersediaan beban kerja dengan secara proaktif menambah armada Anda dengan Instans Spot baru sebelum instans yang berjalan diinterupsi oleh Amazon EC2.

Tujuan dari Capacity Rebalancing adalah untuk terus memproses beban kerja Anda tanpa gangguan. Jika Instans Spot berisiko tinggi mengalami gangguan, layanan Amazon EC2 Spot memberi tahu Auto Scaling Amazon EC2 dengan rekomendasi penyeimbangan ulang instans EC2.

Saat Anda mengaktifkan Penyeimbangan Kembali Kapasitas untuk grup Auto Scaling, Auto Scaling Amazon EC2 mencoba mengganti Instans Spot secara proaktif di grup Anda yang telah menerima rekomendasi penyeimbangan ulang. Ini memberikan kesempatan untuk menyeimbangkan kembali beban kerja Anda ke Instans Spot baru yang tidak berisiko tinggi mengalami gangguan. Beban kerja Anda dapat terus memproses pekerjaan sementara Amazon EC2 Auto Scaling meluncurkan Instans Spot baru sebelum instans yang ada terputus.

Jika Anda tidak menggunakan Penyeimbangan Kembali Kapasitas, Auto Scaling Amazon EC2 tidak menggantikan Instans Spot hingga setelah layanan Amazon EC2 Spot mengganggu instans dan pemeriksaan kesehatannya gagal. Sebelum menyela instans, Amazon EC2 selalu memberikan rekomendasi penyeimbangan ulang instans EC2 dan pemberitahuan interupsi instans dua menit Spot.

Daftar Isi

- [Gambaran Umum](#)
- [Perilaku penyeimbangan kembali kapasitas](#)
- [Pertimbangan](#)
- [Aktifkan Rebalancing Kapasitas \(konsol\)](#)
- [Aktifkan Rebalancing Kapasitas \(AWS CLI\)](#)
- [Sumber daya terkait](#)
- [Batasan](#)

Gambaran Umum

Untuk menggunakan Capacity Rebalancing dengan grup Auto Scaling Anda, langkah-langkah dasarnya adalah:

1. Konfigurasi grup Auto Scaling Anda untuk menggunakan beberapa jenis instans dan Availability Zone. Dengan cara ini, Auto Scaling Amazon EC2 dapat melihat kapasitas yang tersedia untuk Instans Spot di setiap Availability Zone. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).
2. Tambahkan kait siklus hidup sesuai kebutuhan untuk melakukan shutdown aplikasi Anda yang anggun di dalam instance yang menerima notifikasi penyeimbangan ulang. Untuk informasi selengkapnya, lihat [Hook siklus aktif Amazon EC2 Auto Scaling](#).

Berikut ini adalah beberapa alasan mengapa Anda mungkin menggunakan pengait siklus hidup:

- Untuk pemastian worker Amazon SQS secara baik
 - Untuk menyelesaikan deregistrasi dari Domain Name System (DNS)
 - Untuk menarik log sistem atau aplikasi dan mengunggahnya ke Amazon Simple Storage Service (Amazon S3)
3. Kembangkan tindakan kustom untuk kait siklus hidup. Untuk menjalankan tindakan kustom Anda sesegera mungkin, Anda perlu tahu kapan instance siap untuk dihentikan. Cari tahu ini dengan mendeteksi status siklus hidup instance.
- Untuk menjalankan tindakan di luar instance, tulis EventBridge aturan dan otomatisasi tindakan apa yang harus diambil ketika pola peristiwa cocok dengan aturan.
 - Untuk menjalankan tindakan di dalam instance, konfigurasi instance untuk menjalankan skrip shutdown dan mengambil status siklus hidup melalui metadata instance.

Sangat penting untuk merancang tindakan khusus untuk diselesaikan dalam waktu kurang dari dua menit. Ini memastikan ada cukup waktu untuk menyelesaikan tugas sebelum penghentian instance.

Setelah Anda menyelesaikan langkah-langkah ini, Anda dapat mulai menggunakan Capacity Rebalancing.

Perilaku penyeimbangan kembali kapasitas

Dengan Penyeimbangan Kembali Kapasitas, Auto Scaling Amazon EC2 berperilaku seperti berikut saat instans menerima rekomendasi penyeimbangan ulang:

- Saat Instans Spot baru diluncurkan, Auto Scaling Amazon EC2 menunggu hingga instans baru melewati pemeriksaan kesehatannya sebelum menghentikan instance sebelumnya. Saat mengganti lebih dari satu instance, penghentian setiap instance sebelumnya dimulai setelah instans baru diluncurkan dan lulus pemeriksaan kesehatannya.
- Karena Auto Scaling Amazon EC2 mencoba meluncurkan instans baru sebelum menghentikan instans sebelumnya, berada pada atau mendekati kapasitas maksimum yang ditentukan dapat menghambat atau sepenuhnya menghentikan aktivitas penyeimbangan kembali. Untuk menghindari masalah ini, Amazon EC2 Auto Scaling untuk sementara dapat melebihi ukuran maksimum grup hingga 10 persen dari kapasitas yang diinginkan.
- Jika Anda tidak menambahkan kait siklus hidup ke grup Auto Scaling, Auto Scaling Amazon EC2 mulai menghentikan instans sebelumnya segera setelah instans baru lulus pemeriksaan kesehatannya.

- Jika Anda menambahkan kait siklus hidup, ini memperpanjang jumlah waktu yang diperlukan sebelum kami mulai menghentikan instance sebelumnya dengan nilai batas waktu yang Anda tentukan untuk hook siklus hidup.
- Jika Anda menggunakan kebijakan penskalaan atau penskalaan terjadwal, aktivitas penskalaan berjalan secara paralel. Jika aktivitas penskalaan sedang berlangsung dan grup Auto Scaling Anda berada di bawah kapasitas baru yang diinginkan, Auto Scaling Amazon EC2 akan keluar terlebih dahulu sebelum menghentikan instans sebelumnya.

Jika tidak ada kapasitas untuk jenis instans Anda di satu Availability Zone, Auto Scaling Amazon EC2 terus mencoba meluncurkan Instans Spot di Availability Zone lain yang diaktifkan hingga berhasil.

Dalam skenario terburuk, jika instance baru gagal diluncurkan atau pemeriksaan kesehatannya gagal, Auto Scaling Amazon EC2 terus mencoba meluncurkannya kembali. Saat mencoba meluncurkan instance baru, yang sebelumnya pada akhirnya akan terganggu dan dihentikan secara paksa dengan pemberitahuan interupsi dua menit.

Pertimbangan

Pertimbangkan hal berikut saat menggunakan Capacity Rebalancing:

Rancang aplikasi Anda agar toleran terhadap interupsi Spot

Aplikasi Anda harus dapat menangani perubahan dinamis dalam jumlah instance dan kemungkinan Instance Spot terputus lebih awal. Misalnya, jika grup Auto Scaling Anda berada di belakang penyeimbang beban Elastic Load Balancing, Auto Scaling Amazon EC2 akan menunggu instans membatalkan pendaftaran dari penyeimbang beban sebelum memanggil hook siklus hidup Anda. Jika waktu untuk membatalkan pendaftaran instans dan menyelesaikan tindakan siklus hidup terlalu lama, instans mungkin terputus saat Auto Scaling Amazon EC2 menunggu tindakan siklus hidup Anda selesai sebelum menghentikan instance.

Amazon EC2 tidak selalu memungkinkan untuk mengirim sinyal rekomendasi penyeimbangan ulang sebelum pemberitahuan interupsi Instans Spot dua menit. Terkadang, sinyal rekomendasi penyeimbangan kembali tiba bersamaan dengan pemberitahuan interupsi dua menit. Ketika ini terjadi, Amazon EC2 Auto Scaling memanggil hook siklus hidup dan mencoba meluncurkan Instans Spot baru dengan segera.

Hindari peningkatan risiko gangguan penggantian Instans Spot

Instans Spot pengganti Anda mungkin berisiko tinggi mengalami gangguan jika Anda menggunakan strategi `lowest-price` alokasi. Ini karena kami meluncurkan instans di

kolam dengan harga terendah yang memiliki kapasitas yang tersedia pada saat itu, meskipun Instans Spot pengganti Anda kemungkinan akan terganggu segera setelah diluncurkan. Untuk menghindari peningkatan risiko gangguan, kami sangat menyarankan agar Anda tidak menggunakan strategi `lowest-price` alokasi. Sebagai gantinya, kami merekomendasikan strategi `price-capacity-optimized` alokasi. Strategi ini meluncurkan Instans Spot pengganti di kumpulan Spot yang paling tidak mungkin terganggu dan memiliki harga serendah mungkin. Oleh karena itu, mereka cenderung tidak terganggu dalam waktu dekat.

Auto Scaling Amazon EC2 hanya akan meluncurkan instance baru jika ketersediaannya sama atau lebih baik

Salah satu tujuan dari Penyeimbangan Ulang kapasitas adalah untuk meningkatkan ketersediaan Instans Spot. Jika Instans Spot yang ada menerima rekomendasi penyeimbangan ulang, Auto Scaling Amazon EC2 hanya akan meluncurkan instans baru jika instans baru memberikan ketersediaan yang sama atau lebih baik daripada instans yang ada. Jika risiko gangguan instans baru akan lebih buruk daripada instans yang ada, maka Amazon EC2 Auto Scaling tidak akan meluncurkan instans baru. Namun, Amazon EC2 Auto Scaling akan terus menilai kumpulan kapasitas Spot berdasarkan informasi yang diberikan oleh layanan Amazon EC2 Spot, dan akan meluncurkan instans baru jika ketersediaan meningkat.

Ada kemungkinan instans Anda yang ada akan terganggu tanpa Auto Scaling Amazon EC2 secara proaktif meluncurkan instans baru. Ketika ini terjadi, Amazon EC2 Auto Scaling mencoba meluncurkan instans baru segera setelah menerima pemberitahuan gangguan Instans Spot. Ini terjadi terlepas dari apakah instance baru memiliki risiko gangguan yang tinggi.

Penyeimbangan Ulang Kapasitas tidak meningkatkan tingkat interupsi Instans Spot Anda

Saat Anda mengaktifkan Penyeimbangan Ulang Kapasitas, hal tersebut tidak meningkatkan [tingkat interupsi Instans Spot](#) Anda (jumlah Instans Spot yang diklaim kembali saat Amazon EC2 membutuhkan kapasitas kembali). Namun, jika Capacity Rebalancing mendeteksi instans berisiko mengalami gangguan, Auto Scaling Amazon EC2 akan segera mencoba meluncurkan instans baru. Oleh karena itu, lebih banyak instans mungkin diganti daripada jika Anda menunggu Auto Scaling Amazon EC2 meluncurkan instans baru setelah instans berisiko terputus.

Meskipun Anda mungkin mengganti lebih banyak instance dengan Capacity Rebalancing diaktifkan, Anda mendapat manfaat dari menjadi proaktif daripada reaktif. Ini memberi Anda lebih banyak waktu untuk mengambil tindakan sebelum instans Anda terganggu. Dengan [pemberitahuan interupsi Instans Spot](#), Anda biasanya hanya memiliki waktu hingga dua menit untuk mematikan instans Anda dengan baik. Dengan Capacity Rebalancing meluncurkan instans baru sebelumnya, Anda memberikan kesempatan yang lebih baik untuk menyelesaikan proses

yang ada pada instans berisiko Anda. Anda juga dapat memulai prosedur penghentian instans, mencegah pekerjaan baru dijadwalkan pada instans berisiko Anda, dan menyiapkan instans yang baru diluncurkan untuk mengambil alih aplikasi. Dengan penggantian proaktif dalam Penyeimbangan Kembali Kapasitas, Anda mendapat manfaat dari kontinuitas yang anggun.

Contoh teoritis berikut menunjukkan risiko dan manfaat menggunakan Capacity Rebalancing:

- 14:00 - Rekomendasi penyeimbangan ulang diterima misalnya A. Auto Scaling Amazon EC2 segera mencoba meluncurkan instans pengganti B, memberi Anda waktu untuk memulai prosedur shutdown Anda.
- 2:30 PM — Rekomendasi penyeimbangan ulang diterima misalnya B, yang diganti dengan contoh C. Ini memberi Anda waktu untuk memulai prosedur shutdown Anda.
- 14:32 — Jika Penyeimbangan Kembali Kapasitas tidak diaktifkan, dan jika pemberitahuan gangguan Instans Spot akan diterima pada pukul 14:32 misalnya A, Anda hanya memiliki dua menit untuk mengambil tindakan. Namun, Instance A akan terus berjalan hingga saat ini.

Aktifkan Rebalancing Kapasitas (konsol)

Anda dapat mengaktifkan atau menonaktifkan Capacity Rebalancing saat membuat atau memperbarui grup Auto Scaling.

Untuk mengaktifkan Capacity Rebalancing untuk grup Auto Scaling baru

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih Buat grup Auto Scaling.
3. Untuk Langkah 1: Pilih template peluncuran atau konfigurasi, masukkan nama untuk grup Auto Scaling, pilih template peluncuran, lalu pilih Berikutnya untuk melanjutkan ke langkah berikutnya.
4. Untuk Langkah 2: Pilih opsi peluncuran instance, untuk persyaratan jenis Instance, pilih pengaturan untuk membuat grup instance campuran. Ini termasuk jenis instans yang dapat diluncurkan, opsi pembelian instans, dan strategi alokasi untuk Instans Spot dan Sesuai Permintaan. Secara default, pengaturan ini tidak dikonfigurasi. Untuk mengonfigurasinya, Anda harus memilih Override launch template. Untuk informasi selengkapnya tentang membuat grup instance campuran, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).
5. Di bawah Jaringan, pilih opsi yang diinginkan. Verifikasi bahwa subnet yang ingin Anda gunakan berada di Availability Zone yang berbeda.

6. Di bawah bagian Strategi alokasi, pilih strategi alokasi Spot. Aktifkan atau nonaktifkan Penyeimbangan Kembali Kapasitas dengan memilih atau menghapus kotak centang di bawah Penyeimbangan Kembali Kapasitas. Anda hanya melihat opsi ini ketika Anda meminta persentase grup Auto Scaling untuk diluncurkan sebagai Instans Spot di bagian Opsi pembelian Instans.
7. Buat grup Auto Scaling.
8. (Opsional) Tambahkan kait siklus hidup sesuai kebutuhan. Untuk informasi selengkapnya, lihat [Tambahkan hook siklus aktif](#).

Untuk mengaktifkan atau menonaktifkan Penyeimbangan Ulang Kapasitas untuk grup Auto Scaling yang ada

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda. Panel split terbuka di bagian bawah halaman.
3. Pada tab Detail, pilih Strategi alokasi, Edit.
4. Di bawah bagian Strategi alokasi, aktifkan atau nonaktifkan Penyeimbangan Kembali Kapasitas dengan memilih atau menghapus kotak centang di bawah Penyeimbangan Kembali Kapasitas.
5. Pilih Perbarui.

Aktifkan Rebalancing Kapasitas ()AWS CLI

Contoh berikut menunjukkan cara menggunakan untuk mengaktifkan dan menonaktifkan Capacity Rebalancing. AWS CLI

Gunakan perintah [create-auto-scaling-group](#) atau [update-auto-scaling-group](#) dengan parameter berikut:

- `--capacity-rebalance/--no-capacity-rebalance`— Nilai Boolean yang menunjukkan apakah Capacity Rebalancing diaktifkan.

Sebelum memanggil perintah [create-auto-scaling-group](#), Anda perlu nama templat peluncuran yang dikonfigurasi untuk digunakan dengan grup Auto Scaling. Untuk informasi selengkapnya, lihat [Membuat template peluncuran untuk grup Auto Scaling](#).

Note

Prosedur berikut menunjukkan cara menggunakan file konfigurasi dalam format JSON atau YAML. Jika Anda menggunakan AWS CLI versi 1, Anda harus menentukan file konfigurasi berformat JSON. Jika Anda menggunakan AWS CLI versi 2, Anda dapat menentukan file konfigurasi yang diformat dalam YAMAL atau JSON.

JSON

Untuk membuat dan mengonfigurasi grup Auto Scaling baru

- Gunakan [create-auto-scaling-group](#) perintah berikut untuk membuat grup Auto Scaling baru dan aktifkan Capacity Rebalancing. Perintah ini mereferensikan file JSON sebagai satu-satunya parameter untuk grup Auto Scaling Anda.

```
aws autoscaling create-auto-scaling-group --cli-input-json file://~/config.json
```

Jika Anda belum memiliki file konfigurasi CLI yang menentukan [kebijakan bauran instance](#), buat file tersebut.

Tambahkan baris berikut ke objek JSON level-atas di file konfigurasi.

```
{  
  "CapacityRebalance": true  
}
```

Berikut ini adalah contoh file `config.json`.

```
{  
  "AutoScalingGroupName": "my-asg",  
  "DesiredCapacity": 12,  
  "MinSize": 12,  
  "MaxSize": 15,  
  "CapacityRebalance": true,  
  "MixedInstancesPolicy": {  
    "InstancesDistribution": {  
      "OnDemandBaseCapacity": 0,  
      "OnDemandPercentageAboveBaseCapacity": 25,  
      "SpotAllocationStrategy": "price-capacity-optimized"    }  
  }  
}
```

```
    },
    "LaunchTemplate": {
      "LaunchTemplateSpecification": {
        "LaunchTemplateName": "my-launch-template",
        "Version": "$Default"
      },
      "Overrides": [
        {
          "InstanceType": "c5.large"
        },
        {
          "InstanceType": "c5a.large"
        },
        {
          "InstanceType": "m5.large"
        },
        {
          "InstanceType": "m5a.large"
        },
        {
          "InstanceType": "c4.large"
        },
        {
          "InstanceType": "m4.large"
        },
        {
          "InstanceType": "c3.large"
        },
        {
          "InstanceType": "m3.large"
        }
      ]
    },
    "TargetGroupARNs": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-alb-target-group/943f017f100becff",
    "VPCZoneIdentifier": "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
  }
}
```

YAML

Untuk membuat dan mengonfigurasi grup Auto Scaling baru

- Gunakan [create-auto-scaling-group](#) perintah berikut untuk membuat grup Auto Scaling baru dan aktifkan Capacity Rebalancing. Perintah ini mereferensikan file YAMAL sebagai satu-satunya parameter untuk grup Auto Scaling Anda.

```
aws autoscaling create-auto-scaling-group --cli-input-yaml file://~/config.yaml
```

Tambahkan baris berikut ke file konfigurasi Anda dalam format YAML.

```
CapacityRebalance: true
```

Berikut ini adalah contoh file `config.yaml`.

```
---
AutoScalingGroupName: my-asg
DesiredCapacity: 12
MinSize: 12
MaxSize: 15
CapacityRebalance: true
MixedInstancesPolicy:
  InstancesDistribution:
    OnDemandBaseCapacity: 0
    OnDemandPercentageAboveBaseCapacity: 25
    SpotAllocationStrategy: price-capacity-optimized
  LaunchTemplate:
    LaunchTemplateSpecification:
      LaunchTemplateName: my-launch-template
      Version: $Default
    Overrides:
      - InstanceType: c5.large
      - InstanceType: c5a.large
      - InstanceType: m5.large
      - InstanceType: m5a.large
      - InstanceType: c4.large
      - InstanceType: m4.large
      - InstanceType: c3.large
      - InstanceType: m3.large
TargetGroupARNs:
```

```
- arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/my-alb-target-  
group/943f017f100becff  
VPCZoneIdentifier: subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782
```

Untuk mengaktifkan Capacity Rebalancing untuk grup Auto Scaling yang ada

- Gunakan perintah [update-auto-scaling-group](#) berikut ini untuk mengaktifkan Capacity Rebalancing.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
--capacity-rebalance
```

Untuk memverifikasi bahwa Capacity Rebalancing sudah diaktifkan untuk grup Auto Scaling

- Gunakan perintah [describe-auto-scaling-groups](#) berikut ini untuk memverifikasi bahwa Capacity Rebalancing diaktifkan dan untuk melihat detail.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

Berikut ini adalah contoh respons.

```
{  
  "AutoScalingGroups": [  
    {  
      "AutoScalingGroupName": "my-asg",  
      "AutoScalingGroupARN": "arn",  
      ...  
      "CapacityRebalance": true  
    }  
  ]  
}
```

Untuk menonaktifkan Capacity Rebalancing

Gunakan perintah [update-auto-scaling-group](#) dengan opsi `--no-capacity-rebalance` untuk menonaktifkan Capacity Rebalancing.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
--no-capacity-rebalance
```

```
--no-capacity-rebalance
```

Sumber daya terkait

Untuk informasi selengkapnya tentang Penyeimbangan Kembali Kapasitas, lihat [Mengelola siklus hidup Instans Spot secara proaktif menggunakan fitur Rebalancing Kapasitas baru untuk EC2 Auto Scaling](#) di Blog Komputasi. AWS

Untuk informasi selengkapnya tentang rekomendasi penyeimbangan ulang instans EC2, lihat rekomendasi [penyeimbangan ulang instans EC2 di Panduan Pengguna](#) Amazon EC2 untuk Instans Linux.

Untuk mempelajari lebih lanjut tentang kait siklus hidup, lihat sumber daya berikut.

- [Tutorial: Konfigurasi kait siklus hidup yang memanggil fungsi Lambda](#) (menggunakan EventBridge)
- [Tutorial: Konfigurasi data pengguna untuk mengambil status siklus hidup target melalui metadata instance](#)

Batasan

- Auto Scaling Amazon EC2 dapat menggantikan instans yang menerima pemberitahuan penyeimbangan ulang hanya jika instans tidak dilindungi dari skala masuk. Namun, perlindungan scale-in tidak mencegah penghentian dari gangguan Spot. Untuk informasi selengkapnya, lihat [Gunakan perlindungan skala dalam instance](#).
- Support for Capacity Rebalancing tersedia di semua iklan Wilayah AWS di mana Amazon EC2 Auto Scaling tersedia, kecuali untuk Wilayah Timur Tengah (UEA).

Gunakan Reservasi Kapasitas Sesuai Permintaan untuk memesan kapasitas di Availability Zone tertentu

Reservasi Kapasitas Sesuai Permintaan Amazon EC2 membantu Anda memesan kapasitas komputasi di Availability Zone tertentu. Untuk mulai menggunakan Reservasi Kapasitas, Anda membuat Reservasi Kapasitas di Availability Zone tertentu. Kemudian, Anda dapat meluncurkan instans ke dalam kapasitas terpesan, melihat pemanfaatan kapasitasnya dalam waktu nyata, dan menambah atau mengurangi kapasitasnya sesuai kebutuhan.

Reservasi Kapasitas dikonfigurasi sebagai open atau targeted. Jika Reservasi Kapasitas adalah open, semua instans baru dan yang sudah ada yang memiliki atribut yang cocok secara otomatis berjalan dalam kapasitas Reservasi Kapasitas. Jika Reservasi Kapasitas targeted, instans harus secara khusus menargetkannya untuk dijalankan dalam kapasitas terpesan.

Topik ini menunjukkan cara membuat grup Auto Scaling yang meluncurkan Instans Sesuai Permintaan ke Reservasi Kapasitas. targeted Ini memberi Anda kontrol lebih besar atas kapan harus menggunakan Reservasi Kapasitas tertentu.

Langkah dasarnya adalah:

1. Buat Reservasi Kapasitas di beberapa Availability Zone yang memiliki jenis instans, platform, dan jumlah instans yang sama.
2. Reservasi Kapasitas Grup menggunakan AWS Resource Groups.
3. Buat grup Auto Scaling dengan templat peluncuran yang menargetkan grup sumber daya, menggunakan Availability Zone yang sama dengan Reservasi Kapasitas.

Daftar Isi

- [Langkah 1: Buat Reservasi Kapasitas](#)
- [Langkah 2: Buat grup Reservasi Kapasitas](#)
- [Langkah 3: Buat template peluncuran](#)
- [Langkah 4: Buat grup Auto Scaling](#)
- [Sumber daya terkait](#)

Langkah 1: Buat Reservasi Kapasitas

Langkah pertama adalah membuat Reservasi Kapasitas di setiap Availability Zone tempat grup Auto Scaling Anda akan digunakan.

Note

Anda hanya dapat membuat targeted reservasi saat pertama kali membuat Reservasi Kapasitas.

Console

Untuk membuat Reservasi Kapasitas

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pilih Reservasi Kapasitas, lalu pilih Buat Reservasi Kapasitas.
3. Pada halaman Buat Reservasi Kapasitas, perhatikan pengaturan berikut di bagian Detail instans. Tipe instans, platform, dan Zona Ketersediaan dari instans yang Anda luncurkan harus cocok dengan tipe instans, platform, dan Zona Ketersediaan yang Anda tentukan di sini atau Reservasi Kapasitas tidak akan diterapkan.
 - a. Untuk tipe Instance, pilih jenis instance yang akan diluncurkan ke kapasitas cadangan.
 - b. Untuk Platform, pilih sistem operasi untuk instans Anda.
 - c. Untuk Availability Zone, pilih Availability Zone pertama yang ingin Anda pesan kapasitasnya.
 - d. Untuk Kapasitas total, pilih jumlah instans yang Anda butuhkan. Hitung jumlah total instans yang Anda perlukan untuk grup Auto Scaling dibagi dengan jumlah Availability Zone yang akan Anda gunakan.
4. Di bawah rincian Reservasi Kapasitas, untuk Reservasi Kapasitas berakhir, pilih salah satu opsi berikut:
 - Pada waktu tertentu — Batalkan Reservasi Kapasitas secara otomatis pada tanggal dan waktu yang ditentukan.
 - Secara manual — Pesan kapasitas sampai Anda membatalkannya secara eksplisit.
5. Untuk kelayakan Instans, pilih Targeted: Hanya instance yang menargetkan Reservasi Kapasitas.
6. (Opsional) Untuk Tag, tentukan tag apa pun yang akan dikaitkan dengan Reservasi Kapasitas.
7. Pilih Buat.
8. Catat ID Reservasi Kapasitas yang baru dibuat. Anda memerlukannya untuk mengatur grup Reservasi Kapasitas.

Ulangi prosedur ini untuk setiap Availability Zone yang ingin Anda aktifkan untuk grup Auto Scaling Anda, hanya mengubah nilai opsi Availability Zone.

AWS CLI

Untuk membuat Reservasi Kapasitas

Gunakan [create-capacity-reservation](#) perintah berikut untuk membuat Reservasi Kapasitas. Ganti nilai sampel untuk `--availability-zone`, `--instance-type`, `--instance-platform`, dan `--instance-count`.

```
aws ec2 create-capacity-reservation \  
  --availability-zone us-east-1a \  
  --instance-type c5.xlarge \  
  --instance-platform Linux/UNIX \  
  --instance-count 3 \  
  --instance-match-criteria targeted
```

Contoh ID Reservasi Kapasitas yang dihasilkan

```
{  
  "CapacityReservation": {  
    "CapacityReservationId": "cr-1234567890abcdef1",  
    "OwnerId": "123456789012",  
    "CapacityReservationArn": "arn:aws:ec2:us-east-1:123456789012:capacity-  
reservation/cr-1234567890abcdef1",  
    "InstanceType": "c5.xlarge",  
    "InstancePlatform": "Linux/UNIX",  
    "AvailabilityZone": "us-east-1a",  
    "Tenancy": "default",  
    "TotalInstanceCount": 3,  
    "AvailableInstanceCount": 3,  
    "EbsOptimized": false,  
    "EphemeralStorage": false,  
    "State": "active",  
    "StartDate": "2023-07-26T21:36:14+00:00",  
    "EndDateType": "unlimited",  
    "InstanceMatchCriteria": "targeted",  
    "CreateDate": "2023-07-26T21:36:14+00:00"  
  }  
}
```

Catat ID Reservasi Kapasitas yang baru dibuat. Anda memerlukannya untuk mengatur grup Reservasi Kapasitas.

Ulangi perintah ini untuk setiap Availability Zone yang ingin Anda aktifkan untuk grup Auto Scaling Anda, hanya mengubah nilai opsi. `--availability-zone`

Langkah 2: Buat grup Reservasi Kapasitas

Setelah selesai membuat Reservasi Kapasitas, Anda dapat mengelompokkannya bersama-sama menggunakan layanan AWS Resource Groups. AWS Resource Groups mendukung beberapa jenis grup yang berbeda untuk penggunaan yang berbeda. Amazon EC2 menggunakan grup tujuan khusus, yang dikenal sebagai grup sumber daya terkait layanan, untuk menargetkan grup Reservasi Kapasitas. Untuk berinteraksi dengan grup sumber daya terkait layanan ini, Anda dapat menggunakan AWS CLI atau SDK tetapi bukan konsol. Untuk informasi selengkapnya tentang grup sumber daya terkait layanan, lihat [Konfigurasi layanan untuk grup sumber daya](#) di Panduan Pengguna AWS Resource Groups.

Untuk membuat grup Reservasi Kapasitas menggunakan AWS CLI

Gunakan perintah [create-group untuk membuat grup](#) sumber daya yang hanya dapat berisi Reservasi Kapasitas. Dalam contoh ini, grup sumber daya diberi nama *my-cr-group*.

```
aws resource-groups create-group \  
  --name my-cr-group \  
  --configuration '{"Type":"AWS::EC2::CapacityReservationPool"}'  
'{"Type":"AWS::ResourceGroups::Generic", "Parameters": [{"Name": "allowed-resource-  
types", "Values": ["AWS::EC2::CapacityReservation"]}]]'
```

Berikut ini adalah contoh respons.

```
{  
  "Group": {  
    "GroupArn": "arn:aws:resource-groups:us-east-1:123456789012:group/my-cr-group",  
    "Name": "my-cr-group"  
  },  
  "GroupConfiguration": {  
    "Configuration": [  
      {  
        "Type": "AWS::EC2::CapacityReservationPool"  
      },  
      {  
        "Type": "AWS::ResourceGroups::Generic",  
        "Parameters": [  

```

```

        {
            "Name": "allowed-resource-types",
            "Values": [
                "AWS::EC2::CapacityReservation"
            ]
        }
    ],
    "Status": "UPDATE_COMPLETE"
}
}

```

Perhatikan ARN dari grup sumber daya. Anda memerlukannya untuk mengatur template peluncuran untuk grup Auto Scaling Anda.

Untuk mengaitkan Reservasi Kapasitas Anda ke grup yang baru dibuat menggunakan AWS CLI

Gunakan perintah [group-resources](#) berikut untuk mengaitkan Reservasi Kapasitas ke grup Reservasi Kapasitas yang baru dibuat. Untuk `--resource-arns` opsi, tentukan Reservasi Kapasitas menggunakan ARN mereka. Buat ARN menggunakan Wilayah yang relevan, ID akun Anda, dan ID reservasi yang Anda catat sebelumnya. Dalam contoh ini, reservasi dengan ID `cr-1234567890abcdef1` dan `cr-54321abcdef567890` akan dikelompokkan bersama dalam grup bernama `my-cr-group`

```

aws resource-groups group-resources \
  --group my-cr-group \
  --resource-arns \
    arn:aws:ec2:region:account-id:capacity-reservation/cr-1234567890abcdef1 \
    arn:aws:ec2:region:account-id:capacity-reservation/cr-54321abcdef567890

```

Berikut ini adalah contoh respons.

```

{
  "Succeeded": [
    "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-1234567890abcdef1",
    "arn:aws:ec2:us-east-1:123456789012:capacity-reservation/cr-54321abcdef567890"
  ],
  "Failed": [],
  "Pending": []
}

```

Untuk informasi tentang memodifikasi atau menghapus grup sumber daya, lihat Referensi [API AWS Resource Groups](#).

Langkah 3: Buat template peluncuran

Console

Untuk membuat templat peluncuran

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Pada panel navigasi, di bawah Instans, pilih Luncurkan Template.
3. Pilih Buat templat peluncuran. Masukkan nama dan berikan deskripsi untuk versi awal templat peluncuran.
4. Di bawah panduan Auto Scaling, pilih kotak centang.
5. Buat templat peluncuran . Pilih AMI dan jenis instans yang sesuai dengan Reservasi Kapasitas yang akan Anda gunakan, dan secara opsional, key pair, satu atau beberapa grup keamanan, dan volume EBS tambahan atau volume penyimpanan instans untuk instans Anda.
6. Perluas Detail lanjutan, dan lakukan hal berikut:
 - a. Untuk Reservasi Kapasitas, pilih Target berdasarkan grup.
 - b. Untuk reservasi Kapasitas - Targetkan berdasarkan grup, pilih grup Reservasi Kapasitas yang Anda buat di bagian sebelumnya, lalu pilih Simpan.
7. Pilih Buat templat peluncuran.
8. Pada halaman konfirmasi, pilih Buat grup Auto Scaling.

AWS CLI

Untuk membuat templat peluncuran

Gunakan [create-launch-template](#) perintah berikut untuk membuat template peluncuran yang menentukan bahwa Reservasi Kapasitas menargetkan grup sumber daya tertentu. Ganti nilai sampel untuk `--launch-template-name`. Ganti `c5.xlarge` dengan jenis instans yang Anda gunakan dalam Reservasi Kapasitas dan `ami-0123456789EXAMPLE` dengan ID AMI yang ingin Anda gunakan. Ganti `arn:aws:resource-groups:region:account-id:group/my-cr-group` dengan ARN dari grup sumber daya yang Anda buat di awal bagian sebelumnya.

```
aws ec2 create-launch-template \  
  --launch-template-name my-launch-template \  
  --launch-template-data \  
    '{"InstanceType": "c5.xlarge",  
     "ImageId": "ami-0123456789EXAMPLE",  
     "CapacityReservationSpecification":  
       {"CapacityReservationTarget":  
         { "CapacityReservationResourceGroupArn": "arn:aws:resource-  
groups:region:account-id:group/my-cr-group" }  
       }  
    }'
```

Berikut ini adalah contoh respons.

```
{  
  "LaunchTemplate": {  
    "LaunchTemplateId": "lt-0dd77bd41dEXAMPLE",  
    "LaunchTemplateName": "my-launch-template",  
    "CreateTime": "2023-07-26T21:42:48+00:00",  
    "CreatedBy": "arn:aws:iam::123456789012:user/Bob",  
    "DefaultVersionNumber": 1,  
    "LatestVersionNumber": 1  
  }  
}
```

Langkah 4: Buat grup Auto Scaling

Console

Buat grup Auto Scaling Anda seperti biasanya, tetapi ketika Anda memilih subnet VPC, pilih subnet dari setiap Availability Zone yang cocok dengan Reservasi Kapasitas yang Anda buat. targeted Kemudian, saat grup Auto Scaling Anda meluncurkan Instans Sesuai Permintaan di salah satu Availability Zone ini, instance akan dijalankan dalam kapasitas cadangan untuk Availability Zone tersebut. Jika grup sumber daya kehabisan Reservasi Kapasitas sebelum kapasitas yang Anda inginkan terpenuhi, kami meluncurkan apa pun di luar kapasitas cadangan sebagai kapasitas On-Demand reguler.

Untuk membuat grup Auto Scaling sederhana

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pada bilah navigasi di bagian atas layar, pilih yang sama dengan Wilayah AWS yang Anda gunakan saat membuat templat peluncuran.
3. Pilih Buat grup Auto Scaling.
4. Pada halaman Pilih templat peluncuran atau konfigurasi, untuk Nama grup Auto Scaling, masukkan nama untuk grup Auto Scaling Anda.
5. Untuk Templat peluncuran, pilih templat peluncuran yang ada.
6. Untuk Versi templat peluncuran, pilih apakah grup Auto Scaling menggunakan default, terbaru, atau versi spesifik templat peluncuran saat menskalakan.
7. Pada halaman Opsi peluncuran instance Choose, lewati bagian Persyaratan tipe instans untuk menggunakan tipe instans EC2 yang ditentukan dalam template peluncuran.
8. Di bawah Jaringan, untuk VPC, pilih VPC. Grup Auto Scaling harus dibuat dalam VPC yang sama dengan grup keamanan yang Anda tentukan dalam template peluncuran Anda. Jika Anda tidak menentukan grup keamanan dalam template peluncuran, Anda dapat memilih VPC mana pun yang memiliki subnet di Availability Zone yang sama dengan Reservasi Kapasitas Anda.
9. Untuk Zona Ketersediaan dan subnet, pilih subnet dari setiap Availability Zone yang ingin Anda sertakan, berdasarkan Availability Zone tempat Reservasi Kapasitas Anda berada.
10. Pilih Selanjutnya dua kali.
11. Pada halaman Konfigurasi ukuran grup dan kebijakan penskalaan, untuk kapasitas yang diinginkan, masukkan jumlah awal instance yang akan diluncurkan. Ketika Anda mengubah angka ini ke nilai di luar batas kapasitas minimum atau maksimum, Anda harus memperbarui nilai Kapasitas minimum atau Kapasitas maksimum. Untuk informasi selengkapnya, lihat [Tetapkan batas penskalaan untuk grup Auto Scaling](#).
12. Pilih Lewati untuk meninjau.
13. Pada halaman Peninjauan, pilih Buat grup Auto Scaling.

AWS CLI

Untuk membuat grup Auto Scaling sederhana

Gunakan [create-auto-scaling-group](#) perintah berikut dan tentukan nama dan versi template peluncuran Anda sebagai nilai untuk `--launch-template` opsi. Ganti nilai sampel untuk `--auto-scaling-group-name`, `--min-size`, `--max-size`, dan `--vpc-zone-identifier`.

Untuk `--availability-zones` opsi, tentukan Availability Zone yang Anda buat untuk Reservasi Kapasitas. Misalnya, jika Reservasi Kapasitas menentukan `us-east-1a` dan `us-east-1b` Availability Zone, maka Anda harus membuat grup Auto Scaling di zona yang sama. Kemudian, saat grup Auto Scaling Anda meluncurkan Instans Sesuai Permintaan di salah satu Availability Zone ini, instance akan dijalankan dalam kapasitas cadangan untuk Availability Zone tersebut. Jika grup sumber daya kehabisan Reservasi Kapasitas sebelum kapasitas yang Anda inginkan terpenuhi, kami meluncurkan apa pun di luar kapasitas cadangan sebagai kapasitas On-Demand reguler.

```
aws autoscaling create-auto-scaling-group \  
  --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateName=my-launch-template,Version='1' \  
  --min-size 6 \  
  --max-size 6 \  
  --vpc-zone-identifier "subnet-5f46ec3b,subnet-0ecac448" \  
  --availability-zones us-east-1a us-east-1b
```

Sumber daya terkait

Untuk implementasi contoh, lihat AWS CloudFormation template di GitHub repositori AWS sampel berikut: <https://github.com/aws-samples/aws-auto-scaling-backed-by-reservations-by-on-demand-capacity>.

Topik terkait berikut dapat membantu saat Anda mempelajari tentang Reservasi Kapasitas.

- Reservasi Kapasitas Sesuai Permintaan
 - [Membuat Reservasi Kapasitas](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux
 - [Reservasi Kapasitas Sesuai Permintaan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux
 - [Targetkan grup Reservasi Kapasitas Sesuai Permintaan Amazon EC2 di Blog Operasi & Migrasi AWS Cloud](#)
- Blok Kapasitas (reservasi kapasitas dengan durasi yang ditentukan)
 - [Blok Kapasitas untuk ML](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux

- [Gunakan Blok Kapasitas untuk beban kerja pembelajaran mesin](#)

Buat grup Auto Scaling dari baris perintah menggunakan AWS CloudShell

Di [didukung Wilayah AWS](#), Anda dapat menjalankan AWS CLI perintah menggunakan AWS CloudShell shell pra-autentikasi berbasis browser yang diluncurkan langsung dari file. AWS Management Console Anda dapat menjalankan AWS CLI perintah terhadap layanan menggunakan shell pilihan Anda (Bash, PowerShell, atau Z shell).

Anda dapat memulai AWS CloudShell dari AWS Management Console menggunakan salah satu dari dua metode berikut:

- Pilih AWS CloudShell ikon di bilah navigasi konsol. Itu terletak di sebelah kanan kotak pencarian.
- Gunakan kotak pencarian di bilah navigasi konsol untuk mencari CloudShell dan kemudian pilih CloudShellopsi.

Saat AWS CloudShell diluncurkan di jendela browser baru untuk pertama kalinya, panel selamat datang menampilkan dan mencantumkan fitur-fitur utama. Setelah Anda menutup panel ini, pembaruan status disediakan saat shell mengonfigurasi dan meneruskan kredensi konsol Anda. Ketika command prompt ditampilkan, shell siap untuk interaksi.

Untuk informasi selengkapnya tentang layanan ini, lihat [Panduan Pengguna AWS CloudShell](#).

Buat grup Auto Scaling dengan AWS CloudFormation

Auto Scaling Amazon EC2 terintegrasi AWS CloudFormation dengan, layanan yang membantu Anda memodelkan dan menyiapkan sumber daya sehingga AWS Anda dapat menghabiskan lebih sedikit waktu untuk membuat dan mengelola sumber daya dan infrastruktur Anda. Anda membuat templat yang menjelaskan semua AWS sumber daya yang Anda inginkan (seperti grup Auto Scaling), serta menyediakan serta AWS CloudFormation mengonfigurasi sumber daya tersebut untuk Anda.

Saat menggunakan AWS CloudFormation, Anda dapat menggunakan kembali template untuk menyiapkan sumber daya Auto Scaling Amazon EC2 secara konsisten dan berulang kali. Jelaskan sumber daya Anda sekali, lalu sediakan sumber daya yang sama berulang-ulang di beberapa Akun AWS dan Wilayah.

Auto Scaling dan template Amazon EC2 AWS CloudFormation

[Untuk menyediakan dan mengonfigurasi sumber daya untuk Auto Scaling Amazon EC2 dan layanan terkait, Anda harus memahami templat.AWS CloudFormation](#) Templat adalah file teks dengan format JSON atau YAML. Template ini menjelaskan sumber daya yang ingin Anda sediakan di AWS CloudFormation tumpukan Anda. Jika Anda tidak terbiasa dengan JSON atau YAMAL, Anda dapat menggunakan AWS CloudFormation Designer untuk membantu Anda memulai dengan template. AWS CloudFormation Untuk informasi lebih lanjut, lihat [Apa itu AWS CloudFormation Desainer?](#) dalam AWS CloudFormation User Guide.

Untuk mulai membuat template tumpukan Anda sendiri untuk Auto Scaling Amazon EC2, selesaikan tugas-tugas berikut:

- Buat template peluncuran menggunakan [AWS::EC2::LaunchTemplate](#).

-

Untuk panduan yang menunjukkan cara menerapkan grup Auto Scaling di belakang Application Load Balancer, lihat [Panduan: Membuat aplikasi yang diskalakan dan diseimbangkan beban di Panduan Pengguna.AWS CloudFormation](#)

Anda dapat menemukan contoh berguna tambahan dari cuplikan templat yang membuat grup Auto Scaling dan sumber daya terkait di bagian AWS CloudFormation Panduan Pengguna berikut:

- Referensi jenis sumber daya [Penskalaan Otomatis Amazon EC2 Referensi jenis sumber daya Amazon](#)
- [Konfigurasi sumber daya Auto Scaling Amazon EC2 dengan AWS CloudFormation](#)

Pelajari lebih lanjut tentang AWS CloudFormation

Untuk mempelajari selengkapnya AWS CloudFormation, lihat sumber daya berikut:

- [AWS CloudFormation](#)
- [AWS CloudFormation Panduan Pengguna](#)
- [AWS CloudFormation Referensi API](#)
- [AWS CloudFormation Panduan Pengguna Antarmuka Baris Perintah](#)

Gunakan AWS Compute Optimizer untuk mendapatkan rekomendasi untuk jenis instans untuk grup Auto Scaling

AWS memberikan rekomendasi instans Amazon EC2 untuk membantu Anda meningkatkan kinerja, menghemat uang, atau keduanya, dengan menggunakan fitur yang didukung oleh AWS Compute Optimizer Anda dapat menggunakan rekomendasi ini untuk memutuskan apakah akan beralih ke tipe instans yang baru.

Untuk membuat rekomendasi, Compute Optimizer menganalisis spesifikasi instans yang ada dan riwayat metrik terbaru. Data yang dikumpulkan kemudian digunakan untuk merekomendasikan jenis instans Amazon EC2 mana yang paling optimal untuk menangani beban kerja performa yang ada. Rekomendasi dikembalikan bersama dengan harga instans per jam.

Note

Untuk mendapatkan rekomendasi dari Compute Optimizer, Anda harus terlebih dahulu memilih Compute Optimizer. Untuk informasi selengkapnya, lihat [Memulai dengan AWS Compute Optimizer](#) dalam Panduan Pengguna AWS Compute Optimizer .

Daftar Isi

- [Batasan](#)
- [Temuan](#)
- [Melihat rekomendasi](#)
- [Pertimbangan untuk mengevaluasi rekomendasi](#)

Batasan

Compute Optimizer menghasilkan rekomendasi untuk instans dalam grup Auto Scaling yang dikonfigurasi untuk meluncurkan dan menjalankan jenis instans M, C, R, T, dan X. Namun, itu tidak menghasilkan rekomendasi untuk tipe instans -g yang didukung oleh prosesor AWS Graviton2 (misalnya, C6g), dan untuk tipe instans -n yang memiliki kinerja bandwidth jaringan yang lebih tinggi (misalnya, M5n).

Grup Auto Scaling juga harus dikonfigurasi untuk menjalankan satu jenis instans tunggal (yaitu, tidak ada jenis instans campuran), tidak boleh memiliki kebijakan penskalaan yang melekat padanya, dan

memiliki nilai yang sama untuk kapasitas minimum dan maksimum yang diinginkan (yaitu, kelompok Auto Scaling dengan jumlah instans tetap). Compute Optimizer menghasilkan rekomendasi untuk instans dalam grup Auto Scaling yang memenuhi semua persyaratan konfigurasi ini.

Temuan

Compute Optimizer mengklasifikasikan temuannya untuk grup Auto Scaling sebagai berikut:

- Tidak dioptimalkan – Grup Auto Scaling dianggap tidak optimal ketika Compute Optimizer telah mengidentifikasi rekomendasi yang dapat memberikan performa yang lebih baik untuk beban kerja Anda.
- Dioptimalkan – Grup Auto Scaling dianggap optimal ketika Compute Optimizer menentukan bahwa grup disiapkan dengan tepat untuk menjalankan beban kerja Anda, berdasarkan jenis instans yang dipilih. Untuk sumber daya yang dioptimalkan, Compute Optimizer terkadang dapat merekomendasikan jenis instans generasi baru.
- Tidak ada – Tidak ada rekomendasi untuk grup Auto Scaling ini. Hal ini mungkin terjadi jika Anda telah memilih Compute Optimizer selama kurang dari 12 jam, atau jika grup Auto Scaling telah berjalan selama kurang dari 30 jam, atau ketika grup Auto Scaling atau jenis instans tidak didukung oleh Compute Optimizer. Untuk informasi selengkapnya, lihat bagian [Batasan](#).

Melihat rekomendasi

Setelah Anda memilih untuk Compute Optimizer, Anda dapat melihat temuan dan rekomendasi yang dihasilkannya untuk grup Auto Scaling Anda. Jika Anda baru saja memilih, rekomendasi mungkin tidak tersedia hingga 12 jam.

Untuk melihat rekomendasi yang dibuat untuk grup Auto Scaling

1. Buka konsol Compute Optimizer di <https://console.aws.amazon.com/compute-optimizer/>.

Halaman Dasbor terbuka.

2. Pilih Lihat rekomendasi untuk semua grup Auto Scaling.
3. Pilih grup Auto Scaling Anda.
4. Pilih Lihat detail.

Tampilan berubah untuk menampilkan hingga tiga rekomendasi instans yang berbeda dalam tampilan yang sudah dikonfigurasi, berdasarkan pengaturan tabel default. Ini juga menyediakan

data CloudWatch metrik terbaru (pemanfaatan CPU rata-rata, jaringan rata-rata masuk, dan jaringan rata-rata keluar) untuk grup Auto Scaling.

Tentukan apakah Anda ingin menggunakan salah satu rekomendasi. Tentukan apakah akan mengoptimalkan peningkatan performa, untuk pengurangan biaya, atau untuk kombinasi kedua hal ini.

Untuk mengubah jenis instans di grup Auto Scaling Anda, perbarui templat peluncuran atau perbarui grup Auto Scaling untuk menggunakan konfigurasi peluncuran baru. Proses yang ada tetap menggunakan konfigurasi sebelumnya. Untuk memperbarui instans yang sudah ada, akhiri sehingga digantikan oleh grup Auto Scaling Anda, atau biarkan penskalaan otomatis menggantikan instans yang lebih lama secara bertahap dengan instans yang lebih baru berdasarkan [kebijakan pengakhiran](#).

Note

Dengan masa aktif instans maksimum dan fitur refresh instans, Anda juga dapat mengganti instans yang sudah ada dalam grup Auto Scaling Anda untuk meluncurkan instans baru yang menggunakan templat peluncuran baru atau konfigurasi peluncuran baru. Untuk informasi selengkapnya, lihat [Ganti instans Auto Scaling berdasarkan masa pakai instans maksimum](#) dan [Menggunakan penyegaran instans untuk memperbarui instance dalam grup Auto Scaling](#).

Pertimbangan untuk mengevaluasi rekomendasi

Sebelum beralih ke jenis instans baru, pertimbangkan hal berikut:

- Rekomendasi tidak memprakirakan penggunaan Anda. Rekomendasi didasarkan pada penggunaan historis Anda selama periode waktu 14 hari terakhir. Pastikan untuk memilih jenis instans yang diharapkan untuk memenuhi kebutuhan penggunaan mendatang Anda.
- Fokus pada metrik grafik untuk menentukan apakah penggunaan aktual lebih rendah daripada kapasitas instans. Anda juga dapat melihat data metrik (rata-rata, puncak, persentil) CloudWatch untuk mengevaluasi lebih lanjut rekomendasi instans EC2 Anda. Misalnya, perhatikan cara metrik persentase CPU berubah sepanjang hari dan apakah ada puncak yang perlu diakomodasi. Untuk informasi selengkapnya, lihat [Melihat metrik yang tersedia](#) di Panduan CloudWatch Pengguna Amazon.

- Compute Optimizer mungkin menyediakan rekomendasi untuk instans performa yang dapat melonjak, yaitu instans T3, T3a, dan T2. Jika Anda secara berkala meledak di atas garis dasar Anda, pastikan Anda dapat terus melakukannya berdasarkan vCPU dari jenis instans baru. Untuk informasi selengkapnya, lihat [Kredit CPU dan performa garis dasar untuk instans performa yang dapat terpecah](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Jika Anda telah membeli Instans Tersimpan, Instans Sesuai Permintaan Anda mungkin akan ditagih sebagai Instans Tersimpan. Sebelum Anda mengubah jenis instans saat ini, pertama-tama evaluasi dampaknya terhadap penggunaan dan cakupan Instans Tersimpan.
- Pertimbangkan konversi ke instans generasi yang lebih baru, jika memungkinkan.
- Saat bermigrasi ke kelompok instans yang berbeda, pastikan jenis instans saat ini dan jenis instans baru kompatibel, misalnya dalam hal virtualisasi, arsitektur, atau jenis jaringan. Untuk informasi selengkapnya, lihat [Kompatibilitas untuk mengubah ukuran instans](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Terakhir, pertimbangkan penilaian risiko kinerja yang diberikan untuk setiap rekomendasi. Risiko kinerja menunjukkan jumlah upaya yang mungkin perlu Anda keluarkan untuk memvalidasi apakah tipe instans yang direkomendasikan memenuhi persyaratan kinerja beban kerja Anda. Kami juga menyarankan pengujian beban dan performa yang ketat sebelum dan setelah membuat perubahan apa pun.

Sumber daya tambahan

Selain topik di halaman ini, lihat sumber daya berikut:

- [Jenis Instans Amazon EC2](#)
- [AWS Compute Optimizer Panduan Pengguna](#)

Gunakan Elastic Load Balancing untuk mendistribusikan lalu lintas di seluruh instans dalam grup Auto Scaling

Elastic Load Balancing secara otomatis mendistribusikan lalu lintas aplikasi masuk Anda di semua instans EC2 yang sedang Anda jalankan. Elastic Load Balancing membantu mengelola permintaan masuk dengan merutekan lalu lintas secara optimal sehingga tidak ada satu instance yang kewalahan.

Untuk menggunakan Elastic Load Balancing dengan grup Auto Scaling Anda, [pasang load balancer ke grup Auto Scaling Anda](#). Ini mendaftarkan grup dengan penyeimbang beban, yang bertindak sebagai titik kontak tunggal untuk semua lalu lintas web yang masuk ke grup Auto Scaling Anda.

Bila Anda menggunakan Elastic Load Balancing dengan grup Auto Scaling Anda, Anda tidak perlu mendaftarkan instans EC2 individual ke load balancer. Instance yang diluncurkan oleh grup Auto Scaling Anda secara otomatis terdaftar dengan load balancer. Demikian pula, pendaftaran instance yang dihentikan oleh grup Auto Scaling Anda secara otomatis dikeluarkan dari load balancer.

Setelah melampirkan penyeimbang beban ke grup Auto Scaling, Anda dapat mengonfigurasi grup Auto Scaling untuk menggunakan metrik Elastic Load Balancing (seperti jumlah permintaan Application Load Balancer per target) untuk menskalakan jumlah instans dalam grup saat permintaan berfluktuasi.

Secara opsional, Anda dapat menambahkan pemeriksaan kesehatan Elastic Load Balancing ke grup Auto Scaling sehingga Auto Scaling Amazon EC2 dapat mengidentifikasi dan mengganti instans yang tidak sehat berdasarkan pemeriksaan kesehatan tambahan ini. Jika tidak, Anda dapat membuat CloudWatch alarm yang memberi tahu Anda jika jumlah host yang sehat dari grup target lebih rendah dari yang diizinkan.

Daftar Isi

- [Jenis Elastic Load Balancing](#)
- [Bersiaplah untuk memasang penyeimbang beban Elastic Load Balancing ke grup Auto Scaling Anda](#)
- [Pasang penyeimbang beban Elastic Load Balancing ke grup Auto Scaling Anda](#)
- [Konfigurasi Application Load Balancer atau Network Load Balancer dari konsol Auto Scaling Amazon EC2](#)
- [Verifikasi status lampiran penyeimbang beban Anda](#)
- [Menambahkan dan menghapus Availability Zones](#)
- [Contoh untuk bekerja dengan Elastic Load Balancing dengan \(\) AWS Command Line InterfaceAWS CLI](#)

Jenis Elastic Load Balancing

Elastic Load Balancing menyediakan empat jenis load balancer yang dapat digunakan bersama grup Auto Scaling Anda: Application Load Balancers, Network Load Balancers, Gateway Load Balancers, dan Classic Load Balancers.

Ada perbedaan utama dalam bagaimana jenis Load Balancer dikonfigurasi. Dengan Application Load Balancers, Network Load Balancers, dan Gateway Load Balancers, instance terdaftar sebagai target dengan grup target, dan Anda merutekan lalu lintas ke grup target. Dengan Classic Load Balancers, instans terdaftar langsung dengan load balancer.

Penyeimbang Beban Aplikasi

Perutean dan load balancer pada layer aplikasi (HTTP/HTTPS), dan mendukung rute berbasis jalur. Application Load Balancer dapat mengirimkan permintaan ke port di satu atau beberapa target yang terdaftar, seperti instance EC2, di virtual private cloud (VPC) Anda.

Network Load Balancer

Rute dan load balances pada layer transport (TCP/UDP Layer-4), berdasarkan informasi alamat yang diekstrak dari header Layer-4. Network Load Balancer dapat menangani rentetan lalu lintas, mempertahankan IP sumber klien, dan menggunakan IP tetap untuk masa aktif load balancer.

Gateway Load Balancer

Mendistribusikan lalu lintas ke armada instance alat. Menyediakan skala, ketersediaan, dan kesederhanaan untuk peralatan virtual pihak ketiga, seperti firewall, deteksi intrusi dan sistem pencegahan, dan peralatan lainnya. Gateway Load Balancer bekerja dengan peranti virtual yang mendukung protokol GENEVE. Integrasi teknis tambahan diperlukan, jadi pastikan untuk berkonsultasi dengan panduan pengguna sebelum memilih Load Balancer Gateway.

Classic Load Balancer

Rute dan load balancer baik di lapisan transportasi (TCP/SSL) maupun di lapisan aplikasi (HTTP/HTTPS).

Untuk mendapatkan pemahaman yang lebih dalam tentang berbagai jenis penyeimbang beban yang tersedia, lihat sumber daya berikut:

- [Apa itu Elastic Load Balancing?](#)
- [Apa itu Application Load Balancer?](#)
- [Apa itu Network Load Balancer?](#)
- [Apa itu Load Balancer Gateway?](#)
- [Apa itu Classic Load Balancer?](#)

Bersiaplah untuk memasang penyeimbang beban Elastic Load Balancing ke grup Auto Scaling Anda

Sebelum Anda memasang penyeimbang beban Elastic Load Balancing ke grup Auto Scaling Anda, Anda harus menyelesaikan prasyarat berikut:

- Anda harus sudah membuat penyeimbang beban dan grup target yang digunakan untuk merutekan lalu lintas ke grup Auto Scaling Anda.

Ada dua cara untuk membuat load balancer dan target group:

- Menggunakan Elastic Load Balancing — Ikuti prosedur dalam dokumentasi Elastic Load Balancing untuk membuat dan mengonfigurasi load balancer dan grup target sebelum membuat grup Auto Scaling. Lewati langkah untuk mendaftarkan instans Amazon EC2 Anda. Auto Scaling Amazon EC2 secara otomatis menangani pendaftaran (dan membatalkan pendaftaran) instans saat Anda melampirkan grup target ke grup Auto Scaling. Untuk informasi lebih lanjut, lihat [Memulai Elastic Load Balancing](#) di Panduan Pengguna Elastic Load Balancing.
- Menggunakan Auto Scaling Amazon EC2 — Buat, konfigurasi, dan pasang load balancer dan grup target dengan konfigurasi dasar dari konsol Auto Scaling Amazon EC2. Untuk informasi selengkapnya, lihat [Konfigurasi Application Load Balancer atau Network Load Balancer dari konsol Auto Scaling Amazon EC2](#).
- Sebelum membuat load balancer, ketahui jenis load balancer yang Anda butuhkan. Untuk informasi selengkapnya, lihat [Jenis Elastic Load Balancing](#).
- Load balancer dan grup targetnya harus sama Akun AWS, VPC, dan Region sebagai grup Auto Scaling Anda.
- Kelompok target harus menentukan jenis target instance. Anda tidak dapat menentukan jenis target ip saat menggunakan grup Auto Scaling.
- Jika template peluncuran untuk grup Auto Scaling Anda tidak berisi grup keamanan yang benar untuk mengizinkan lalu lintas masuk yang diperlukan dari penyeimbang beban, Anda harus memperbarui template peluncuran. Aturan yang direkomendasikan bergantung pada jenis load balancer dan jenis backend yang digunakan load balancer. Misalnya, untuk merutekan lalu lintas ke server web, izinkan akses HTTP masuk pada port 80 dari penyeimbang beban. Instance yang ada tidak diperbarui dengan pengaturan baru saat template peluncuran dimodifikasi. Untuk memperbarui instans yang ada, Anda dapat memulai penyegaran instans untuk mengganti instance. Untuk informasi selengkapnya, lihat [Menggunakan penyegaran instans untuk memperbarui instance dalam grup Auto Scaling](#).

- Grup keamanan dalam template peluncuran juga harus mengizinkan akses dari penyeimbang beban pada port yang benar untuk Elastic Load Balancing untuk melakukan pemeriksaan kesehatannya.
- Saat menerapkan peralatan virtual di belakang Load Balancer Gateway, Amazon Machine Image (AMI) di template peluncuran harus menentukan ID AMI yang mendukung protokol GENEVE agar grup Auto Scaling dapat bertukar lalu lintas dengan Load Balancer Gateway. Selain itu, grup keamanan dalam template peluncuran harus mengizinkan lalu lintas UDP di port 6081.

Tip

Jika Anda memiliki skrip bootstrap yang membutuhkan waktu cukup lama untuk diselesaikan, Anda dapat menambahkan hook siklus hidup peluncuran ke grup Auto Scaling Anda untuk menunda instance mendaftar di belakang penyeimbang beban sebelum skrip bootstrap Anda berhasil diselesaikan dan aplikasi pada instance siap menerima lalu lintas. Anda tidak dapat menambahkan kait siklus hidup saat pertama kali membuat grup Auto Scaling di konsol Auto Scaling Amazon EC2. Namun, Anda dapat menambahkan kait siklus hidup setelah grup dibuat. Untuk informasi selengkapnya, lihat [Hook siklus aktif Amazon EC2 Auto Scaling](#).

Konfigurasi pemeriksaan kesehatan untuk target

Anda dapat mengonfigurasi pemeriksaan kesehatan untuk target yang terdaftar dengan penyeimbang beban Elastic Load Balancing untuk memastikan mereka dapat menangani lalu lintas dengan benar. Langkah-langkah spesifik bervariasi berdasarkan jenis penyeimbang beban yang Anda gunakan. Untuk informasi selengkapnya, lihat sumber daya berikut:

- Application Load Balancer — Lihat [pemeriksaan Kesehatan untuk kelompok target Anda](#) di Panduan Pengguna untuk Application Load Balancers.
- Network Load Balancer — Lihat [pemeriksaan Kesehatan untuk grup target Anda](#) di Panduan Pengguna untuk Network Load Balancer.
- Load Balancer Gateway — Lihat [pemeriksaan Kesehatan untuk grup target Anda](#) di Panduan Pengguna untuk Penyeimbang Beban Gateway.
- Classic Load Balancer — Lihat [Mengonfigurasi pemeriksaan kesehatan untuk Classic Load Balancer Anda di Panduan Pengguna untuk Penyeimbang Beban Klasik](#).

Secara default, Auto Scaling Amazon EC2 tidak menganggap instans tidak sehat dan menggantinya jika gagal dalam pemeriksaan kesehatan Elastic Load Balancing. Pemeriksaan kesehatan default untuk grup Auto Scaling hanya pemeriksaan kesehatan EC2. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

Untuk mengaktifkan Auto Scaling Amazon EC2 menggantikan instans yang dilaporkan tidak sehat oleh Elastic Load Balancing, Anda dapat mengonfigurasi grup Auto Scaling untuk menggunakan pemeriksaan kesehatan Elastic Load Balancing. Dengan demikian, Amazon EC2 Auto Scaling menganggap instans tidak sehat jika gagal baik pemeriksaan kesehatan EC2 atau pemeriksaan kesehatan Elastic Load Balancing. Jika Anda melampirkan beberapa grup target penyeimbang beban atau Classic Load Balancer ke grup, semuanya harus melaporkan bahwa sebuah instance sehat agar instans tersebut dianggap sehat. Jika salah satu dari mereka melaporkan instance sebagai tidak sehat, grup Auto Scaling menggantikan instance, bahkan jika orang lain melaporkannya sebagai sehat.

Untuk informasi tentang cara mengaktifkan pemeriksaan kesehatan ini untuk grup Auto Scaling Anda, lihat [Pasang penyeimbang beban Elastic Load Balancing ke grup Auto Scaling Anda](#)

Note

Untuk memastikan bahwa pemeriksaan kesehatan ini dimulai sesegera mungkin, pastikan masa tenggang pemeriksaan kesehatan kelompok Anda tidak ditetapkan terlalu tinggi, tetapi cukup tinggi untuk pemeriksaan kesehatan Elastic Load Balancing Anda untuk menentukan apakah target tersedia untuk menangani permintaan. Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).

Pasang penyeimbang beban Elastic Load Balancing ke grup Auto Scaling Anda

Topik ini menjelaskan cara melampirkan penyeimbang beban Elastic Load Balancing ke grup Auto Scaling. Ini juga menjelaskan cara mengaktifkan pemeriksaan kesehatan Elastic Load Balancing agar Auto Scaling Amazon EC2 mengganti instans yang dilaporkan Elastic Load Balancing sebagai tidak sehat.

Secara default, Auto Scaling Amazon EC2 hanya menggantikan instans yang tidak sehat atau tidak dapat dijangkau berdasarkan pemeriksaan kesehatan Amazon EC2. Jika Anda mengaktifkan pemeriksaan kesehatan Elastic Load Balancing, Auto Scaling Amazon EC2 dapat menggantikan

instans yang sedang berjalan jika salah satu penyeimbang beban Elastic Load Balancing yang Anda lampirkan ke grup Auto Scaling melaporkannya sebagai tidak sehat.

Untuk tutorial tentang melampirkan Application Load Balancer ke grup Auto Scaling Anda, lihat.

[Tutorial: Siapkan aplikasi yang diskalakan dan diseimbangkan beban](#)

 Important

Sebelum Anda melanjutkan, lengkapi semua [prasyarat di bagian](#) sebelumnya.

Daftar Isi

- [Melampirkan grup target atau Classic Load Balancer](#)
- [Lepaskan grup target atau Classic Load Balancer](#)

Melampirkan grup target atau Classic Load Balancer

Saat membuat atau memperbarui grup Auto Scaling, Anda dapat melampirkan satu atau beberapa grup target atau Classic Load Balancer. Ketika Anda melampirkan Application Load Balancer, Network Load Balancer, atau Gateway Load Balancer, Anda melampirkan grup target daripada load balancer itu sendiri.

Ikuti langkah-langkah di bagian ini untuk menggunakan konsol untuk:

- Melampirkan grup target atau Classic Load Balancer ke grup Auto Scaling
- Nyalakan pemeriksaan kesehatan untuk Elastic Load Balancing

Untuk melampirkan penyeimbang beban yang ada saat Anda membuat grup Auto Scaling baru

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pada bilah navigasi di bagian atas layar, pilih tempat Wilayah AWS Anda membuat penyeimbang beban.
3. Pilih Buat grup Auto Scaling.
4. Pada langkah 1 dan 2, pilih opsi yang diinginkan dan lanjutkan ke Langkah 3: Konfigurasi opsi lanjutan.
5. Untuk Load balancing, pilih Lampirkan ke penyeimbang beban yang ada.

6. Di bawah Lampirkan ke penyeimbang beban yang ada, lakukan salah satu hal berikut:
 - a. Untuk Aplikasi Load Balancer, Network Load Balancers, dan Gateway Load Balancer:

Pilih Pilih dari grup target penyeimbang beban Anda, lalu pilih grup target di bidang Kelompok target penyeimbang beban yang ada.
 - b. Untuk Penyeimbang Beban Klasik:

Pilih Pilih dari Classic Load Balancers, lalu pilih load balancer Anda di bidang Classic Load Balancers.
7. (Opsional) Untuk pemeriksaan Kesehatan, Jenis pemeriksaan kesehatan tambahan, pilih Aktifkan pemeriksaan kesehatan Elastic Load Balancing.
8. (Opsional) Untuk Health check masa tenggang, masukkan jumlah waktu, dalam hitungan detik. Ini adalah berapa lama Amazon EC2 Auto Scaling perlu menunggu sebelum memeriksa status kesehatan suatu instans setelah memasuki status. InService Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).
9. Lanjutkan untuk membuat grup Auto Scaling. Instans Anda akan secara otomatis terdaftar ke penyeimbang beban setelah grup Auto Scaling dibuat.

Untuk melampirkan penyeimbang beban yang ada ke grup Auto Scaling Anda setelah dibuat

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.
3. Pada tab Detail, pilih Load balancing, Edit.
4. Di Bawah Load balancing, lakukan salah satu hal berikut:
 - a. Untuk grup target Application, Network atau Gateway Load Balancer, pilih kotak centang dan pilih grup target.
 - b. Untuk Classic Load Balancers, pilih kotak centang dan pilih penyeimbang beban Anda.
5. Pilih Perbarui.

Ketika Anda selesai memasang penyeimbang beban, Anda dapat mengaktifkan pemeriksaan kesehatan yang menggunakannya secara opsional.

Untuk mengaktifkan pemeriksaan kesehatan Elastic Load Balancing

1. Pada tab Detail, pilih Pemeriksaan kesehatan, Edit.
2. Untuk pemeriksaan Kesehatan, jenis pemeriksaan kesehatan tambahan, pilih Aktifkan pemeriksaan kesehatan Elastic Load Balancing.
3. Untuk Health check masa tenggang, masukkan jumlah waktu, dalam hitungan detik. Ini adalah berapa lama Amazon EC2 Auto Scaling perlu menunggu sebelum memeriksa status kesehatan suatu instans setelah memasuki status. InService Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).
4. Pilih Pembaruan.

Note

Anda dapat memantau status penyeimbang beban saat sedang dilampirkan dengan menggunakan AWS CLI. Jika Auto Scaling Amazon EC2 berhasil mendaftarkan instans dan setidaknya satu instans terdaftar lolos pemeriksaan kesehatan, Anda akan menerima status. InService Untuk informasi selengkapnya, lihat [Verifikasi status lampiran penyeimbang beban Anda](#).

Lepaskan grup target atau Classic Load Balancer

Saat Anda tidak lagi memerlukan load balancer, gunakan prosedur berikut untuk melepaskannya dari grup Auto Scaling Anda.

Untuk memisahkan load balancer dari grup

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup yang sudah ada.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Pada tab Detail, pilih Load balancing, Edit.
4. Di Bawah Load balancing, lakukan salah satu hal berikut:
 - a. Untuk grup target Application, Network atau Gateway Load Balancer, pilih ikon delete (X) di sebelah grup target.

- b. Untuk Classic Load Balancers, pilih ikon delete (X) di sebelah load balancer.
5. Pilih Perbarui.

Setelah Anda selesai melepaskan kelompok target, Anda dapat mematikan pemeriksaan kesehatan Elastic Load Balancing.

Untuk mematikan pemeriksaan kesehatan Elastic Load Balancing

1. Pada tab Detail, pilih Pemeriksaan kesehatan, Edit.
2. Untuk pemeriksaan Kesehatan, jenis pemeriksaan kesehatan tambahan, batalkan pilihan Nyalakan pemeriksaan kesehatan Elastic Load Balancing.
3. Pilih Perbarui.

Konfigurasi Application Load Balancer atau Network Load Balancer dari konsol Auto Scaling Amazon EC2

Gunakan prosedur berikut untuk membuat dan melampirkan Application Load Balancer atau Network Load Balancer saat Anda membuat grup Auto Scaling Anda.

Untuk membuat dan melampirkan penyeimbang beban baru saat Anda membuat grup Auto Scaling baru

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih Buat grup Auto Scaling.
3. Pada langkah 1 dan 2, pilih opsi yang diinginkan dan lanjutkan ke Langkah 3: Konfigurasi opsi lanjutan.
4. Untuk Load balancing, pilih Lampirkan ke penyeimbang beban baru.
 - a. Di bawah Lampirkan ke penyeimbang beban baru, untuk jenis Load balancer, pilih apakah akan membuat Application Load Balancer atau Network Load Balancer.
 - b. Untuk nama Load balancer, masukkan nama untuk penyeimbang beban, atau simpan nama default.
 - c. Untuk skema Load balancer, pilih apakah akan membuat penyeimbang beban yang menghadap ke internet publik, atau pertahankan default untuk penyeimbang beban internal.

- d. Untuk Availability Zones dan subnet, pilih subnet publik untuk setiap Availability Zone yang Anda pilih untuk meluncurkan instans EC2 Anda. (Ini sudah terisi dari langkah 2.).
 - e. Untuk Pendengar dan perutean, perbarui nomor port untuk pendengar Anda (jika perlu), dan di bawah Perutean default, pilih Buat grup target. Atau, Anda dapat memilih grup target yang ada dari daftar drop-down.
 - f. Jika Anda memilih Buat grup target di langkah terakhir, untuk Nama grup target baru, masukkan nama untuk grup target, atau simpan nama default.
 - g. Untuk menambahkan tag ke penyeimbang beban Anda, pilih Tambahkan tag, dan berikan kunci tag dan nilai untuk setiap tag.
5. (Opsional) Untuk pemeriksaan Kesehatan, Jenis pemeriksaan kesehatan tambahan, pilih Aktifkan pemeriksaan kesehatan Elastic Load Balancing.
 6. (Opsional) Untuk Health check masa tenggang, masukkan jumlah waktu, dalam hitungan detik. Ini adalah berapa lama Amazon EC2 Auto Scaling perlu menunggu sebelum memeriksa status kesehatan suatu instans setelah memasuki status. InService Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).
 7. Lanjutkan untuk membuat grup Auto Scaling. Instans Anda akan secara otomatis terdaftar ke penyeimbang beban setelah grup Auto Scaling dibuat.

Note

Setelah membuat grup Auto Scaling, Anda dapat menggunakan konsol Elastic Load Balancing untuk membuat pendengar tambahan. Ini berguna jika Anda perlu membuat pendengar dengan protokol aman, seperti HTTPS, atau pendengar UDP. Anda dapat menambahkan lebih banyak pendengar ke penyeimbang beban yang ada, selama Anda menggunakan port yang berbeda.

Verifikasi status lampiran penyeimbang beban Anda

Setelah Anda melampirkan penyeimbang beban, ia memasuki Adding status saat mendaftarkan instance dalam grup. Ketika semua instance dalam grup terdaftar, ia memasuki Added negara bagian. Setelah setidaknya satu instance yang didaftarkan lolos pemeriksaan kesehatan, proses ini masuk ke status InService. Saat penyeimbang beban dalam InService status, Auto Scaling Amazon EC2 dapat menghentikan dan mengganti instans apa pun yang dilaporkan tidak sehat. Jika tidak ada instance terdaftar yang lolos pemeriksaan kesehatan (misalnya, karena kesalahan

konfigurasi pemeriksaan kesehatan), load balancer tidak masuk ke status `InService`. Amazon EC2 Auto Scaling tidak menghentikan dan menggantikan instance.

Saat Anda melepaskan load balancer, sistem masuk ke status `Removing` saat membatalkan pendaftaran instance dalam grup. Contoh tetap berjalan setelah mereka deregister. Secara default, pengurusan koneksi (penundaan deregistrasi) diaktifkan untuk Application Load Balancers, Network Load Balancers, dan Gateway Load Balancers. Jika pengurusan koneksi diaktifkan, Elastic Load Balancing menunggu permintaan dalam penerbangan selesai atau batas waktu maksimum berakhir (mana yang lebih dulu) sebelum membatalkan pendaftaran instans.

Anda dapat memverifikasi status lampiran dengan menggunakan AWS Command Line Interface (AWS CLI) atau AWS SDK. Anda tidak dapat memverifikasi status lampiran dari konsol.

Untuk menggunakan AWS CLI untuk memverifikasi status lampiran

[describe-traffic-sources](#) Perintah berikut mengembalikan status lampiran semua sumber lalu lintas untuk grup Auto Scaling yang ditentukan.

```
aws autoscaling describe-traffic-sources --auto-scaling-group-name my-asg
```

Contoh mengembalikan ARN dari kelompok target Elastic Load Balancing yang dilampirkan ke grup Auto Scaling, bersama dengan status lampiran grup target dalam elemen `State`

```
{
  "TrafficSources": [
    {
      "Identifier": "arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-targets/1234567890123456",
      "State": "InService",
      "Type": "elbv2"
    }
  ]
}
```

Menambahkan dan menghapus Availability Zones

Untuk memanfaatkan keamanan dan keandalan redundansi geografis, rentangkan grup Auto Scaling Anda di beberapa Availability Zone di Wilayah tempat Anda bekerja dan lampirkan penyeimbang beban untuk mendistribusikan lalu lintas masuk di seluruh Availability Zone tersebut.

Jika satu Availability Zone menjadi tidak sehat atau tidak tersedia, Auto Scaling Amazon EC2 meluncurkan instans baru di Availability Zone yang tidak terpengaruh. Ketika Availability Zone yang tidak sehat kembali ke keadaan sehat, Auto Scaling Amazon EC2 secara otomatis mendistribusikan ulang instans aplikasi secara merata di semua Availability Zone untuk grup Auto Scaling Anda. Amazon EC2 Auto Scaling melakukan ini dengan mencoba meluncurkan instance baru di Zona Ketersediaan dengan beberapa instance paling sedikit. Namun, jika upaya gagal, Amazon EC2 Auto Scaling berupaya untuk meluncurkan di Zona Ketersediaan lainnya hingga berhasil.

Elastic Load Balancing membuat simpul penyeimbang beban untuk setiap Availability Zone yang Anda aktifkan untuk load balancer. Jika Anda mengaktifkan penyeimbangan beban lintas zona untuk penyeimbang beban, setiap node penyeimbang beban mendistribusikan lalu lintas secara merata di seluruh instans terdaftar di semua Availability Zone yang diaktifkan. Jika penyeimbangan beban lintas zona dinonaktifkan, setiap simpul penyeimbang beban mendistribusikan permintaan secara merata di seluruh instans terdaftar di Availability Zone saja.

Anda harus menentukan setidaknya satu Availability Zone saat membuat grup Auto Scaling. Kemudian, Anda dapat memperluas ketersediaan aplikasi Anda dengan menambahkan Availability Zone ke grup Auto Scaling Anda dan mengaktifkan Availability Zone untuk penyeimbang beban Anda (jika penyeimbang beban mendukungnya).

Daftar Isi

- [Tambahkan Availability Zone](#)
- [Menghapus Availability Zone](#)
- [Sumber daya terkait](#)
- [Batasan](#)

Tambahkan Availability Zone

Gunakan prosedur berikut untuk memperluas grup Auto Scaling dan load balancer ke subnet di Zona Ketersediaan tambahan.

Untuk menambahkan Availability Zone

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup yang sudah ada.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Pada tab Detail, pilih Jaringan, Edit.
4. Di Subnet, pilih subnet yang sesuai dengan Availability Zone yang ingin Anda tambahkan ke grup Auto Scaling.
5. Pilih Perbarui.
6. Untuk memperbarui Availability Zone untuk penyeimbang beban Anda sehingga berbagi Availability Zone yang sama dengan grup Auto Scaling Anda, selesaikan langkah-langkah berikut:
 - a. Pada panel navigasi, di bawah PENYEIMBANGAN BEBAN, pilih Penyeimbang beban.
 - b. Pilih load balancer Anda.
 - c. Lakukan salah satu hal berikut:
 - Untuk Penyeimbang Beban Aplikasi dan Penyeimbang Beban Jaringan:
 1. Pada tab Deskripsi, untuk Availability Zones, pilih Edit subnet.
 2. Pada halaman Edit subnet, untuk Availability Zones, pilih kotak centang untuk Availability Zone yang akan ditambahkan. Jika ada satu subnet untuk zona tersebut, subnet tersebut dipilih. Jika ada lebih dari satu subnet untuk zona tersebut, pilih salah satu subnet.
 - Untuk Classic Load Balancer dalam VPC:
 1. Pada tab Instans, pilih Edit Availability Zones.
 2. Pada halaman Tambah dan Hapus Subnet, untuk subnet yang tersedia, pilih subnet menggunakan ikon add (+). Subnet dipindahkan di bawah Subnet yang dipilih.
 - d. Pilih Simpan.

Menghapus Availability Zone

Untuk menghapus Availability Zone dari grup Auto Scaling dan load balancer, gunakan prosedur berikut.

Untuk menghapus Availability Zone

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup yang sudah ada.

Panel terpisah terbuka di bagian bawah halaman grup Auto Scaling.

3. Pada tab Detail, pilih Jaringan, Edit.
4. Di Subnet, pilih ikon hapus (X) untuk subnet yang sesuai dengan Availability Zone yang ingin Anda hapus dari grup Auto Scaling. Jika ada lebih dari satu subnet untuk zona itu, pilih ikon hapus (X) untuk masing-masing subnet.
5. Pilih Perbarui.
6. Untuk memperbarui Availability Zone untuk penyeimbang beban Anda sehingga berbagi Availability Zone yang sama dengan grup Auto Scaling Anda, selesaikan langkah-langkah berikut:
 - a. Pada panel navigasi, di bawah PENYEIMBANGAN BEBAN, pilih Penyeimbang beban.
 - b. Pilih load balancer Anda.
 - c. Lakukan salah satu hal berikut:
 - Untuk Penyeimbang Beban Aplikasi dan Penyeimbang Beban Jaringan:
 1. Pada tab Deskripsi, untuk Availability Zones, pilih Edit subnet.
 2. Pada halaman Edit subnet, untuk Availability Zones, kosongkan kotak centang untuk menghapus subnet untuk Availability Zone tersebut.
 - Untuk Classic Load Balancer dalam VPC:
 1. Pada tab Instans, pilih Edit Availability Zones.
 2. Pada halaman Tambah dan Hapus Subnet, untuk subnet yang tersedia, hapus subnet menggunakan ikon delete (-). Subnet dipindahkan di bawah subnet yang tersedia.
 - d. Pilih Simpan.

Sumber daya terkait

Auto Scaling Amazon EC2 menyeimbangkan kembali grup Anda saat Anda mengubah Availability Zone. Ini berarti mengganti dan mendistribusikan kembali beberapa contoh. Untuk informasi selengkapnya, lihat [Contoh: Mendistribusikan instance di seluruh Availability Zone](#).

Jika Anda telah mendaftarkan target di Availability Zone yang tidak diaktifkan untuk penyeimbang beban, penyeimbang beban tidak merutekan lalu lintas ke target tersebut. Untuk informasi selengkapnya, lihat [Cara kerja Elastic Load Balancing](#) di Panduan Pengguna Elastic Load Balancing.

Batasan

Untuk memperbarui Availability Zone yang diaktifkan untuk penyeimbang beban Anda, Anda perlu mengetahui batasan berikut:

- Saat Anda mengaktifkan Zona Ketersediaan untuk load balancer, Anda menentukan satu subnet dari Zona Ketersediaan tersebut. Perhatikan bahwa Anda dapat mengaktifkan paling banyak satu subnet per Availability Zone untuk penyeimbang beban Anda.
- Untuk penyeimbang beban yang menghadap ke internet, subnet yang Anda tentukan untuk penyeimbang beban harus memiliki setidaknya delapan alamat IP yang tersedia.
- Untuk Application Load Balancers, Anda harus mengaktifkan setidaknya dua Availability Zone.
- Untuk Network Load Balancers, Anda tidak dapat menonaktifkan Availability Zone yang diaktifkan, tetapi Anda dapat mengaktifkan yang tambahan.
- Untuk Gateway Load Balancers, Anda tidak dapat menonaktifkan Availability Zone yang diaktifkan, tetapi Anda dapat mengaktifkan yang tambahan.

Contoh untuk bekerja dengan Elastic Load Balancing dengan () AWS Command Line InterfaceAWS CLI

Gunakan AWS CLI untuk melampirkan, melepaskan, dan menjelaskan penyeimbang beban dan grup target, menambah dan menghapus pemeriksaan kesehatan Elastic Load Balancing, dan ubah Availability Zone mana yang diaktifkan.

Topik ini menunjukkan contoh AWS CLI perintah yang melakukan tugas umum untuk Amazon EC2 Auto Scaling.

Important

Untuk contoh perintah tambahan, lihat [aws elbv2](#) dan [aws elb](#) di AWS CLI Command Reference.

Daftar Isi

- [Lampirkan grup target Anda atau Classic Load Balancer](#)
- [Jelaskan grup target Anda atau Classic Load Balancer](#)

- [Tambahkan pemeriksaan kesehatan Elastic Load Balancing](#)
- [Ubah Availability Zone](#)
- [Lepaskan grup target atau Classic Load Balancer](#)
- [Hapus pemeriksaan kesehatan Elastic Load Balancing](#)
- [Perintah warisan](#)

Lampirkan grup target Anda atau Classic Load Balancer

Gunakan [create-auto-scaling-group](#) perintah berikut untuk membuat grup Auto Scaling dan secara bersamaan melampirkan grup target dengan menentukan Nama Sumber Daya Amazon (ARN). Grup target dapat dikaitkan dengan Application Load Balancer, Network Load Balancer, atau Gateway Load Balancer.

Ganti nilai sampel untuk `--auto-scaling-group-name`, `--vpc-zone-identifier`, `--min-size`, dan `--max-size`. Untuk `--launch-template` opsi, ganti *my-launch-template* dan *1* dengan nama dan versi template peluncuran untuk grup Auto Scaling Anda. Untuk `--traffic-sources` opsi, ganti sampel ARN dengan ARN dari grup target untuk Application Load Balancer, Network Load Balancer, atau Gateway Load Balancer.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateName=my-launch-template,Version='1' \  
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \  
  --min-size 1 --max-size 5 \  
  --traffic-sources "Identifier=arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-targets/12345678EXAMPLE1"
```

Gunakan [attach-traffic-sources](#) perintah untuk melampirkan grup target tambahan ke grup Auto Scaling setelah dibuat.

Perintah berikut menambahkan grup target lain ke grup yang sama.

```
aws autoscaling attach-traffic-sources --auto-scaling-group-name my-asg \  
  --traffic-sources "Identifier=arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-targets/12345678EXAMPLE2"
```

Atau, untuk melampirkan Classic Load Balancer ke grup Anda, tentukan `--traffic-sources` dan `--type` opsi saat Anda menggunakan `create-auto-scaling-group` atau `attach-traffic-sources`, seperti

pada contoh berikut. Ganti *my-classic-load-balancer* dengan nama Classic Load Balancer. Untuk `--type` opsi, tentukan nilai **elb**.

```
--traffic-sources "Identifier=my-classic-load-balancer" --type elb
```

Jelaskan grup target Anda atau Classic Load Balancer

Untuk menjelaskan penyeimbang beban atau grup target yang dilampirkan ke grup Auto Scaling Anda, gunakan perintah berikut. [describe-traffic-sources](#) Ganti *my-asg* dengan nama grup Anda.

```
aws autoscaling describe-traffic-sources --auto-scaling-group-name my-asg
```

Contoh mengembalikan ARN dari grup target Elastic Load Balancing yang Anda lampirkan ke grup Auto Scaling.

```
{
  "TrafficSources": [
    {
      "Identifier": "arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-targets/12345678EXAMPLE1",
      "State": "InService",
      "Type": "elbv2"
    },
    {
      "Identifier": "arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-targets/12345678EXAMPLE2",
      "State": "InService",
      "Type": "elbv2"
    }
  ]
}
```

Untuk penjelasan State bidang dalam output, lihat [Verifikasi status lampiran penyeimbang beban Anda](#).

Tambahkan pemeriksaan kesehatan Elastic Load Balancing

Untuk menambahkan pemeriksaan kesehatan Elastic Load Balancing ke pemeriksaan kesehatan yang dilakukan grup Auto Scaling Anda pada instance, gunakan perintah [update-auto-scaling-group](#) berikut dan **ELB** tentukan sebagai nilai untuk opsi tersebut. `--health-check-type` Ganti *my-asg* dengan nama grup Anda.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
--health-check-type "ELB"
```

Contoh baru sering membutuhkan waktu untuk pemanasan singkat sebelum mereka dapat lulus pemeriksaan kesehatan. Jika masa tenggang tidak memberikan waktu pemanasan yang cukup, contoh mungkin tidak tampak siap untuk melayani lalu lintas. Auto Scaling Amazon EC2 mungkin menganggap instans tersebut tidak sehat dan menggantikannya.

Untuk memperbarui masa tenggang pemeriksaan kesehatan, gunakan `--health-check-grace-period` opsi saat Anda menggunakan `update-auto-scaling-group`, seperti pada contoh berikut. Ganti `300` dengan jumlah detik untuk menyimpan instance baru dalam layanan sebelum menghentikannya jika ternyata tidak sehat.

```
--health-check-grace-period 300
```

Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

Ubah Availability Zone

Mengubah Availability Zone Anda memiliki beberapa batasan yang harus Anda waspadai. Untuk informasi selengkapnya, lihat [Batasan](#).

Untuk mengubah Availability Zone untuk Application Load Balancer atau Network Load Balancer

1. Sebelum Anda mengubah Availability Zones dari load balancer, sebaiknya Anda memperbarui Availability Zone dari grup Auto Scaling terlebih dahulu untuk memverifikasi bahwa ada ketersediaan untuk jenis instans Anda di zona yang ditentukan.

Untuk memperbarui Availability Zones untuk grup Auto Scaling Anda, gunakan perintah berikut [update-auto-scaling-group](#). Ganti ID subnet sampel dengan ID subnet di Availability Zones untuk mengaktifkan. Subnet yang ditentukan menggantikan subnet yang diaktifkan sebelumnya. Ganti `my-asg` dengan nama grup Anda.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
--vpc-zone-identifier "subnet-41767929, subnet-cb663da2, subnet-8360a9e7"
```

2. Gunakan [describe-auto-scaling-groups](#) perintah berikut untuk memverifikasi bahwa instance di subnet baru telah diluncurkan. Jika instans telah diluncurkan, Anda melihat daftar instance dan statusnya. Ganti `my-asg` dengan nama grup Anda.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

- Gunakan perintah [set-subnet](#) berikut untuk menentukan subnet untuk penyeimbang beban Anda. Ganti ID subnet sampel dengan ID subnet di Availability Zones untuk mengaktifkan. Anda hanya dapat menentukan satu subnet per Availability Zone. Subnet yang ditentukan menggantikan subnet yang diaktifkan sebelumnya. Ganti *my-lb-arn* dengan ARN penyeimbang beban Anda.

```
aws elbv2 set-subnets --load-balancer-arn my-lb-arn \  
--subnets subnet-41767929 subnet-cb663da2 subnet-8360a9e7
```

Untuk mengubah Availability Zone untuk Classic Load Balancer

- Sebelum Anda mengubah Availability Zones dari load balancer, sebaiknya Anda memperbarui Availability Zone dari grup Auto Scaling terlebih dahulu untuk memverifikasi bahwa ada ketersediaan untuk jenis instans Anda di zona yang ditentukan.

Untuk memperbarui Availability Zones untuk grup Auto Scaling Anda, gunakan perintah berikut [update-auto-scaling-group](#). Ganti ID subnet sampel dengan ID subnet di Availability Zones untuk mengaktifkan. Subnet yang ditentukan menggantikan subnet yang diaktifkan sebelumnya. Ganti *my-asg* dengan nama grup Anda.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
--vpc-zone-identifier "subnet-41767929,subnet-cb663da2"
```

- Gunakan [describe-auto-scaling-groups](#) perintah berikut untuk memverifikasi bahwa instance di subnet baru telah diluncurkan. Jika instans telah diluncurkan, Anda melihat daftar instance dan statusnya. Ganti *my-asg* dengan nama grup Anda.

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name my-asg
```

- Gunakan perintah [attach-load-balancer-to-subnets](#) berikut untuk mengaktifkan Availability Zone baru untuk Classic Load Balancer Anda. Ganti ID subnet sampel dengan ID subnet untuk Availability Zone untuk mengaktifkan. Ganti *my-lb* dengan nama penyeimbang beban Anda.

```
aws elb attach-load-balancer-to-subnets --load-balancer-name my-lb \  
--subnets subnet-cb663da2
```

Untuk menonaktifkan Availability Zone, gunakan perintah [detach-load-balancer-from-subnets](#) berikut. Ganti contoh subnet ID dengan ID subnet untuk Availability Zone untuk menonaktifkan. Ganti *my-lb* dengan nama penyeimbang beban Anda.

```
aws elb detach-load-balancer-from-subnets --load-balancer-name my-lb \  
--subnets subnet-8360a9e7
```

Lepaskan grup target atau Classic Load Balancer

[detach-traffic-sources](#) Perintah berikut melepaskan grup target dari grup Auto Scaling Anda saat Anda tidak lagi membutuhkannya.

Untuk `--auto-scaling-group-name` opsi, ganti *my-asg* dengan nama grup Anda. Untuk `--traffic-sources` opsi, ganti sampel ARN dengan ARN dari grup target untuk Application Load Balancer, Network Load Balancer, atau Gateway Load Balancer.

```
aws autoscaling detach-traffic-sources --auto-scaling-group-name my-asg \  
--traffic-sources "Identifier=arn:aws:elasticloadbalancing:region:account-  
id:targetgroup/my-targets/1234567890123456"
```

Untuk melepaskan Classic Load Balancer dari grup Anda, tentukan `--traffic-sources` opsi `--type` dan, seperti pada contoh berikut. Ganti *my-classic-load-balancer* dengan nama Classic Load Balancer. Untuk `--type` opsi, tentukan nilai **elb**.

```
--traffic-sources "Identifier=my-classic-load-balancer" --type elb
```

Hapus pemeriksaan kesehatan Elastic Load Balancing

Untuk menghapus pemeriksaan kesehatan Elastic Load Balancing dari grup Auto Scaling Anda, gunakan perintah [update-auto-scaling-group](#) berikut dan **EC2** tentukan sebagai nilai untuk opsi tersebut. `--health-check-type` Ganti *my-asg* dengan nama grup Anda.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
--health-check-type "EC2"
```

Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

Perintah warisan

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan perintah CLI lama untuk melampirkan, melepaskan, dan mendeskripsikan penyeimbang beban dan grup target. Mereka tetap dalam dokumen ini sebagai referensi untuk setiap pelanggan yang ingin menggunakannya. Kami terus mendukung perintah CLI lama, tetapi kami menyarankan Anda menggunakan perintah CLI “sumber lalu lintas” baru, yang dapat melampirkan dan melepaskan beberapa jenis sumber lalu lintas. Anda dapat menggunakan perintah CLI lama dan perintah CLI “sumber lalu lintas” pada grup Auto Scaling yang sama.

Lampirkan grup target Anda atau Classic Load Balancer (warisan)

Untuk melampirkan grup target Anda

[create-auto-scaling-group](#) Perintah berikut membuat grup Auto Scaling dengan grup target terlampir. Tentukan Nama Sumber Daya Amazon (ARN) dari grup target untuk Application Load Balancer, Network Load Balancer, atau Gateway Load Balancer.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateName=my-launch-template,Version='1' \  
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \  
  --target-group-arns "arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-  
targets/1234567890123456" \  
  --min-size 1 --max-size 5
```

Perintah [attach-load-balancer-target-groups](#) berikut melampirkan grup target ke grup Auto Scaling yang ada.

```
aws autoscaling attach-load-balancer-target-groups --auto-scaling-group-name my-asg \  
  --target-group-arns "arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-  
targets/1234567890123456"
```

Untuk melampirkan Classic Load Balancer

[create-auto-scaling-group](#) Perintah berikut membuat grup Auto Scaling dengan Classic Load Balancer yang terpasang.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \  
  --launch-configuration-name my-launch-config \  
  --target-group-arns "arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-  
targets/1234567890123456"
```

```
--vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \  
--load-balancer-names "my-load-balancer" \  
--min-size 1 --max-size 5
```

[attach-load-balancers](#) Perintah berikut melampirkan Classic Load Balancer yang ditentukan ke grup Auto Scaling yang ada.

```
aws autoscaling attach-load-balancers --auto-scaling-group-name my-asg \  
--load-balancer-names my-lb
```

Jelaskan grup target Anda atau Classic Load Balancer (warisan)

Untuk menggambarkan kelompok sasaran

Untuk mendeskripsikan grup target yang terkait dengan grup Auto Scaling, gunakan perintah [describe-load-balancer-target-groups](#). Contoh berikut mencantumkan kelompok target untuk *my-asg*.

```
aws autoscaling describe-load-balancer-target-groups --auto-scaling-group-name my-asg
```

Untuk menggambarkan Classic Load Balancers

Untuk mendeskripsikan Classic Load Balancer yang terkait dengan grup Auto Scaling, gunakan perintah tersebut. [describe-load-balancers](#) *Contoh berikut mencantumkan Classic Load Balancers untuk my-asg.*

```
aws autoscaling describe-load-balancers --auto-scaling-group-name my-asg
```

Lepaskan grup target Anda atau Classic Load Balancer (warisan)

Untuk melepaskan kelompok sasaran

Perintah [detach-load-balancer-target-groups](#) berikut melepaskan grup target dari grup Auto Scaling Anda saat Anda tidak lagi membutuhkannya.

```
aws autoscaling detach-load-balancer-target-groups --auto-scaling-group-name my-asg \  
--target-group-arns "arn:aws:elasticloadbalancing:region:account-id:targetgroup/my-  
targets/1234567890123456"
```

Untuk melepaskan Classic Load Balancer

[detach-load-balancers](#) Perintah berikut melepaskan Classic Load Balancer dari grup Auto Scaling Anda saat Anda tidak lagi membutuhkannya.

```
aws autoscaling detach-load-balancers --auto-scaling-group-name my-asg \  
--load-balancer-names my-lb
```

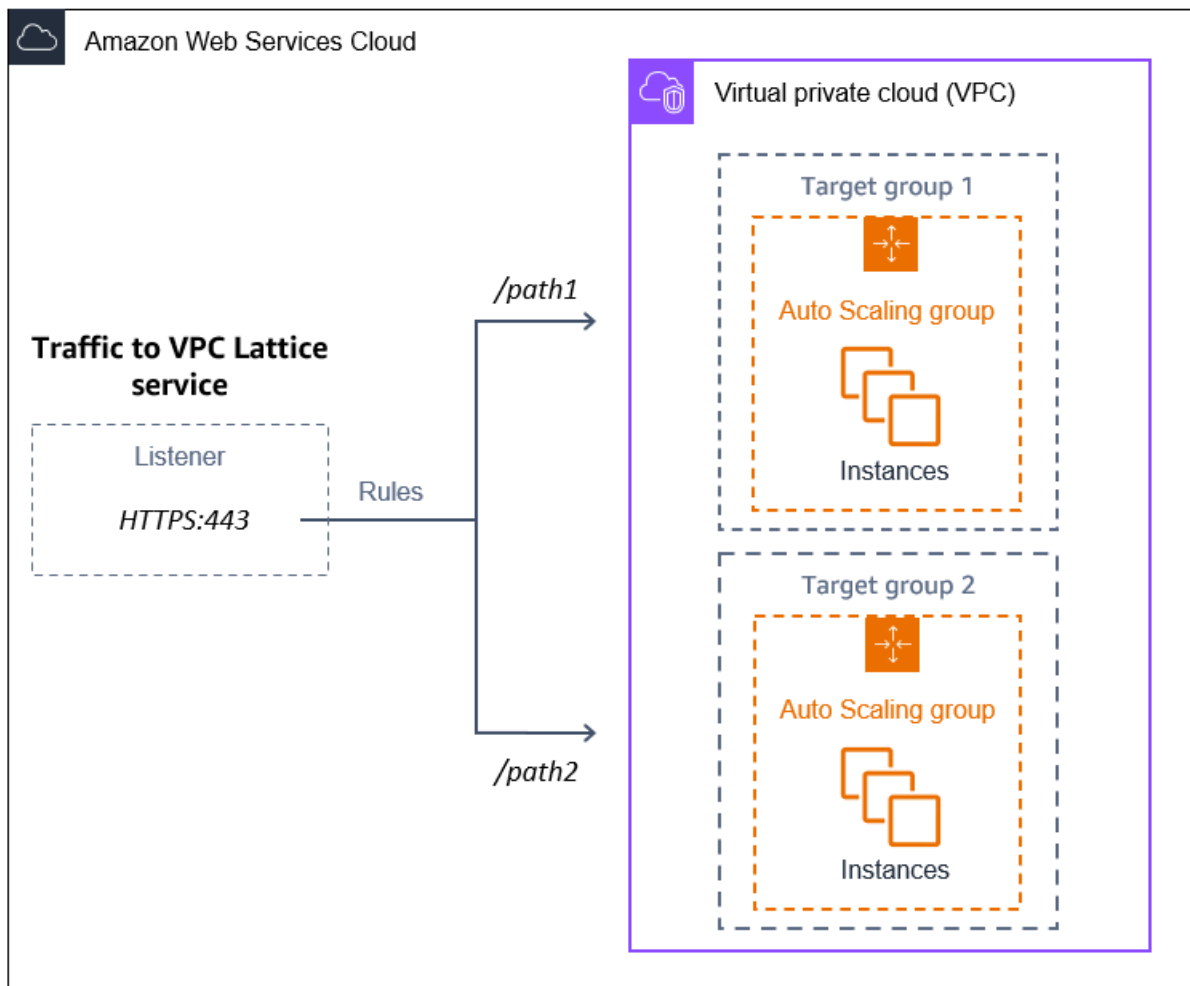
Rutekan lalu lintas ke grup Auto Scaling Anda dengan grup target VPC Lattice

Anda dapat menggunakan Amazon VPC Lattice untuk mengelola arus lalu lintas dan panggilan API antara aplikasi dan layanan yang berjalan pada sumber daya terpisah, seperti grup Auto Scaling atau fungsi Lambda. VPC Lattice adalah layanan jaringan aplikasi yang memungkinkan Anda terhubung, mengamankan, dan memantau semua layanan Anda di beberapa akun dan virtual private cloud (VPC). Untuk mempelajari selengkapnya tentang Kisi VPC, lihat [Apa itu Kisi VPC?](#)

Untuk memulai dengan VPC Lattice, pertama-tama buat sumber daya VPC Lattice yang diperlukan yang memungkinkan sumber daya dalam VPC yang terkait dengan jaringan layanan untuk terhubung satu sama lain. Sumber daya ini mencakup layanan, pendengar, aturan pendengar, dan grup target.

Untuk mengaitkan grup Auto Scaling ke layanan VPC Lattice, buat grup target untuk layanan yang merutekan permintaan ke instance yang didaftarkan berdasarkan ID instans, dan tambahkan listener ke layanan yang mengirimkan permintaan ke grup target. Kemudian, lampirkan grup target ke grup Auto Scaling Anda. Auto Scaling Amazon EC2 secara otomatis mendaftarkan instans EC2 sebagai target dengan grup target. Kemudian, saat Auto Scaling Amazon EC2 perlu menghentikan instans, instans otomatis membatalkan pendaftaran instans dari grup target sebelum penghentian.

Setelah Anda melampirkan grup target, itu adalah titik masuk untuk semua permintaan yang masuk ke grup Auto Scaling Anda. Seperti yang ditunjukkan contoh dalam diagram berikut, permintaan masuk kemudian dapat diarahkan ke grup target yang sesuai menggunakan aturan pendengar yang ditentukan untuk layanan VPC Lattice.



Saat lalu lintas dirutekan melalui VPC Lattice ke grup Auto Scaling Anda, VPC Lattice menyeimbangkan permintaan di antara instance dalam grup menggunakan penyeimbangan beban round robin. VPC Lattice juga dapat memantau kesehatan instans terdaftar dan mengarahkan lalu lintas hanya ke instans yang sehat.

Agar instans tetap tersedia untuk permintaan masuk, Anda dapat menambahkan pemeriksaan kesehatan VPC Lattice secara opsional ke grup Auto Scaling. Dengan cara ini, jika salah satu instans EC2 gagal, grup Auto Scaling Anda secara otomatis meluncurkan instance baru untuk menggantikannya. Perilaku pemeriksaan kesehatan VPC Lattice mirip dengan perilaku pemeriksaan kesehatan Elastic Load Balancing. Pemeriksaan kesehatan default untuk grup Auto Scaling hanya pemeriksaan kesehatan EC2.

Untuk mempelajari selengkapnya tentang Kisi VPC, lihat [Menyederhanakan Konektivitas, Keamanan, dan Pemantauan Layanan-ke-Layanan dengan Kisi VPC Amazon](#) — Sekarang Umumnya Tersedia di Blog. AWS

Daftar Isi

- [Bersiaplah untuk melampirkan grup target VPC Lattice ke grup Auto Scaling](#)
- [Lampirkan grup target VPC Lattice ke grup Auto Scaling](#)
- [Verifikasi status lampiran grup target VPC Lattice](#)

Bersiaplah untuk melampirkan grup target VPC Lattice ke grup Auto Scaling

Sebelum Anda melampirkan grup target VPC Lattice ke grup Auto Scaling Anda, Anda harus menyelesaikan prasyarat berikut:

- Anda harus sudah membuat jaringan layanan, layanan, pendengar, dan grup target VPC Lattice. Untuk informasi selengkapnya, lihat topik berikut di Panduan Pengguna Kisi VPC:
 - [Jaringan layanan](#)
 - [Jasa](#)
 - [Pendengar](#)
 - [Kelompok sasaran](#)
- Grup target harus sama Akun AWS, VPC, dan Wilayah sebagai grup Auto Scaling Anda.
- Kelompok target harus menentukan jenis `targetinstance`. Anda tidak dapat menentukan jenis target `ip` saat menggunakan grup Auto Scaling.
- Anda harus memiliki izin IAM yang cukup untuk melampirkan grup target ke grup Auto Scaling. Contoh kebijakan berikut menunjukkan izin minimum yang diperlukan yang diperlukan untuk melampirkan dan melepaskan grup target.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:AttachTrafficSources",
        "autoscaling:DetachTrafficSources",
        "autoscaling:DescribeTrafficSources",
        "vpc-lattice:RegisterTargets",
        "vpc-lattice:DeregisterTargets"
      ],
      "Resource": "*"
    }
  ]
}
```

```
]
}
```

- Jika template peluncuran untuk grup Auto Scaling Anda tidak berisi pengaturan yang benar untuk VPC Lattice, seperti grup keamanan yang kompatibel, Anda harus memperbarui template peluncuran. Instance yang ada tidak diperbarui dengan pengaturan baru saat template peluncuran dimodifikasi. Untuk memperbarui instance yang ada, Anda dapat memulai penyegaran instans untuk mengganti instance. Untuk informasi selengkapnya, lihat [Menggunakan penyegaran instans untuk memperbarui instance dalam grup Auto Scaling](#).
- Sebelum mengaktifkan pemeriksaan kesehatan VPC Lattice pada grup Auto Scaling, Anda dapat mengonfigurasi pemeriksaan kesehatan berbasis aplikasi untuk memverifikasi bahwa aplikasi Anda merespons seperti yang diharapkan. Untuk informasi selengkapnya, lihat [Pemeriksaan Kesehatan untuk grup target Anda](#) di Panduan Pengguna VPC Lattice.

Grup keamanan: Aturan masuk dan keluar

Grup keamanan bertindak sebagai firewall untuk instans EC2 terkait, mengendalikan lalu lintas masuk dan keluar pada tingkat instans.

Note

Konfigurasi jaringan cukup kompleks sehingga kami sangat menyarankan Anda membuat grup keamanan baru untuk digunakan dengan VPC Lattice. Ini juga memudahkan AWS Support untuk membantu Anda jika Anda perlu menghubungi mereka. Bagian berikut didasarkan pada asumsi bahwa Anda mengikuti rekomendasi ini.

Untuk mempelajari selengkapnya tentang membuat grup keamanan untuk Kisi VPC yang dapat Anda gunakan dengan grup Auto Scaling, lihat [Mengontrol lalu lintas menggunakan grup keamanan di Panduan Pengguna Kisi VPC](#). Untuk memecahkan masalah arus lalu lintas, lihat Panduan Pengguna VPC Lattice untuk informasi lebih lanjut.

Untuk informasi tentang cara membuat grup keamanan, lihat [Membuat grup keamanan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux dan gunakan tabel berikut untuk menentukan opsi apa yang harus dipilih.

Opsi	Nilai	
Nama	Sebuah nama yang mudah bagi Anda untuk diingat.	
Deskripsi	Deskripsi untuk membantu Anda mengidentifikasi grup keamanan.	
VPC	VPC yang sama dengan grup Auto Scaling.	

Aturan-aturan ke dalam

Saat Anda membuat grup keamanan, grup ini tidak memiliki aturan masuk. Tidak ada lalu lintas masuk yang berasal dari klien dalam jaringan layanan VPC Lattice ke instans Anda yang diizinkan hingga Anda menambahkan aturan masuk ke grup keamanan.

Agar klien dalam jaringan layanan VPC Lattice dapat terhubung ke instans di grup Auto Scaling Anda, grup keamanan untuk grup Auto Scaling Anda harus diatur dengan benar. Dalam hal ini, berikan aturan masuk untuk mengizinkan lalu lintas dari nama daftar awalan AWS terkelola untuk VPC Lattice, bukan alamat IP tertentu. Daftar awalan VPC Lattice adalah rentang alamat IP yang digunakan oleh VPC Lattice dalam notasi CIDR. Untuk informasi selengkapnya, lihat [Bekerja dengan daftar awalan AWS-terkelola](#) di Panduan Pengguna Amazon VPC.

Untuk informasi tentang cara menambahkan aturan ke grup keamanan, lihat [Menambahkan aturan ke grup keamanan Anda](#) di Panduan Pengguna Amazon VPC dan gunakan tabel berikut untuk menentukan opsi apa yang harus dipilih.

Opsi	Nilai	
Aturan HTTP	Jenis: HTTP Sumber: <code>com.amazonaws.wilayah .vpc-kisi</code>	
Aturan HTTPS	Jenis: HTTPS	

Opsi	Nilai	
	Sumber: com.amazonaws. <i>wilayah .vpc-kisi</i>	

Grup keamanan stateful: memungkinkan lalu lintas dari klien dalam jaringan layanan VPC Lattice ke instance di grup Auto Scaling Anda, dan kemudian mengirimkan respons kembali ke klien yang sebelumnya ditinggalkan.

Aturan-aturan ke luar

Secara default, suatu grup keamanan mencakup aturan keluar yang mengizinkan semua lalu lintas keluar. Anda dapat menghapus aturan default ini secara opsional dan menambahkan aturan keluar untuk mengakomodasi kebutuhan keamanan tertentu.

Batasan

- [Grup instance campuran](#) tidak didukung. Jika Anda mencoba melampirkan grup target Kisi VPC ke grup Auto Scaling yang memiliki kebijakan instance campuran, Anda menerima pesan galat. Saat ini, Grup Auto Scaling dengan instance campuran tidak dapat diintegrasikan dengan layanan Kisi VPC. Ini karena algoritma load balancing mendistribusikan beban secara merata ke semua sumber daya yang tersedia dan mengasumsikan bahwa instance cukup serupa untuk menangani beban yang sama.

Lampirkan grup target VPC Lattice ke grup Auto Scaling

Topik ini menjelaskan cara melampirkan grup target VPC Lattice ke grup Auto Scaling. Ini juga menjelaskan cara mengaktifkan pemeriksaan kesehatan VPC Lattice agar Amazon EC2 Auto Scaling mengganti instance yang dilaporkan VPC Lattice sebagai tidak sehat.

Secara default, Auto Scaling Amazon EC2 hanya menggantikan instans yang tidak sehat atau tidak dapat dijangkau berdasarkan pemeriksaan kesehatan Amazon EC2. Jika Anda mengaktifkan pemeriksaan kesehatan VPC Lattice, Auto Scaling Amazon EC2 dapat menggantikan instance yang sedang berjalan jika salah satu grup target VPC Lattice yang Anda lampirkan ke grup Auto Scaling melaporkannya sebagai tidak sehat. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

⚠ Important

Sebelum Anda melanjutkan, lengkapi semua [prasyarat di bagian](#) sebelumnya.

Lampirkan grup target VPC Lattice

Anda dapat melampirkan satu atau beberapa grup target ke grup Auto Scaling saat membuat atau memperbarui grup.

Console

Ikuti langkah-langkah di bagian ini untuk menggunakan konsol untuk:

- Lampirkan grup target VPC Lattice ke grup Auto Scaling
- Nyalakan pemeriksaan kesehatan untuk VPC Lattice

Untuk melampirkan grup target VPC Lattice ke grup Auto Scaling baru

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pada bilah navigasi di bagian atas layar, pilih tempat Wilayah AWS Anda membuat grup target.
3. Pilih Buat grup Auto Scaling.
4. Pada langkah 1 dan 2, pilih opsi yang Anda inginkan dan lanjutkan ke Langkah 3: Konfigurasi opsi lanjutan.
5. Untuk opsi integrasi VPC Lattice, pilih Lampirkan ke layanan VPC Lattice.
6. Di bawah Pilih grup target VPC Lattice, pilih grup target Anda.
7. (Opsional) Untuk pemeriksaan Kesehatan, Jenis pemeriksaan kesehatan tambahan, pilih Aktifkan pemeriksaan kesehatan VPC Lattice.
8. (Opsional) Untuk Health check masa tenggang, masukkan jumlah waktu, dalam hitungan detik. Ini adalah berapa lama Amazon EC2 Auto Scaling perlu menunggu sebelum memeriksa status kesehatan suatu instans setelah memasuki status. InService Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).

9. Lanjutkan untuk membuat grup Auto Scaling. Instance Anda akan secara otomatis terdaftar ke grup target VPC Lattice setelah grup Auto Scaling dibuat.

Untuk melampirkan grup target VPC Lattice ke grup Auto Scaling yang ada

Gunakan prosedur berikut untuk melampirkan grup target untuk layanan ke grup Auto Scaling yang ada.

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup Auto Scaling Anda.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Detail, pilih opsi integrasi VPC Lattice, Edit.
4. Di bawah opsi integrasi VPC Lattice, pilih Lampirkan ke layanan Kisi VPC.
5. Di bawah Pilih grup target VPC Lattice, pilih grup target Anda.
6. Pilih Perbarui.

Ketika Anda selesai melampirkan kelompok sasaran, Anda dapat mengaktifkan pemeriksaan kesehatan yang menggunakannya.

Untuk mengaktifkan pemeriksaan kesehatan VPC Lattice

1. Pada tab Detail, pilih Pemeriksaan kesehatan, Edit.
2. Untuk pemeriksaan Kesehatan, Jenis pemeriksaan kesehatan tambahan, pilih Aktifkan pemeriksaan kesehatan VPC Lattice.
3. Untuk Health check masa tenggang, masukkan jumlah waktu, dalam hitungan detik. Ini adalah berapa lama Amazon EC2 Auto Scaling perlu menunggu sebelum memeriksa status kesehatan suatu instans setelah memasuki status. InService Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).
4. Pilih Pembaruan.

AWS CLI

Ikuti langkah-langkah di bagian ini untuk menggunakan AWS CLI to:

- Lampirkan grup target VPC Lattice ke grup Auto Scaling
- Nyalakan pemeriksaan kesehatan untuk VPC Lattice

Untuk melampirkan grup target VPC Lattice ke grup Auto Scaling

Gunakan [create-auto-scaling-group](#) perintah berikut untuk membuat grup Auto Scaling dan secara bersamaan melampirkan grup target VPC Lattice dengan menentukan Nama Sumber Daya Amazon (ARN).

Ganti nilai sampel untuk `--auto-scaling-group-name`, `--vpc-zone-identifier`, `--min-size`, dan `--max-size`. Untuk `--launch-template` opsi, ganti *my-launch-template* dan *1* dengan nama dan versi template peluncuran yang Anda buat untuk instance yang terdaftar ke grup target VPC Lattice. Untuk `--traffic-sources` opsi, ganti sampel ARN dengan ARN dari grup target VPC Lattice Anda.

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg \  
  --launch-template LaunchTemplateName=my-launch-template,Version='1' \  
  --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782" \  
  --min-size 1 --max-size 5 \  
  --traffic-sources "Identifier=arn:aws:vpc-lattice:region:account-id:targetgroup/  
tg-0e2f2665eEXAMPLE"
```

Gunakan [attach-traffic-sources](#) perintah berikut untuk melampirkan grup target VPC Lattice ke grup Auto Scaling setelah dibuat.

```
aws autoscaling attach-traffic-sources --auto-scaling-group-name my-asg \  
  --traffic-sources "Identifier=arn:aws:vpc-lattice:region:account-id:targetgroup/  
tg-0e2f2665eEXAMPLE"
```

Untuk mengaktifkan pemeriksaan kesehatan untuk VPC Lattice

Jika Anda telah mengonfigurasi pemeriksaan kesehatan berbasis aplikasi untuk grup target VPC Lattice, Anda dapat mengaktifkan pemeriksaan kesehatan ini. Gunakan [update-auto-scaling-group](#) perintah [create-auto-scaling-group](#) dengan `--health-check-type` opsi dan nilai `VPC_LATTICE`. Untuk menentukan masa tenggang untuk pemeriksaan kesehatan yang dilakukan oleh grup Auto Scaling Anda, sertakan `--health-check-grace-period` opsi dan berikan nilainya dalam hitungan detik.

```
--health-check-type "VPC_LATTICE" --health-check-grace-period 60
```

Lepaskan grup target VPC Lattice

Jika Anda tidak perlu lagi menggunakan VPC Lattice, gunakan prosedur berikut untuk melepaskan grup target dari grup Auto Scaling Anda.

Console

Ikuti langkah-langkah di bagian ini untuk menggunakan konsol untuk:

- Lepaskan grup target VPC Lattice dari grup Auto Scaling
- Matikan pemeriksaan kesehatan untuk VPC Lattice

Untuk melepaskan grup target VPC Lattice dari grup Auto Scaling

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>, dan pilih Grup Auto Scaling dari panel navigasi.
2. Pilih kotak centang di samping grup yang sudah ada.

Panel split terbuka di bagian bawah halaman.

3. Pada tab Detail, pilih opsi integrasi VPC Lattice, Edit.
4. Di bawah opsi integrasi VPC Lattice, pilih ikon hapus (X) di sebelah grup target.
5. Pilih Perbarui.

Setelah Anda selesai melepaskan grup target, Anda dapat mematikan pemeriksaan kesehatan VPC Lattice.

Untuk mematikan pemeriksaan kesehatan VPC Lattice

1. Pada tab Detail, pilih Pemeriksaan kesehatan, Edit.
2. Untuk pemeriksaan Kesehatan, Jenis pemeriksaan kesehatan tambahan, batal pilih Aktifkan pemeriksaan kesehatan VPC Lattice.
3. Pilih Perbarui.

AWS CLI

Ikuti langkah-langkah di bagian ini untuk menggunakan AWS CLI to:

- Lepaskan grup target VPC Lattice dari grup Auto Scaling

- Matikan pemeriksaan kesehatan untuk VPC Lattice

Gunakan [detach-traffic-sources](#) perintah untuk melepaskan grup target dari grup Auto Scaling Anda saat Anda tidak lagi membutuhkannya.

```
aws autoscaling detach-traffic-sources --auto-scaling-group-name my-asg \  
  --traffic-sources "Identifier=arn:aws:vpc-lattice:region:account-id:targetgroup/  
tg-0e2f2665eEXAMPLE"
```

Untuk memperbarui pemeriksaan kesehatan pada grup Auto Scaling sehingga tidak lagi menggunakan pemeriksaan kesehatan VPC Lattice, gunakan perintah. [update-auto-scaling-group](#) Sertakan `--health-check-type` opsi dan nilai **EC2**.

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name my-asg \  
  --health-check-type "EC2"
```

Verifikasi status lampiran grup target VPC Lattice

Setelah Anda melampirkan grup target VPC Lattice ke grup Auto Scaling, grup tersebut memasuki Adding status saat mendaftarkan instance dalam grup. Ketika semua instance dalam grup terdaftar, ia memasuki Added negara bagian. Setelah setidaknya satu instance yang didaftarkan lolos pemeriksaan kesehatan, proses ini masuk ke status InService. Saat grup target dalam InService status, Auto Scaling Amazon EC2 dapat menghentikan dan mengganti instans apa pun yang dilaporkan tidak sehat. Jika tidak ada instance terdaftar yang lulus pemeriksaan kesehatan (misalnya, karena pemeriksaan kesehatan yang salah dikonfigurasi), grup target tidak memasuki InService status. Amazon EC2 Auto Scaling tidak menghentikan dan menggantikan instance.

Saat Anda melepaskan grup target untuk layanan, grup tersebut memasuki Removing status saat membatalkan pendaftaran instance dalam grup. Contoh tetap berjalan setelah mereka deregister. Secara default, pengurusan koneksi (penundaan deregistrasi) diaktifkan. Jika pengurusan koneksi diaktifkan, VPC Lattice menunggu permintaan dalam penerbangan selesai atau batas waktu maksimum berakhir (mana yang lebih dulu) sebelum membatalkan pendaftaran instans.

Anda dapat memverifikasi status lampiran dengan menggunakan AWS Command Line Interface (AWS CLI) atau AWS SDK. Anda tidak dapat memverifikasi status lampiran dari konsol.

Untuk menggunakan AWS CLI untuk memverifikasi status lampiran

[describe-traffic-sources](#) Perintah berikut mengembalikan status lampiran semua sumber lalu lintas untuk grup Auto Scaling yang ditentukan.

```
aws autoscaling describe-traffic-sources --auto-scaling-group-name my-asg
```

Contoh mengembalikan ARN dari grup target VPC Lattice yang dilampirkan ke grup Auto Scaling, bersama dengan status lampiran grup target dalam elemen. State

```
{
  "TrafficSources": [
    {
      "Identifier": "arn:aws:vpc-lattice:region:account-id:targetgroup/tg-0e2f2665eEXAMPLE",
      "State": "InService",
      "Type": "vpc-lattice"
    }
  ]
}
```

Gunakan EventBridge untuk menangani peristiwa Auto Scaling

Amazon EventBridge, sebelumnya disebut CloudWatch Events, membantu Anda mengatur aturan berbasis peristiwa yang memantau sumber daya dan memulai tindakan target yang menggunakan layanan lain. AWS

Acara dari Amazon EC2 Auto Scaling dikirimkan EventBridge ke dalam waktu dekat. Anda dapat menetapkan EventBridge aturan yang memanggil tindakan dan pemberitahuan terprogram sebagai tanggapan terhadap berbagai peristiwa ini. Misalnya, saat instance sedang dalam proses peluncuran atau penghentian, Anda dapat memanggil AWS Lambda fungsi untuk melakukan tugas yang telah dikonfigurasi sebelumnya.

Target EventBridge aturan dapat mencakup AWS Lambda fungsi, topik Amazon SNS, tujuan API, bus acara di lainnya Akun AWS, dan banyak lagi. Untuk informasi tentang target yang didukung, lihat [EventBridge Target Amazon](#) di Panduan EventBridge Pengguna Amazon.

Mulailah dengan membuat EventBridge aturan dengan contoh menggunakan topik Amazon SNS dan aturan. EventBridge Kemudian, ketika pengguna memulai penyegaran instans, Amazon SNS memberi tahu Anda melalui email setiap kali pos pemeriksaan tercapai. Untuk informasi selengkapnya, lihat [Buat EventBridge aturan untuk acara penyegaran misalnya](#).

Daftar Isi

- [Referensi acara Auto Scaling Amazon EC2](#)
- [Acara dan pola contoh kolam hangat](#)
- [Buat EventBridge aturan](#)

Referensi acara Auto Scaling Amazon EC2

Menggunakan Amazon EventBridge, Anda dapat membuat aturan yang cocok dengan peristiwa yang masuk dan merutekannya ke target untuk diproses.

Daftar Isi

- [Peristiwa tindakan siklus hidup](#)
- [Acara penskalaan yang sukses](#)
- [Peristiwa penskalaan yang gagal](#)
- [Acara penyegaran contoh](#)

Peristiwa tindakan siklus hidup

Saat menambahkan kait siklus hidup ke grup Auto Scaling, Auto Scaling Amazon EC2 akan mengirimkan peristiwa saat instance bertransisi EventBridge ke status tunggu. Acara diproduksi atas dasar upaya terbaik.

Jenis peristiwa

- [Tindakan siklus hidup penskalaan](#)
- [Tindakan siklus hidup skala dalam](#)

Tindakan siklus hidup penskalaan

Contoh peristiwa berikut menunjukkan bahwa Auto Scaling Amazon EC2 memindahkan instance ke status karena Pending:Wait hook siklus hidup peluncuran.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-launch Lifecycle Action",
  "source": "aws.autoscaling",
```

```

"account": "123456789012",
"time": "yyyy-mm-ddThh:mm:ssZ",
"region": "us-west-2",
"resources": [
  "auto-scaling-group-arn"
],
"detail": {
  "LifecycleActionToken": "87654321-4321-4321-4321-210987654321",
  "AutoScalingGroupName": "my-asg",
  "LifecycleHookName": "my-lifecycle-hook",
  "EC2InstanceId": "i-1234567890abcdef0",
  "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
  "NotificationMetadata": "additional-info",
  "Origin": "EC2",
  "Destination": "AutoScalingGroup"
}
}

```

Tindakan siklus hidup skala dalam

Contoh peristiwa berikut menunjukkan bahwa Auto Scaling Amazon EC2 memindahkan instance ke status karena hook `Terminating:Wait` siklus hidup penghentian.

Important

Saat grup Auto Scaling mengembalikan instance ke kolam hangat dalam skala, mengembalikan instance ke kolam hangat juga dapat menghasilkan peristiwa. EC2 Instance-terminate Lifecycle Action Peristiwa yang dikirimkan saat instance pindah ke status tunggu pada skala memiliki `WarmPool` nilai untuk `Destination`. Untuk informasi selengkapnya, lihat [Instance reuse policy](#).

```

{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-terminate Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [

```



```

    "auto-scaling-group-arn"
  ],
  "detail": {
    "LifecycleActionToken": "87654321-4321-4321-4321-210987654321",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING",
    "NotificationMetadata": "additional-info",
    "Origin": "AutoScalingGroup",
    "Destination": "EC2"
  }
}

```

Acara penskalaan yang sukses

Contoh berikut menunjukkan jenis acara untuk acara penskalaan yang berhasil. Acara diproduksi atas dasar upaya terbaik.

Jenis peristiwa

- [Acara scale-out yang sukses](#)
- [Acara scale-in yang sukses](#)

Acara scale-out yang sukses

Contoh peristiwa berikut menunjukkan bahwa Amazon EC2 Auto Scaling berhasil meluncurkan instans.

```

{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn",
    "instance-arn"
  ],
  "detail": {

```

```

"StatusCode": "InProgress",
"Description": "Launching a new EC2 instance: i-12345678",
"AutoScalingGroupName": "my-asg",
"ActivityId": "87654321-4321-4321-4321-210987654321",
"Details": {
  "Availability Zone": "us-west-2b",
  "Subnet ID": "subnet-12345678"
},
"RequestId": "12345678-1234-1234-1234-123456789012",
"StatusMessage": "",
"EndTime": "yyyy-mm-ddThh:mm:ssZ",
"EC2InstanceId": "i-1234567890abcdef0",
"StartTime": "yyyy-mm-ddThh:mm:ssZ",
"Cause": "description-text",
"Origin": "EC2",
"Destination": "AutoScalingGroup"
}
}

```

Acara scale-in yang sukses

Contoh peristiwa berikut menunjukkan bahwa Auto Scaling Amazon EC2 berhasil menghentikan instance.

Important

Saat grup Auto Scaling mengembalikan instance ke kolam hangat dalam skala, mengembalikan instance ke kolam hangat juga dapat menghasilkan peristiwa. EC2 Instance Terminate Successful Peristiwa yang dikirimkan ketika sebuah instance berhasil kembali ke kolam hangat memiliki WarmPool nilai untuk Destination. Untuk informasi selengkapnya, lihat [Instance reuse policy](#).

```

{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance Terminate Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [

```

```

    "auto-scaling-group-arn",
    "instance-arn"
  ],
  "detail": {
    "StatusCode": "InProgress",
    "Description": "Terminating EC2 instance: i-12345678",
    "AutoScalingGroupName": "my-asg",
    "ActivityId": "87654321-4321-4321-4321-210987654321",
    "Details": {
      "Availability Zone": "us-west-2b",
      "Subnet ID": "subnet-12345678"
    },
    "RequestId": "12345678-1234-1234-1234-123456789012",
    "StatusMessage": "",
    "EndTime": "yyyy-mm-ddThh:mm:ssZ",
    "EC2InstanceId": "i-1234567890abcdef0",
    "StartTime": "yyyy-mm-ddThh:mm:ssZ",
    "Cause": "description-text",
    "Origin": "AutoScalingGroup",
    "Destination": "EC2"
  }
}

```

Peristiwa penskalaan yang gagal

Contoh berikut menunjukkan jenis acara untuk peristiwa penskalaan yang gagal. Acara diproduksi atas dasar upaya terbaik.

Jenis peristiwa

- [Acara scale-out yang gagal](#)
- [Acara scale-in yang gagal](#)

Acara scale-out yang gagal

Contoh peristiwa berikut menunjukkan bahwa Auto Scaling Amazon EC2 gagal meluncurkan instance.

```

{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance Launch Unsuccessful",

```

```

"source": "aws.autoscaling",
"account": "123456789012",
"time": "yyyy-mm-ddThh:mm:ssZ",
"region": "us-west-2",
"resources": [
  "auto-scaling-group-arn",
  "instance-arn"
],
"detail": {
  "StatusCode": "Failed",
  "AutoScalingGroupName": "my-asg",
  "ActivityId": "87654321-4321-4321-4321-210987654321",
  "Details": {
    "Availability Zone": "us-west-2b",
    "Subnet ID": "subnet-12345678"
  },
  "RequestId": "12345678-1234-1234-1234-123456789012",
  "StatusMessage": "message-text",
  "EndTime": "yyyy-mm-ddThh:mm:ssZ",
  "EC2InstanceId": "i-1234567890abcdef0",
  "StartTime": "yyyy-mm-ddThh:mm:ssZ",
  "Cause": "description-text",
  "Origin": "EC2",
  "Destination": "AutoScalingGroup"
}
}

```

Acara scale-in yang gagal

Contoh peristiwa berikut menunjukkan bahwa Auto Scaling Amazon EC2 gagal menghentikan instance.

Important

Ketika grup Auto Scaling mengembalikan instance ke kolam hangat dalam skala, gagal mengembalikan instance ke kolam hangat juga dapat menghasilkan peristiwa. EC2 Instance Terminate Unsuccessful Peristiwa yang dikirimkan ketika sebuah instance gagal kembali ke kolam hangat memiliki WarmPool nilai untuk Destination. Untuk informasi selengkapnya, lihat [Instance reuse policy](#).

```
{
```

```
"version": "0",
"id": "12345678-1234-1234-1234-123456789012",
"detail-type": "EC2 Instance Terminate Unsuccessful",
"source": "aws.autoscaling",
"account": "123456789012",
"time": "yyyy-mm-ddThh:mm:ssZ",
"region": "us-west-2",
"resources": [
  "auto-scaling-group-arn",
  "instance-arn"
],
"detail": {
  "StatusCode": "Failed",
  "AutoScalingGroupName": "my-asg",
  "ActivityId": "87654321-4321-4321-4321-210987654321",
  "Details": {
    "Availability Zone": "us-west-2b",
    "Subnet ID": "subnet-12345678"
  },
  "RequestId": "12345678-1234-1234-1234-123456789012",
  "StatusMessage": "message-text",
  "EndTime": "yyyy-mm-ddThh:mm:ssZ",
  "EC2InstanceId": "i-1234567890abcdef0",
  "StartTime": "yyyy-mm-ddThh:mm:ssZ",
  "Cause": "description-text",
  "Origin": "AutoScalingGroup",
  "Destination": "EC2"
}
}
```

Acara penyegaran contoh

Contoh berikut menunjukkan peristiwa untuk fitur penyegaran instance. Acara diproduksi atas dasar upaya terbaik.

Jenis peristiwa

- [Pos pemeriksaan tercapai](#)
- [Penyegaran instance dimulai](#)
- [Penyegaran instance berhasil](#)
- [Penyegaran instance gagal](#)
- [Penyegaran instans dibatalkan](#)

Pos pemeriksaan tercapai

Ketika jumlah instans yang telah diganti mencapai ambang batas persentase yang ditentukan untuk pos pemeriksaan, Auto Scaling Amazon EC2 mengirimkan peristiwa berikut.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Checkpoint Reached",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "InstanceRefreshId": "ab00cf8f-9126-4f3c-8010-dbb8cad6fb86",
    "AutoScalingGroupName": "my-asg",
    "CheckpointPercentage": "50",
    "CheckpointDelay": "300"
  }
}
```

Penyegaran instance dimulai

Saat status penyegaran instans berubah `InProgress`, Auto Scaling Amazon EC2 akan mengirimkan peristiwa berikut.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Started",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
    "AutoScalingGroupName": "my-asg"
  }
}
```

```
}  
}
```

Penyegaran instance berhasil

Saat status penyegaran instans berubah `Succeeded`, Auto Scaling Amazon EC2 akan mengirimkan peristiwa berikut.

```
{  
  "version": "0",  
  "id": "12345678-1234-1234-1234-123456789012",  
  "detail-type": "EC2 Auto Scaling Instance Refresh Succeeded",  
  "source": "aws.autoscaling",  
  "account": "123456789012",  
  "time": "yyyy-mm-ddThh:mm:ssZ",  
  "region": "us-west-2",  
  "resources": [  
    "auto-scaling-group-arn"  
  ],  
  "detail": {  
    "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",  
    "AutoScalingGroupName": "my-asg"  
  }  
}
```

Penyegaran instance gagal

Saat status penyegaran instans berubah `Failed`, Auto Scaling Amazon EC2 akan mengirimkan peristiwa berikut.

```
{  
  "version": "0",  
  "id": "12345678-1234-1234-1234-123456789012",  
  "detail-type": "EC2 Auto Scaling Instance Refresh Failed",  
  "source": "aws.autoscaling",  
  "account": "123456789012",  
  "time": "yyyy-mm-ddThh:mm:ssZ",  
  "region": "us-west-2",  
  "resources": [  
    "auto-scaling-group-arn"  
  ],  
  "detail": {  
    "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
```

```
"AutoScalingGroupName": "my-asg"
}
```

Penyegaran instans dibatalkan

Saat status penyegaran instans berubah `Cancelled`, Auto Scaling Amazon EC2 akan mengirimkan peristiwa berikut.

```
{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Auto Scaling Instance Refresh Cancelled",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "yyyy-mm-ddThh:mm:ssZ",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "InstanceRefreshId": "c613620e-07e2-4ed2-a9e2-ef8258911ade",
    "AutoScalingGroupName": "my-asg"
  }
}
```

Acara dan pola contoh kolam hangat

Auto Scaling Amazon EC2 mendukung beberapa pola yang telah ditentukan di Amazon. EventBridge ini menyederhanakan bagaimana pola acara dibuat. Anda memilih nilai bidang pada formulir, dan EventBridge menghasilkan pola untuk Anda. Saat ini, Amazon EC2 Auto Scaling tidak mendukung pola yang telah ditentukan sebelumnya untuk setiap peristiwa yang dipancarkan oleh grup Auto Scaling dengan kolam hangat. Anda harus memasukkan pola sebagai objek JSON. Bagian ini dan [Buat EventBridge aturan untuk acara kolam hangat](#) topik menunjukkan cara menggunakan pola acara untuk memilih acara dan mengirimkannya ke target.

Untuk membuat EventBridge aturan yang memfilter peristiwa terkait kolam hangat yang EventBridge dikirimkan oleh Auto Scaling Amazon EC2, sertakan bidang `Destination` dan `Origin` dari bagian acara. `detail`

Nilai-nilai `Origin` dan `Destination` dapat sebagai berikut:

EC2 | AutoScalingGroup | WarmPool

Daftar Isi

- [Contoh peristiwa](#)
- [Contoh pola acara](#)

Contoh peristiwa

Saat menambahkan kait siklus hidup ke grup Auto Scaling, Auto Scaling Amazon EC2 akan mengirimkan peristiwa saat instance bertransisi EventBridge ke status tunggu. Untuk informasi selengkapnya, lihat [Gunakan kait siklus hidup dengan kolam hangat](#).

Bagian ini mencakup contoh peristiwa ini ketika grup Auto Scaling Anda memiliki kolam hangat. Peristiwa dipancarkan atas dasar upaya terbaik.

Note

Untuk peristiwa yang dikirimkan oleh Amazon EC2 Auto Scaling saat penskalaan EventBridge berhasil, lihat. [Acara penskalaan yang sukses](#) Untuk acara saat penskalaan tidak berhasil, lihat. [Peristiwa penskalaan yang gagal](#)

Contoh acara

- [Tindakan siklus hidup penskalaan](#)
- [Tindakan siklus hidup skala dalam](#)

Tindakan siklus hidup penskalaan

Peristiwa yang dikirimkan saat instance bertransisi ke status tunggu untuk peristiwa scale-out memiliki nilai EC2 Instance-launch Lifecycle Action untuk detail-type. Dalam detail objek, nilai untuk Destination atribut Origin dan menunjukkan dari mana instance berasal dan ke mana arahnya.

Dalam contoh acara scale-out ini, instance baru diluncurkan dan statusnya berubah `Warmup:Pending:Wait` karena ditambahkan ke kolam hangat. Untuk informasi selengkapnya, lihat [Transisi status siklus hidup untuk instance di kolam hangat](#).

```
{
```

```

"version": "0",
"id": "12345678-1234-1234-1234-123456789012",
"detail-type": "EC2 Instance-launch Lifecycle Action",
"source": "aws.autoscaling",
"account": "123456789012",
"time": "2021-01-13T00:12:37.214Z",
"region": "us-west-2",
"resources": [
  "auto-scaling-group-arn"
],
"detail": {
  "LifecycleActionToken": "71514b9d-6a40-4b26-8523-05e7eEXAMPLE",
  "AutoScalingGroupName": "my-asg",
  "LifecycleHookName": "my-launch-lifecycle-hook",
  "EC2InstanceId": "i-1234567890abcdef0",
  "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
  "NotificationMetadata": "additional-info",
  "Origin": "EC2",
  "Destination": "WarmPool"
}
}

```

Dalam contoh peristiwa scale-out ini, status instance berubah menjadi Pending:Wait karena ditambahkan ke grup Auto Scaling dari kolam hangat. Untuk informasi selengkapnya, lihat [Transisi status siklus hidup untuk instance di kolam hangat](#).

```

{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-launch Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2021-01-19T00:35:52.359Z",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "LifecycleActionToken": "19cc4d4a-e450-4d1c-b448-0de67EXAMPLE",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-launch-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_LAUNCHING",
  }
}

```

```

    "NotificationMetadata": "additional-info",
    "Origin": "WarmPool",
    "Destination": "AutoScalingGroup"
  }
}

```

Tindakan siklus hidup skala dalam

Peristiwa yang dikirimkan saat instance bertransisi ke status tunggu untuk peristiwa skala masuk memiliki nilai `EC2 Instance-terminate Lifecycle Action` untuk `detail-type`. Dalam detail objek, nilai untuk `Destination` atribut `Origin` dan menunjukkan dari mana instance berasal dan ke mana arahnya.

Dalam contoh peristiwa scale-in ini, status instance berubah menjadi `Warmmed:Pending:Wait` karena dikembalikan ke kolam hangat. Untuk informasi selengkapnya, lihat [Transisi status siklus hidup untuk instance di kolam hangat](#).

```

{
  "version": "0",
  "id": "12345678-1234-1234-1234-123456789012",
  "detail-type": "EC2 Instance-terminate Lifecycle Action",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2022-03-28T00:12:37.214Z",
  "region": "us-west-2",
  "resources": [
    "auto-scaling-group-arn"
  ],
  "detail": {
    "LifecycleActionToken": "42694b3d-4b70-6a62-8523-09a1eEXAMPLE",
    "AutoScalingGroupName": "my-asg",
    "LifecycleHookName": "my-termination-lifecycle-hook",
    "EC2InstanceId": "i-1234567890abcdef0",
    "LifecycleTransition": "autoscaling:EC2_INSTANCE_TERMINATING",
    "NotificationMetadata": "additional-info",
    "Origin": "AutoScalingGroup",
    "Destination": "WarmPool"
  }
}

```

Contoh pola acara

Bagian sebelumnya memberikan contoh peristiwa yang dipancarkan oleh Amazon EC2 Auto Scaling.

EventBridge pola acara memiliki struktur yang sama dengan peristiwa yang cocok. Pola mengutip bidang yang ingin Anda cocokkan dan memberikan nilai yang Anda cari.

Bidang berikut dalam acara membentuk pola peristiwa yang didefinisikan dalam aturan untuk memanggil tindakan:

```
"source": "aws.autoscaling"
```

Mengidentifikasi bahwa acara tersebut berasal dari Amazon EC2 Auto Scaling.

```
"detail-type": "EC2 Instance-launch Lifecycle Action"
```

Mengidentifikasi tipe peristiwa.

```
"Origin": "EC2"
```

Mengidentifikasi dari mana instance berasal.

```
"Destination": "WarmPool"
```

Mengidentifikasi ke mana instance akan.

Gunakan contoh pola peristiwa berikut untuk menangkap semua EC2 Instance-launch Lifecycle Action peristiwa yang terkait dengan instance yang memasuki kolam hangat.

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
    "Origin": [ "EC2" ],
    "Destination": [ "WarmPool" ]
  }
}
```

Gunakan pola peristiwa contoh berikut untuk menangkap semua EC2 Instance-launch Lifecycle Action peristiwa yang terkait dengan instance yang meninggalkan kolam hangat karena peristiwa penskalaan.

```
{
```

```

"source": [ "aws.autoscaling" ],
"detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
"detail": {
  "Origin": [ "WarmPool" ],
  "Destination": [ "AutoScalingGroup" ]
}
}

```

Gunakan contoh pola peristiwa berikut untuk menangkap semua EC2 Instance-launch Lifecycle Action peristiwa yang terkait dengan peluncuran instans langsung ke grup Auto Scaling.

```

{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
    "Origin": [ "EC2" ],
    "Destination": [ "AutoScalingGroup" ]
  }
}

```

Gunakan contoh pola peristiwa berikut untuk menangkap semua EC2 Instance-terminate Lifecycle Action peristiwa yang terkait dengan instance yang kembali ke kolam hangat dalam skala.

```

{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-terminate Lifecycle Action" ],
  "detail": {
    "Origin": [ "AutoScalingGroup" ],
    "Destination": [ "WarmPool" ]
  }
}

```

Gunakan contoh pola peristiwa berikut untuk menangkap semua peristiwa yang terkait dengan EC2 Instance-launch Lifecycle Action, terlepas dari asal atau tujuan.

```

{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ]
}

```

Buat EventBridge aturan

Ketika suatu peristiwa dipancarkan oleh Amazon EC2 Auto Scaling, pemberitahuan peristiwa dikirim ke EventBridge Amazon sebagai file JSON. Anda dapat menulis EventBridge aturan untuk mengotomatiskan tindakan apa yang harus diambil ketika pola peristiwa cocok dengan aturan. Jika EventBridge mendeteksi pola peristiwa yang cocok dengan pola yang ditentukan dalam aturan, EventBridge memanggil target (atau target) yang ditentukan dalam aturan.

Anda dapat menggunakan contoh prosedur di bagian ini sebagai titik awal.

Anda juga dapat menemukan dokumentasi berikut berguna.

- Untuk melakukan tindakan kustom pada instance saat diluncurkan atau sebelum dihentikan menggunakan fungsi Lambda, lihat [Tutorial: Konfigurasi kait siklus hidup yang memanggil fungsi Lambda](#)
- Untuk menjalankan fungsi Lambda pada panggilan API yang dicatat CloudTrail, [lihat Tutorial: AWS Log panggilan API EventBridge](#) menggunakan dalam Panduan Pengguna EventBridge Amazon.
- Untuk informasi selengkapnya tentang cara membuat aturan acara, lihat [Membuat EventBridge aturan Amazon yang bereaksi terhadap peristiwa](#) di Panduan EventBridge Pengguna Amazon.

Topik

- [Buat EventBridge aturan untuk acara penyegaran misalnya](#)
- [Buat EventBridge aturan untuk acara kolam hangat](#)

Buat EventBridge aturan untuk acara penyegaran misalnya

Contoh berikut membuat EventBridge aturan untuk mengirim pemberitahuan email. Hal ini dilakukan setiap kali grup Auto Scaling Anda memancarkan peristiwa ketika pos pemeriksaan tercapai selama penyegaran instance. Prosedur untuk mengatur pemberitahuan email menggunakan Amazon SNS disertakan. Untuk menggunakan Amazon SNS guna mengirim notifikasi email, Anda harus membuat topik, lalu berlangganan alamat email Anda sesuai topik.

Untuk informasi selengkapnya tentang fitur penyegaran instans, lihat [Menggunakan penyegaran instans untuk memperbarui instance dalam grup Auto Scaling](#).

Membuat topik Amazon SNS

Topik SNS adalah titik akses logis, saluran komunikasi yang digunakan grup Auto Scaling Anda untuk mengirim notifikasi. Anda membuat topik dengan menentukan nama untuk topik Anda.

Nama topik harus memenuhi persyaratan berikut:

- Memiliki 1-256 karakter
- Memuat huruf besar dan huruf kecil ASCII, angka, garis bawah, atau tanda hubung

Untuk informasi lebih lanjut, lihat [Membuat topik Amazon SNS](#) dalam Panduan Developer Amazon Simple Notification Service.

Berlangganan topik Amazon SNS

Untuk menerima notifikasi yang dikirimkan grup Auto Scaling Anda ke topik, Anda harus melanggankan endpoint ke topik tersebut. Dalam prosedur ini, untuk Endpoint, tentukan alamat email untuk menerima notifikasi dari Amazon EC2 Auto Scaling.

Untuk informasi selengkapnya, lihat [Berlangganan topik Amazon SNS](#) di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

Mengonfirmasi langganan Amazon SNS Anda

Amazon SNS mengirimkan email konfirmasi ke alamat email yang Anda tentukan pada langkah sebelumnya.

Pastikan Anda membuka email dari AWS Notifikasi dan pilih tautan untuk mengonfirmasi langganan sebelum melanjutkan langkah berikutnya.

Anda akan menerima pesan pengakuan dari AWS Amazon SNS kini dikonfigurasi untuk menerima notifikasi dan mengirimkan notifikasi sebagai email ke alamat email yang Anda tentukan.

Rutekan acara ke topik Amazon SNS Anda

Buat aturan yang cocok dengan acara yang dipilih dan rutekan ke topik Amazon SNS Anda untuk memberi tahu alamat email berlangganan.

Untuk membuat aturan yang mengirimkan pemberitahuan ke topik Amazon SNS Anda

1. Buka EventBridge konsol Amazon di <https://console.aws.amazon.com/events/>.
2. Di panel navigasi, pilih Aturan.

3. Pilih Buat aturan.

4. Untuk Tentukan detail aturan, lakukan hal berikut:

- a. Masukkan Nama untuk aturan tersebut dan, secara opsional, deskripsi.

Aturan tidak boleh memiliki nama yang sama dengan aturan lain di Wilayah yang sama dan di bus peristiwa yang sama.

- b. Untuk Bus peristiwa, pilih default. Ketika AWS layanan di akun Anda menghasilkan acara, itu selalu masuk ke bus acara default akun Anda.
- c. Untuk Tipe aturan, pilih Aturan dengan pola peristiwa.
- d. Pilih Selanjutnya.

5. Untuk Pola peristiwa build, lakukan hal berikut ini:

- a. Untuk sumber Acara, pilih AWS acara atau acara EventBridge mitra.

- b. Untuk Pola peristiwa, lakukan hal berikut:

i. Untuk Sumber peristiwa, pilih Layanan AWS.

ii. Untuk Layanan AWS, pilih Auto Scaling.

iii. Untuk jenis Event, pilih Instance Refresh.

iv. Secara default, aturan cocok dengan peristiwa penyegaran instance apa pun. Untuk membuat aturan yang memberi tahu Anda saat pos pemeriksaan tercapai selama penyegaran instans, pilih Peristiwa instans tertentu dan pilih EC2 Auto Scaling Instance Refresh Checkpoint Refresh Dicapai.

v. Secara default, aturan cocok dengan grup Auto Scaling di Wilayah. Untuk membuat aturan cocok dengan grup Auto Scaling tertentu, pilih Nama grup spesifik dan pilih satu atau lebih grup Auto Scaling.

vi. Pilih Selanjutnya.

6. Untuk Pilih target, lakukan hal berikut:

- a. Untuk Tipe target, pilih Layanan AWS.

- b. Untuk Pilih target, pilih topik SNS.

- c. Untuk Topik, pilih topik Amazon SNS Anda.

- d. (Opsional) Pada Pengaturan tambahan, Anda dapat mengonfigurasi pengaturan tambahan secara opsional. Untuk informasi selengkapnya, lihat [Membuat EventBridge aturan Amazon yang bereaksi terhadap peristiwa](#) di Panduan EventBridge Pengguna Amazon.

- e. Pilih Selanjutnya.
7. (Opsional) Untuk Tanda, Anda dapat secara opsional menetapkan satu atau beberapa tanda ke aturan, lalu pilih Berikutnya.
8. Untuk Meninjau dan membuat, tinjau detail aturan dan memodifikasinya seperlunya. Kemudian, pilih Buat aturan.

Buat EventBridge aturan untuk acara kolam hangat

Contoh berikut membuat EventBridge aturan untuk memanggil tindakan terprogram. Ini dilakukan setiap kali grup Auto Scaling Anda memancarkan peristiwa saat instance baru ditambahkan ke kolam hangat.

Sebelum Anda membuat aturan, buat AWS Lambda fungsi yang Anda inginkan aturan untuk digunakan sebagai target. Anda harus menentukan fungsi ini sebagai target aturan. Prosedur berikut hanya menyediakan langkah-langkah untuk membuat EventBridge aturan yang bertindak ketika instance baru memasuki kolam hangat. Untuk tutorial pengantar yang menunjukkan cara membuat fungsi Lambda sederhana untuk dipanggil saat acara masuk cocok dengan aturan, lihat [Tutorial: Konfigurasi kait siklus hidup yang memanggil fungsi Lambda](#)

Untuk informasi lebih lanjut tentang membuat dan bekerja dengan kolam hangat, lihat [Kolam hangat untuk Auto Scaling Amazon EC2](#).

Untuk membuat aturan acara yang memanggil fungsi Lambda

1. Buka EventBridge konsol Amazon di <https://console.aws.amazon.com/events/>.
2. Di panel navigasi, pilih Aturan.
3. Pilih Buat aturan.
4. Untuk Tentukan detail aturan, lakukan hal berikut:
 - a. Masukkan Nama untuk aturan tersebut dan, secara opsional, deskripsi.

Aturan tidak boleh memiliki nama yang sama dengan aturan lain di Wilayah yang sama dan di bus peristiwa yang sama.
 - b. Untuk Bus peristiwa, pilih default. Ketika Layanan AWS di akun Anda menghasilkan acara, itu selalu masuk ke bus acara default akun Anda.
 - c. Untuk Tipe aturan, pilih Aturan dengan pola peristiwa.
 - d. Pilih Selanjutnya.

5. Untuk Pola peristiwa build, lakukan hal berikut ini:
 - a. Untuk sumber Acara, pilih AWS acara atau acara EventBridge mitra.
 - b. Untuk pola Acara, pilih Pola kustom (editor JSON), dan tempelkan pola berikut ke dalam kotak Pola acara, ganti teks dalam *huruf miring* dengan nama grup Auto Scaling Anda.

```
{
  "source": [ "aws.autoscaling" ],
  "detail-type": [ "EC2 Instance-launch Lifecycle Action" ],
  "detail": {
    "AutoScalingGroupName": [ "my-asg" ],
    "Origin": [ "EC2" ],
    "Destination": [ "WarmPool" ]
  }
}
```

Untuk membuat aturan yang cocok untuk acara lain, ubah pola acara. Untuk informasi selengkapnya, lihat [Contoh pola acara](#).

- c. Pilih Berikutnya.
6. Untuk Pilih target, lakukan hal berikut:
 - a. Untuk Tipe target, pilih Layanan AWS.
 - b. Untuk Pilih target, pilih Fungsi Lambda.
 - c. Untuk Fungsi, pilih fungsi yang ingin Anda kirim acara.
 - d. (Opsional) Untuk Konfigurasi versi/alias, masukkan pengaturan versi dan alias untuk fungsi Lambda target.
 - e. (Opsional) Untuk pengaturan tambahan, masukkan pengaturan tambahan yang sesuai untuk aplikasi Anda. Untuk informasi selengkapnya, lihat [Membuat EventBridge aturan Amazon yang bereaksi terhadap peristiwa](#) di Panduan EventBridge Pengguna Amazon.
 - f. Pilih Selanjutnya.
7. (Opsional) Untuk Tanda, Anda dapat secara opsional menetapkan satu atau beberapa tanda ke aturan, lalu pilih Berikutnya.
8. Untuk Meninjau dan membuat, tinjau detail aturan dan memodifikasinya seperlunya. Kemudian, pilih Buat aturan.

Menyediakan konektivitas jaringan untuk instans Auto Scaling menggunakan Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) adalah layanan yang memungkinkan Anda meluncurkan AWS sumber daya seperti grup Auto Scaling dalam jaringan virtual yang terisolasi secara logis yang Anda tentukan.

Subnet di Amazon VPC adalah subdivisi dalam Availability Zone yang ditentukan oleh segmen rentang alamat IP VPC. Dengan menggunakan subnet, Anda dapat mengelompokkan instans berdasarkan kebutuhan keamanan dan operasional Anda. Subnet berada sepenuhnya dalam Availability Zone tempat pembuatannya. Anda meluncurkan instans Auto Scaling di dalam subnet.

Untuk mengaktifkan komunikasi antara internet dan instans dalam subnet Anda, Anda harus membuat gateway internet dan melampirkannya ke VPC Anda. Gateway internet memungkinkan sumber daya Anda di dalam subnet terhubung ke internet melalui jaringan edge Amazon EC2. Jika lalu lintas subnet diarahkan ke gateway internet, subnet dikenal sebagai subnet publik. Jika lalu lintas subnet tidak diarahkan ke gateway internet, subnet dikenal sebagai subnet privat. Menggunakan subnet publik untuk sumber daya yang harus terhubung ke internet, dan subnet privat untuk sumber daya yang tidak perlu terhubung ke internet. Untuk informasi selengkapnya tentang memberikan akses internet ke instans di VPC, [lihat Mengakses internet di](#) Panduan Pengguna VPC Amazon.

Daftar Isi

- [VPC default](#)
- [VPC Non-default](#)
- [Pertimbangan saat memilih subnet VPC](#)
- [Pembuatan alamat IP di VPC](#)
- [Antarmuka jaringan dalam VPC](#)
- [Tenansi penempatan instans](#)
- [AWS Outposts](#)
- [Lebih banyak sumber daya untuk mempelajari VPC](#)

VPC default

Jika Anda membuat Akun AWS setelah 4 Desember 2013 atau Anda membuat grup Auto Scaling Anda di yang baru Wilayah AWS, kami membuat VPC default untuk Anda. VPC bawaan Anda

dilengkapi dengan subnet default di setiap Availability Zone. Jika Anda memiliki VPC default, grup Auto Scaling Anda dibuat di VPC secara default.

Anda dapat melihat VPC Anda di [halaman VPC Anda](#) di konsol VPC Amazon.

Untuk informasi selengkapnya tentang VPC default, lihat [VPC default di Panduan Pengguna Amazon VPC](#).

VPC Non-default

Anda dapat memilih untuk membuat VPC tambahan dengan membuka halaman [Dasbor VPC](#) di AWS Management Console dan memilih Buat VPC.

Untuk informasi selengkapnya, silakan lihat ACL Jaringan di [Panduan Pengguna Amazon VPC](#).

Note

VPC mencakup semua Availability Zone di dalamnya. Wilayah AWS Saat Anda menambahkan subnet ke VPC Anda, pilih beberapa Availability Zone untuk memastikan bahwa aplikasi yang dihosting di subnet tersebut sangat tersedia. Zona Ketersediaan adalah satu atau lebih pusat data diskret dengan daya redundan, jaringan, dan konektivitas dalam Wilayah AWS. Availability Zones membantu Anda membuat aplikasi produksi sangat tersedia, toleran terhadap kesalahan, dan skalabel.

Pertimbangan saat memilih subnet VPC

Perhatikan pertimbangan berikut saat memilih subnet VPC untuk grup Auto Scaling Anda:

- Jika Anda melampirkan penyeimbang beban Elastic Load Balancing ke grup Auto Scaling Anda, instans dapat diluncurkan ke subnet publik atau pribadi. Namun, penyeimbang beban harus dibuat di subnet publik untuk mendukung resolusi DNS.
- Jika Anda mengakses instans Auto Scaling secara langsung melalui SSH, instans hanya dapat diluncurkan ke subnet publik.
- Jika Anda mengakses instans Auto Scaling no-ingress menggunakan AWS Systems Manager Session Manager, instans dapat diluncurkan ke subnet publik atau pribadi.
- Jika Anda menggunakan subnet pribadi, Anda dapat mengizinkan instans Auto Scaling mengakses internet dengan menggunakan gateway NAT publik.

- Secara default, subnet default dalam VPC default adalah subnet publik.

Pembuatan alamat IP di VPC

Saat meluncurkan instans Auto Scaling di VPC, instans Anda secara otomatis diberi alamat IP pribadi dari rentang CIDR subnet tempat instance diluncurkan. Hal ini memungkinkan instans Anda untuk berkomunikasi dengan instans lain di VPC.

Anda dapat mengonfigurasi template peluncuran atau konfigurasi peluncuran untuk menetapkan alamat IPv4 publik ke instance Anda. Menetapkan alamat IP publik ke instans Anda memungkinkan mereka untuk berkomunikasi dengan internet atau layanan lainnya AWS .

Saat Anda meluncurkan instance ke subnet yang dikonfigurasi untuk secara otomatis menetapkan alamat IPv6, mereka menerima alamat IPv4 dan IPv6. Jika tidak, mereka hanya menerima alamat IPv4. Untuk informasi selengkapnya, lihat [alamat IPv6](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

[Untuk informasi tentang menentukan rentang CIDR untuk VPC atau subnet Anda, lihat Panduan Pengguna Amazon VPC.](#)

Auto Scaling Amazon EC2 dapat secara otomatis menetapkan alamat IP pribadi tambahan pada peluncuran instans saat Anda menggunakan templat peluncuran yang menentukan antarmuka jaringan tambahan. Setiap antarmuka jaringan diberi satu alamat IP pribadi dari rentang CIDR subnet di mana instance diluncurkan. Dalam hal ini, sistem tidak dapat lagi menetapkan alamat IPv4 publik secara otomatis ke antarmuka jaringan utama. Anda tidak akan dapat terhubung ke instans Anda melalui alamat IPv4 publik kecuali Anda mengaitkan alamat IP Elastis yang tersedia ke instans Auto Scaling.

Antarmuka jaringan dalam VPC

Setiap instance di VPC Anda memiliki antarmuka jaringan default (antarmuka jaringan utama). Anda tidak dapat melepaskan antarmuka jaringan primer dari sebuah instans. Anda dapat membuat dan melampirkan antarmuka jaringan tambahan untuk setiap instans di VPC Anda. Jumlah antarmuka jaringan yang dapat Anda lampirkan bervariasi tergantung tipe instans.

Saat meluncurkan instance menggunakan template peluncuran, Anda dapat menentukan antarmuka jaringan tambahan. Namun, meluncurkan instance Auto Scaling dengan beberapa antarmuka jaringan secara otomatis membuat setiap antarmuka di subnet yang sama dengan instance. Ini karena Amazon EC2 Auto Scaling mengabaikan subnet yang ditentukan dalam template peluncuran

demikian apa yang ditentukan dalam grup Auto Scaling. Untuk informasi lebih lanjut, lihat [Membuat templat peluncuran untuk Grup Auto Scaling](#).

Jika Anda membuat atau melampirkan dua atau lebih antarmuka jaringan dari subnet yang sama ke sebuah instans, Anda mungkin mengalami masalah jaringan seperti perutean asimetris, terutama pada instance yang menggunakan varian Linux non-Amazon. Jika Anda memerlukan konfigurasi jenis ini, Anda harus mengkonfigurasi antarmuka jaringan sekunder di dalam OS. Sebagai contoh, [lihat Bagaimana cara membuat antarmuka jaringan sekunder saya berfungsi di instance Ubuntu EC2 saya?](#) di pusat AWS pengetahuan.

Tenansi penempatan instans

Secara default, semua instans dalam VPC berjalan sebagai instans tenan berbagi. Amazon EC2 Auto Scaling juga mendukung Instans Khusus dan Host Khusus. Untuk informasi selengkapnya, lihat [Buat template peluncuran menggunakan pengaturan lanjutan](#).

AWS Outposts

AWS Outposts memperluas VPC Amazon dari AWS Wilayah ke Pos Luar dengan komponen VPC yang dapat diakses di Wilayah, termasuk gateway internet, gateway pribadi virtual, Gateway Transit VPC Amazon, dan titik akhir VPC. Pos Luar ditempatkan ke Availability Zone di Region dan merupakan perpanjangan dari Availability Zone yang dapat Anda gunakan untuk ketahanan.

Untuk informasi selengkapnya, silakan lihat [Panduan Pengguna AWS Outposts](#).

Untuk contoh cara menyebarkan grup Auto Scaling yang melayani lalu lintas dari Application Load Balancer dalam Outpost, lihat posting [blog berikut Mengonfigurasi Application Load Balancer aktif](#).
AWS Outposts

Lebih banyak sumber daya untuk mempelajari VPC

Gunakan topik berikut untuk mempelajari lebih lanjut tentang VPC dan subnet.

- Subnet privat dalam VPC
 - [Contoh: VPC dengan server di subnet pribadi dan NAT](#)
 - [Gateway NAT](#)
- Subnet publik dalam VPC
 - [Contoh: VPC untuk lingkungan pengujian](#)

- [Contoh: VPC untuk server web dan database](#)
- Subnet untuk Application Load Balancer
 - [Subnet untuk penyeimbang beban Anda](#)
- Informasi VPC umum
 - [Panduan Pengguna Amazon VPC](#)
 - [Connect VPC menggunakan VPC peering](#)
 - [Antarmuka jaringan elastis](#)
 - [Gunakan titik akhir VPC untuk konektivitas pribadi](#)

Keamanan di Amazon EC2 Auto Scaling

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

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

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [program AWS kepatuhan program AWS](#) . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Auto Scaling Amazon EC2, [AWS lihat layanan dalam cakupan berdasarkan kepatuhan](#).
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan 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 Amazon EC2 Auto Scaling. Topik-topik berikut menunjukkan cara mengonfigurasi Amazon EC2 Auto Scaling untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Auto Scaling Amazon EC2 Anda.

Topik

- [Keamanan infrastruktur di Amazon EC2 Auto Scaling](#)
- [Ketahanan di Amazon EC2 Auto Scaling](#)
- [Perlindungan data di Amazon EC2 Auto Scaling](#)
- [Manajemen Identitas dan Akses untuk Amazon EC2 Auto Scaling](#)
- [Validasi kepatuhan untuk Amazon EC2 Auto Scaling](#)
- [Amazon EC2 Auto Scaling dan endpoint VPC antarmuka](#)

Keamanan infrastruktur di Amazon EC2 Auto Scaling

Sebagai layanan terkelola, Amazon EC2 Auto Scaling dilindungi AWS oleh keamanan jaringan global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur dalam Kerangka Kerja](#) yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Amazon EC2 Auto Scaling 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 dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Anda juga dapat menggunakan titik akhir virtual private cloud (VPC) untuk Amazon EC2 Auto Scaling. Titik akhir VPC antarmuka memungkinkan sumber daya VPC Amazon Anda menggunakan alamat IP pribadinya untuk mengakses Auto Scaling Amazon EC2 tanpa eksposur ke internet publik. Lihat informasi yang lebih lengkap di [Amazon EC2 Auto Scaling dan endpoint VPC antarmuka](#)

Sumber daya terkait

Untuk informasi tentang fitur untuk mengisolasi lalu lintas layanan yang disediakan oleh Amazon EC2, [lihat Keamanan infrastruktur di Amazon EC2 di Panduan Pengguna Amazon EC2](#) untuk Instans Linux.

Ketahanan di Amazon EC2 Auto Scaling

Infrastruktur AWS global dibangun di sekitar Wilayah AWS dan Availability Zones. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan

Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

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

Untuk mendapatkan manfaat dari redundansi geografis desain Availability Zone, lakukan hal berikut:

- Rentang grup Auto Scaling Anda di beberapa Availability Zone.
- Pertahankan setidaknya satu instance di setiap Availability Zone.
- Pasang penyeimbang beban untuk mendistribusikan lalu lintas masuk di Availability Zone yang sama. Jika Anda menggunakan Application Load Balancer, pastikan setiap instans EC2 mendapatkan jumlah lalu lintas yang sama dengan menjaga penyeimbangan beban lintas zona diaktifkan. Ini membantu membatasi dampak peningkatan beban pada instans yang ada selama peristiwa failover dan menghasilkan ketahanan yang lebih besar daripada tanpa penyeimbangan beban lintas zona.
- Pastikan pemeriksaan kesehatan Elastic Load Balancing dikonfigurasi dengan benar, dan juga bahwa pemeriksaan tersebut diaktifkan pada grup Auto Scaling. Kemudian, jika sebuah instans gagal dalam pemeriksaan kesehatannya, Elastic Load Balancing berhenti mengirimkan lalu lintas ke instans tersebut dan mengalihkan lalu lintas ke instans yang sehat, sementara Amazon EC2 Auto Scaling menggantikan instans yang tidak sehat.

Auto Scaling Amazon EC2 membantu mendukung kebutuhan ketahanan aplikasi Anda dengan cara berikut:

- Memeriksa contoh untuk masalah kesehatan dan jangkauan. Ketika sebuah instance menjadi tidak sehat, secara otomatis menghentikan instance dan meluncurkan yang baru.
- Jika kebijakan penskalaan dinamis berlaku, secara otomatis menskalakan kapasitas sesuai dengan lalu lintas yang masuk.
- Mendeteksi masalah dalam keandalan CloudWatch metrik Amazon yang mendukung kebijakan penskalaan dan menghentikan aktivitas penskalaan saat metrik andal tidak tersedia, seperti saat titik data hilang.
- Mencoba mempertahankan jumlah instans yang setara di setiap Availability Zone yang diaktifkan saat grup Anda menskalakan.

- Menggunakan Availability Zone untuk menjaga ketersediaan tinggi. Jika Availability Zone menjadi tidak sehat, Auto Scaling Amazon EC2 melakukan hal berikut:
 - Meluncurkan instance baru di Availability Zone berbeda yang diaktifkan untuk grup Auto Scaling Anda.
 - Mendistribusikan ulang instance di semua Availability Zone yang diaktifkan saat Availability Zone yang tidak sehat kembali ke status sehat.
- Terus mencoba meluncurkan instance di Availability Zone lain yang diaktifkan jika instance gagal diluncurkan di Availability Zone tertentu.
- Secara otomatis mendaftarkan dan membatalkan pendaftaran instans dengan penyeimbang beban yang terkait dengan grup Auto Scaling Anda. Dengan cara ini, Anda tidak perlu mendaftar dan membatalkan pendaftaran instance secara terpisah.

Sumber daya terkait

Untuk informasi tentang fitur yang membantu mendukung kebutuhan ketahanan data yang disediakan oleh Amazon EBS, lihat [Ketahanan di Amazon Elastic Block Store di Panduan Pengguna Amazon EBS](#).

Perlindungan data di Amazon EC2 Auto Scaling

[Model tanggung jawab AWS bersama model tanggung](#) berlaku untuk perlindungan data di Auto Scaling Amazon EC2. 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 Amazon EC2 Auto Scaling atau Layanan AWS lainnya menggunakan konsol, API AWS CLI, atau SDK. AWS 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.

Saat meluncurkan instans Amazon EC2, Anda memiliki opsi untuk meneruskan data pengguna ke instans untuk melakukan konfigurasi tambahan saat instance melakukan boot. Kami juga menyarankan agar Anda tidak pernah memasukkan informasi rahasia atau sensitif ke dalam data pengguna yang akan diteruskan ke sebuah instans.

Gunakan AWS KMS keys untuk mengenkripsi volume Amazon EBS

Anda dapat mengonfigurasi grup Auto Scaling Anda untuk mengenkripsi data volume Amazon EBS yang disimpan di cloud. AWS KMS keys Auto Scaling Amazon EC2 AWS mendukung kunci terkelola dan terkelola pelanggan untuk mengenkripsi data Anda. Perhatikan bahwa KmsKeyId opsi untuk menentukan kunci terkelola pelanggan tidak tersedia saat Anda menggunakan konfigurasi peluncuran. Untuk menentukan kunci terkelola pelanggan Anda, gunakan template peluncuran sebagai gantinya. Untuk informasi selengkapnya, lihat [Membuat template peluncuran untuk grup Auto Scaling](#). Untuk informasi tentang cara membuat, menyimpan, dan mengelola kunci AWS KMS enkripsi, lihat [Panduan AWS Key Management Service Pengembang](#).

Anda juga dapat mengonfigurasi kunci terkelola pelanggan di AMI yang didukung EBS sebelum menyiapkan templat peluncuran atau konfigurasi peluncuran, atau menggunakan enkripsi secara default untuk menerapkan enkripsi volume EBS baru dan salinan snapshot yang Anda buat. Untuk informasi selengkapnya, lihat [Menggunakan enkripsi dengan AMI yang didukung EBS](#) di Panduan Pengguna Amazon EC2 untuk Instans dan Enkripsi Linux [secara default](#) di Panduan Pengguna Amazon EBS.

Note

Untuk informasi tentang cara menyiapkan kebijakan kunci yang Anda perlukan untuk meluncurkan instance Auto Scaling saat Anda menggunakan kunci terkelola pelanggan untuk enkripsi, lihat [Kebijakan AWS KMS kunci yang diperlukan untuk digunakan dengan volume terenkripsi](#)

Sumber daya terkait

Untuk panduan perlindungan data yang disediakan oleh Amazon EBS, lihat [Perlindungan data di Amazon Elastic Block Store](#) di Panduan Pengguna Amazon EBS.

Kebijakan AWS KMS kunci yang diperlukan untuk digunakan dengan volume terenkripsi

Auto Scaling Amazon EC2 [menggunakan peran terkait layanan untuk mendelegasikan izin](#) ke orang lain. Layanan AWS Peran terkait layanan Auto Scaling Amazon EC2 telah ditentukan sebelumnya dan menyertakan izin yang diperlukan oleh Amazon EC2 Auto Scaling untuk memanggil orang lain atas nama Anda. Layanan AWS Izin yang telah ditentukan juga mencakup akses ke Anda. Kunci yang dikelola AWS Namun, mereka tidak menyertakan akses ke kunci yang dikelola pelanggan Anda, memungkinkan Anda untuk mempertahankan kontrol penuh atas kunci ini.

Topik ini menjelaskan cara menyiapkan kebijakan kunci yang Anda perlukan untuk meluncurkan instans Auto Scaling saat Anda menentukan kunci terkelola pelanggan untuk enkripsi Amazon EBS.

Note

Auto Scaling Amazon EC2 tidak memerlukan otorisasi tambahan untuk menggunakan Kunci yang dikelola AWS default guna melindungi volume terenkripsi di akun Anda.

Daftar Isi

- [Gambaran Umum](#)
- [Konfigurasi kebijakan utama](#)
- [Contoh 1: Bagian kebijakan utama yang memungkinkan akses ke kunci yang dikelola pelanggan](#)
- [Contoh 2: Bagian kebijakan utama yang memungkinkan akses lintas akun ke kunci yang dikelola pelanggan](#)
- [Edit kebijakan utama di AWS KMS konsol](#)

Gambaran Umum

Berikut ini AWS KMS keys dapat digunakan untuk enkripsi Amazon EBS saat Amazon EC2 Auto Scaling meluncurkan instans:

- [Kunci yang dikelola AWS](#)— Kunci enkripsi di akun Anda yang dibuat, dimiliki, dan dikelola Amazon EBS. Ini adalah kunci enkripsi default untuk akun baru. Kunci yang dikelola AWS ini digunakan untuk enkripsi kecuali Anda menentukan kunci yang dikelola pelanggan.
- [Kunci terkelola pelanggan](#) — Kunci enkripsi khusus yang Anda buat, miliki, dan kelola. Untuk informasi selengkapnya, lihat [Membuat kunci](#) di Panduan AWS Key Management Service Pengembang.

Catatan: Kuncinya harus simetris. Amazon EBS tidak mendukung kunci yang dikelola pelanggan asimetris.

Anda mengonfigurasi kunci terkelola pelanggan saat membuat snapshot terenkripsi atau templat peluncuran yang menentukan volume terenkripsi, atau mengaktifkan enkripsi secara default.

Konfigurasi kebijakan utama

Kunci KMS Anda harus memiliki kebijakan utama yang memungkinkan Auto Scaling Amazon EC2 meluncurkan instans dengan volume Amazon EBS yang dienkripsi dengan kunci yang dikelola pelanggan.

Gunakan contoh di halaman ini untuk mengonfigurasi kebijakan utama agar dapat memberikan akses Auto Scaling Amazon EC2 ke kunci terkelola pelanggan Anda. Anda dapat mengubah kebijakan kunci yang dikelola pelanggan baik ketika kunci dibuat atau di lain waktu.

Anda harus, setidaknya, menambahkan dua pernyataan kebijakan ke kebijakan utama agar dapat berfungsi dengan Auto Scaling Amazon EC2.

- Pernyataan pertama memungkinkan identitas IAM yang ditentukan dalam `Principal` elemen untuk menggunakan kunci yang dikelola pelanggan secara langsung. Ini termasuk izin untuk melakukan `AWS KMS Encrypt`, `Decrypt`, `ReEncrypt*`, `GenerateDataKey*`, dan `DescribeKey` operasi pada kunci.
- Pernyataan kedua memungkinkan identitas IAM yang ditentukan dalam `Principal` elemen untuk menggunakan `CreateGrant` operasi untuk menghasilkan hibah yang mendelegasikan subset dari izinnya sendiri untuk Layanan AWS yang terintegrasi dengan atau prinsipal lain. AWS KMS Ini memungkinkan mereka untuk menggunakan kunci untuk membuat sumber daya terenkripsi atas nama Anda.

Saat Anda menambahkan pernyataan kebijakan baru ke kebijakan utama Anda, jangan mengubah pernyataan yang ada dalam kebijakan tersebut.

Untuk setiap contoh berikut, argumen yang harus diganti, seperti ID kunci atau nama peran terkait layanan, ditampilkan sebagai teks *placeholder pengguna*. Dalam kebanyakan kasus, Anda dapat mengganti nama peran terkait layanan dengan nama peran terkait layanan Amazon EC2 Auto Scaling.

Untuk informasi selengkapnya, lihat sumber daya berikut:

- Untuk membuat kunci dengan AWS CLI, lihat [create-key](#).
- Untuk memperbarui kebijakan utama dengan AWS CLI, lihat [put-key-policy](#).
- Untuk menemukan ID kunci dan Nama Sumber Daya Amazon (ARN), lihat [Menemukan ID kunci dan ARN](#) di Panduan Pengembang.AWS Key Management Service
- Untuk informasi tentang peran terkait layanan Amazon EC2 Auto Scaling, lihat [Peran terkait layanan untuk Amazon EC2 Auto Scaling](#).
- [Untuk informasi tentang enkripsi Amazon EBS dan KMS secara umum, enkripsi Amazon EBS di Panduan Pengguna Amazon EBS dan Panduan Pengembang.AWS Key Management Service](#)

Contoh 1: Bagian kebijakan utama yang memungkinkan akses ke kunci yang dikelola pelanggan

Tambahkan dua pernyataan kebijakan berikut ke kebijakan kunci yang dikelola pelanggan, ganti contoh ARN dengan ARN dari peran terkait layanan yang sesuai yang diizinkan akses ke kunci. Dalam contoh ini, bagian kebijakan memberikan `AWSServiceRoleForAutoScaling` bernama peran terkait layanan untuk menggunakan kunci terkelola pelanggan.

```
{
  "Sid": "Allow service-linked role use of the customer managed key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::account-id:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

```
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::account-id:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling"
    ]
  },
  "Action": [
    "kms:CreateGrant"
  ],
  "Resource": "*",
  "Condition": {
    "Bool": {
```



```

    "kms:GrantIsForAWSResource": true
  }
}
}

```

Contoh 2: Bagian kebijakan utama yang memungkinkan akses lintas akun ke kunci yang dikelola pelanggan

Jika Anda membuat kunci terkelola pelanggan di akun yang berbeda dari grup Auto Scaling, Anda harus menggunakan hibah yang dikombinasikan dengan kebijakan kunci untuk mengizinkan akses lintas akun ke kunci tersebut.

Ada dua langkah yang harus diselesaikan dengan urutan sebagai berikut:

1. Pertama, tambahkan dua pernyataan kebijakan berikut ke kebijakan kunci yang dikelola pelanggan. Ganti contoh ARN dengan ARN dari akun lain, pastikan untuk mengganti **111122223333** dengan ID akun aktual yang Akun AWS ingin Anda buat grup Auto Scaling. Ini memungkinkan Anda untuk memberikan pengguna IAM atau peran dalam izin akun yang ditentukan untuk membuat hibah untuk kunci menggunakan perintah CLI yang mengikuti. Namun, ini tidak dengan sendirinya memberi pengguna akses ke kunci.

```

{
  "Sid": "Allow external account 111122223333 use of the customer managed key",
  "Effect": "Allow",
  "Principal": {
    "AWS": [
      "arn:aws:iam::111122223333:root"
    ]
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}

```

```

{

```

```

    "Sid": "Allow attachment of persistent resources in external
account 111122223333",
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::111122223333:root"
      ]
    },
    "Action": [
      "kms:CreateGrant"
    ],
    "Resource": "*"
  }

```

2. Kemudian, dari akun tempat Anda ingin membuat grup Auto Scaling, buat hibah yang mendelegasikan izin yang relevan ke peran terkait layanan yang sesuai. Elemen `GranteePrincipal` pemberian izin adalah ARN dari peran terkait layanan yang sesuai. `key-id` adalah ARN kuncinya.

*Berikut ini adalah contoh perintah CLI [create-grant](#) yang memberikan peran terkait layanan yang **AWSServiceRoleForAutoScaling** dinamai di akun **111122223333** izin untuk menggunakan kunci terkelola pelanggan di akun **444455556666**.*

```

aws kms create-grant \
  --region us-west-2 \
  --key-id arn:aws:kms:us-
west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d \
  --grantee-principal arn:aws:iam::<111122223333:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling \
  --operations "Encrypt" "Decrypt" "ReEncryptFrom" "ReEncryptTo" "GenerateDataKey"
  "GenerateDataKeyWithoutPlaintext" "DescribeKey" "CreateGrant"

```

Agar perintah ini berhasil, pengguna yang mengajukan permintaan harus memiliki izin untuk tindakan `CreateGrant`.

*Contoh berikut kebijakan IAM memungkinkan identitas IAM (pengguna atau peran) di akun **111122223333** untuk membuat hibah untuk kunci yang dikelola pelanggan di akun **444455556666**.*

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "AllowCreationOfGrantForTheKMSKeyinExternalAccount444455556666",
    "Effect": "Allow",
    "Action": "kms:CreateGrant",
    "Resource": "arn:aws:kms:us-
west-2:444455556666:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
  }
]
```

Untuk informasi selengkapnya tentang membuat hibah untuk kunci KMS yang berbeda Akun AWS, lihat [Hibah AWS KMS di Panduan AWS Key Management Service](#) Pengembang.

Important

Nama peran terkait layanan yang ditentukan sebagai kepala penerima hibah harus merupakan nama peran yang ada. Setelah membuat hibah, untuk memastikan bahwa hibah memungkinkan Auto Scaling Amazon EC2 menggunakan kunci KMS yang ditentukan, jangan hapus dan buat ulang peran terkait layanan.

Edit kebijakan utama di AWS KMS konsol

Contoh di bagian sebelumnya hanya menunjukkan cara menambahkan pernyataan ke kebijakan kunci, yang merupakan salah satu cara untuk mengubah kebijakan kunci. Cara termudah untuk mengubah kebijakan kunci adalah dengan menggunakan tampilan default AWS KMS konsol untuk kebijakan utama dan menjadikan identitas IAM (pengguna atau peran) sebagai salah satu pengguna utama untuk kebijakan kunci yang sesuai. Untuk informasi selengkapnya, lihat [Menggunakan tampilan AWS Management Console default](#) di Panduan AWS Key Management Service Pengembang.

Important

Berhati-hatilah. Pernyataan kebijakan tampilan default konsol menyertakan izin untuk melakukan AWS KMS Revoke operasi pada kunci terkelola pelanggan. Jika Anda memberikan Akun AWS akses ke kunci yang dikelola pelanggan di akun Anda, dan Anda secara tidak sengaja mencabut hibah yang memberi mereka izin ini, pengguna eksternal

tidak dapat lagi mengakses data terenkripsi mereka atau kunci yang digunakan untuk mengenkripsi data mereka.

Manajemen Identitas dan Akses untuk Amazon EC2 Auto Scaling

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 (login) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya Amazon EC2 Auto Scaling. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Untuk menggunakan Auto Scaling Amazon EC2, Anda memerlukan dan Akun AWS kredensi keamanan Anda untuk masuk ke akun Anda. Untuk informasi selengkapnya, lihat [kredensi AWS keamanan](#) di Panduan Pengguna IAM.

Untuk dokumentasi lengkap IAM, lihat [Panduan Pengguna IAM](#).

Kontrol akses

Anda dapat memiliki kredensial yang valid untuk mengautentikasi permintaan Anda, tetapi kecuali jika Anda memiliki izin, Anda tidak dapat membuat atau mengakses sumber daya Amazon EC2 Auto Scaling. Misalnya, Anda harus memiliki izin untuk membuat grup Auto Scaling, meluncurkan instance dengan template peluncuran, dan sebagainya.

Bagian berikut menyediakan detail tentang bagaimana administrator IAM dapat menggunakan IAM untuk membantu mengamankan sumber daya Amazon EC2 Auto Scaling, dengan mengontrol siapa yang dapat melakukan tindakan Amazon EC2 Auto Scaling.

Kami menyarankan agar Anda membaca topik Amazon EC2 terlebih dahulu. Lihat [Manajemen identitas dan akses untuk Amazon EC2](#) dalam Panduan Pengguna Amazon EC2 untuk Instance Linux. Setelah membaca topik di bagian ini, Anda harus memiliki ide bagus tentang izin kontrol akses yang ditawarkan Amazon EC2 dan bagaimana hal tersebut dapat cocok dengan izin sumber daya Amazon EC2 Auto Scaling Anda.

Topik

- [Cara kerja Amazon EC2 Auto Scaling dengan IAM](#)

- [Izin API Auto Scaling Amazon EC2](#)
- [AWS kebijakan terkelola untuk Auto Scaling Amazon EC2](#)
- [Peran terkait layanan untuk Amazon EC2 Auto Scaling](#)
- [Contoh kebijakan Amazon EC2 Auto Scaling berbasis identitas](#)
- [Pencegahan confused deputy lintas layanan](#)
- [Luncurkan dukungan templat](#)
- [Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2](#)

Cara kerja Amazon EC2 Auto Scaling dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Amazon EC2 Auto Scaling, pelajari fitur IAM yang tersedia untuk digunakan dengan Amazon EC2 Auto Scaling.

Fitur IAM yang dapat Anda gunakan dengan Amazon EC2 Auto Scaling

Fitur IAM	Dukungan Auto Scaling Amazon EC2
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)	Parsial
Kredensial sementara	Ya
Peran layanan	Ya
Peran terkait layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Auto Scaling Amazon EC2 dan fitur Layanan AWS lainnya dengan sebagian besar fitur IAM, [Layanan AWS lihat fitur tersebut berfungsi dengan IAM di Panduan Pengguna IAM](#).

Kebijakan berbasis identitas untuk Auto Scaling Amazon EC2

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 pengguna dan peran, di sumber daya mana, dan dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, serta ketentuan terkait jenis tindakan yang diizinkan atau ditolak. Anda tidak dapat menentukan pengguna utama dalam kebijakan berbasis identitas karena kebijakan ini berlaku untuk pengguna atau peran yang dilampiri kebijakan. Untuk mempelajari semua elemen yang dapat digunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan JSON IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya dalam Auto Scaling Amazon EC2

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 yang dilampiri kebijakan tersebut, kebijakan ini menentukan jenis tindakan yang dapat dilakukan oleh pengguna utama tertentu di sumber daya tersebut dan apa ketentuannya. Anda harus [menentukan pengguna utama](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau entitas IAM di akun lain sebagai pengguna utama dalam kebijakan berbasis sumber daya. Menambahkan

pengguna utama lintas akun ke kebijakan berbasis sumber daya bagian 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. Izin diberikan dengan melampirkan kebijakan berbasis identitas ke entitas tersebut. Namun, jika kebijakan berbasis sumber daya memberikan akses kepada pengguna utama dalam akun yang sama, kebijakan berbasis identitas lainnya tidak diperlukan. Untuk informasi selengkapnya, lihat [Perbedaan peran IAM dengan kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

Tindakan kebijakan untuk Auto Scaling Amazon EC2

Mendukung tindakan kebijakan Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam 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 suatu kebijakan untuk memberikan izin melakukan operasi terkait.

Untuk melihat daftar tindakan Auto Scaling Amazon EC2, [lihat Tindakan yang ditentukan oleh Amazon EC2 Auto Scaling di Referensi Otorisasi Layanan](#).

Tindakan kebijakan di Amazon EC2 Auto Scaling menggunakan awalan berikut sebelum tindakan:

```
autoscaling
```

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

```
"Action": [  
  "autoscaling:action1",
```

```
"autoscaling:action2"  
]
```

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

```
"Action": "autoscaling:Describe*"
```

Sumber daya kebijakan untuk Auto Scaling Amazon EC2

Mendukung sumber daya kebijakan

Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek atau beberapa 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 mengindikasikan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Anda dapat menggunakan ARN untuk mengidentifikasi grup Auto Scaling dan meluncurkan konfigurasi yang berlaku untuk kebijakan IAM.

Grup Auto Scaling memiliki ARN berikut.

```
"Resource": "arn:aws:autoscaling:region:account-  
id:autoScalingGroup:uuid:autoScalingGroupName/asg-name"
```

Konfigurasi peluncuran memiliki ARN berikut.


```
"Resource": "arn:aws:autoscaling:region:account-id:launchConfiguration:uuid:launchConfigurationName/lc-name"
```

Untuk menentukan grup Auto Scaling dengan `CreateAutoScalingGroup` tindakan, Anda harus mengganti UUID dengan wildcard (*) seperti yang ditunjukkan pada contoh berikut.

```
"Resource": "arn:aws:autoscaling:region:account-id:autoScalingGroup:*:autoScalingGroupName/asg-name"
```

Untuk menentukan konfigurasi peluncuran dengan `CreateLaunchConfiguration` tindakan, Anda harus mengganti UUID dengan wildcard (*) seperti yang ditunjukkan pada contoh berikut.

```
"Resource": "arn:aws:autoscaling:region:account-id:launchConfiguration:*:launchConfigurationName/lc-name"
```

Untuk informasi selengkapnya tentang jenis sumber daya Auto Scaling Amazon EC2 dan ARNnya, [lihat Sumber daya yang ditentukan oleh Auto Scaling Amazon EC2](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditentukan oleh Auto Scaling Amazon EC2](#).

Note

Untuk contoh kebijakan IAM yang menggunakan ARN untuk mengontrol akses ke grup Auto Scaling, lihat. [Kontrol grup Auto Scaling mana yang dapat dihapus](#)

Tidak semua tindakan Amazon EC2 Auto Scaling mendukung izin tingkat sumber daya. Untuk tindakan yang tidak mendukung izin tingkat sumber daya, Anda harus menggunakan wildcard () sebagai sumber daya. *

Tindakan Amazon EC2 Auto Scaling berikut tidak mendukung izin tingkat sumber daya.

- `DescribeAccountLimits`
- `DescribeAdjustmentTypes`
- `DescribeAutoScalingGroups`
- `DescribeAutoScalingInstances`

- DescribeAutoScalingNotificationTypes
- DescribeInstanceRefreshes
- DescribeLaunchConfigurations
- DescribeLifecycleHooks
- DescribeLifecycleHookTypes
- DescribeLoadBalancers
- DescribeLoadBalancerTargetGroups
- DescribeMetricCollectionTypes
- DescribeNotificationConfigurations
- DescribePolicies
- DescribeScalingActivities
- DescribeScalingProcessTypes
- DescribeScheduledActions
- DescribeTags
- DescribeTerminationPolicyTypes
- DescribeWarmPool

Kunci kondisi kebijakan untuk Auto Scaling Amazon EC2

Mendukung kunci kondisi kebijakan spesifik layanan	Ya
--	----

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Condition` (atau blok `Condition`) memungkinkan Anda menentukan kondisi di mana suatu pernyataan akan diterapkan. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi kondisional 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 satu pernyataan, atau beberapa kunci dalam satu elemen `Condition`, AWS akan mengevaluasinya dengan 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. Misalnya, 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](#) di 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.

Auto Scaling Amazon EC2 mendukung kunci kondisi berikut yang dapat digunakan untuk mengontrol akses ke tindakan yang didukung dan menerapkan konfigurasi grup Auto Scaling:

- `autoscaling:InstanceTypes`
- `autoscaling:LaunchConfigurationName`
- `autoscaling:LaunchTemplateVersionSpecified`
- `autoscaling:LoadBalancerNames`
- `autoscaling:MaxSize`
- `autoscaling:MinSize`
- `autoscaling:ResourceTag/key-name: tag-value`
- `autoscaling:TargetGroupARNs`
- `autoscaling:VPCZoneIdentifiers`

Kunci kondisi berikut khusus untuk membuat permintaan konfigurasi peluncuran:

- `autoscaling:ImageId`
- `autoscaling:InstanceType`
- `autoscaling:MetadataHttpEndpoint`
- `autoscaling:MetadataHttpPutResponseHopLimit`
- `autoscaling:MetadataHttpTokens`
- `autoscaling:SpotPrice`

Auto Scaling Amazon EC2 juga mendukung kunci kondisi global berikut yang dapat Anda gunakan untuk menentukan izin berdasarkan tag dalam permintaan atau yang ada di grup Auto Scaling. Untuk informasi selengkapnya, lihat [Menandai grup dan instance Auto Scaling](#).

- `aws:RequestTag/key-name: tag-value`
- `aws:ResourceTag/key-name: tag-value`
- `aws:TagKeys: [tag-key, ...]`

Untuk mempelajari tindakan API Auto Scaling Amazon EC2 yang dapat digunakan untuk menggunakan kunci kondisi, [lihat Tindakan yang ditentukan oleh Amazon EC2 Auto Scaling](#) di Referensi Otorisasi Layanan. Untuk informasi selengkapnya tentang kunci kondisi Auto Scaling Amazon EC2, [lihat Kunci kondisi untuk Auto Scaling Amazon EC2](#).

Note

Untuk contoh kebijakan IAM yang menggunakan kunci kondisi untuk mengontrol akses ke tindakan yang didukung dan menerapkan konfigurasi grup Auto Scaling, lihat sumber daya berikut:

- [Memerlukan template peluncuran dan nomor versi](#)— Contoh ini memberlakukan bahwa template peluncuran dan nomor versi template peluncuran harus ditentukan saat membuat atau memperbarui grup Auto Scaling.
- [Kontrol ukuran grup Auto Scaling yang dapat dibuat](#)— Contoh ini memberlakukan batasan pada nilai yang mungkin untuk MaxSize properti MinSize dan saat membuat atau memperbarui grup Auto Scaling dengan tag tertentu.
- [Kontrol kebijakan penskalaan yang dapat dihapus](#)— Contoh ini memberlakukan bahwa menghapus kebijakan penskalaan hanya diperbolehkan untuk grup Auto Scaling tanpa tag tertentu.

ACL di Auto Scaling Amazon EC2

Mendukung ACL

Tidak

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

ABAC dengan Auto Scaling Amazon EC2

Mendukung ABAC (tanda dalam kebijakan)	Parsial
--	---------

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. Pemberian tanda ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian, rancanglah kebijakan ABAC untuk mengizinkan operasi saat tag milik pengguna utama cocok dengan tag yang ada di sumber daya yang ingin diakses.

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi di mana pengelolaan kebijakan menjadi rumit.

Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di [elemen kondisi](#) dari kebijakan dengan 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 hanya untuk beberapa jenis sumber daya, nilainya adalah Parsial.

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

ABAC dimungkinkan untuk sumber daya yang mendukung tag, tetapi tidak semuanya mendukung tag. Konfigurasi peluncuran dan kebijakan penskalaan tidak mendukung tag, tetapi grup Auto Scaling mendukung tag.

Untuk informasi selengkapnya, lihat [Menandai grup dan instance Auto Scaling](#).

Menggunakan kredensial sementara dengan Amazon EC2 Auto Scaling

Mendukung kredensial sementara	Ya
--------------------------------	----

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensial sementara. Untuk informasi tambahan, termasuk yang Layanan AWS bekerja dengan kredensial 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 kredensial sementara. Anda juga akan membuat kredensial sementara secara otomatis saat masuk ke konsol sebagai pengguna dan kemudian beralih peran. Untuk informasi selengkapnya tentang cara beralih peran, lihat [Beralih peran \(konsol\)](#) di 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](#).

Peran layanan untuk Amazon EC2 Auto Scaling

Mendukung peran layanan

Ya

Peran layanan adalah [peran IAM](#) yang diambil 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 peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

Saat membuat hook siklus hidup yang memberi tahu topik Amazon SNS atau antrean Amazon SQS, Anda harus menentukan peran untuk mengizinkan Penskalaan Otomatis Amazon EC2 mengakses Amazon SNS atau Amazon SQS atas nama Anda. Gunakan konsol IAM untuk mengatur peran layanan untuk hook siklus hidup Anda. Konsol membantu Anda membuat peran dengan sekumpulan izin yang memadai menggunakan kebijakan terkelola. Lihat informasi yang lebih lengkap di [Terima notifikasi menggunakan Amazon SNS](#) dan [Menerima pemberitahuan menggunakan Amazon SQS](#).

Saat membuat grup Auto Scaling, Anda dapat meneruskan peran layanan secara opsional untuk mengizinkan instans Amazon EC2 mengakses instans lain atas nama Anda. Layanan AWS Peran layanan untuk instans Amazon EC2 (juga disebut profil instans Amazon EC2 untuk templat peluncuran atau konfigurasi peluncuran) adalah jenis peran layanan khusus yang ditetapkan ke

setiap instans EC2 dalam grup Auto Scaling saat instans diluncurkan. Anda dapat menggunakan konsol IAM dan AWS CLI untuk membuat atau mengedit peran layanan ini. Untuk informasi selengkapnya, lihat [Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2](#).

Warning

Mengubah izin untuk peran layanan dapat merusak fungsionalitas Auto Scaling Amazon EC2. Edit peran layanan hanya jika Auto Scaling Amazon EC2 memberikan panduan untuk melakukannya.

Peran terkait layanan untuk Amazon EC2 Auto Scaling

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

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

Untuk informasi selengkapnya tentang cara membuat atau mengelola peran Amazon EC2 Auto Scaling terkait layanan, lihat [Peran terkait layanan untuk Amazon EC2 Auto Scaling](#).

Izin API Auto Scaling Amazon EC2

Anda harus memberikan izin kepada pengguna untuk memanggil tindakan API Auto Scaling Amazon EC2 yang mereka butuhkan, seperti yang dijelaskan dalam [Tindakan kebijakan untuk Auto Scaling Amazon EC2](#). Selain itu, untuk beberapa tindakan Auto Scaling Amazon EC2, Anda harus memberikan izin kepada pengguna untuk memanggil tindakan tertentu dari API lain. AWS

Izin yang diperlukan dari API lain AWS

Selain izin API Auto Scaling Amazon EC2, pengguna harus memiliki izin berikut dari API AWS lain agar berhasil melakukan tindakan terkait.

Membuat grup Auto Scaling () `autoscaling:CreateAutoScalingGroup`

- `iam:CreateServiceLinkedRole`— Untuk membuat peran terkait layanan default jika peran itu belum ada.

- `iam:PassRole`— Untuk meneruskan peran IAM ke layanan atau ke instans EC2 saat diluncurkan. Diperlukan saat peran terkait layanan nondefault, peran IAM untuk pengait siklus hidup, atau templat peluncuran yang menentukan profil instance (wadah untuk peran IAM) disediakan.
- `ec2:RunInstance`— Untuk meluncurkan instance saat template peluncuran disediakan.
- `ec2:CreateTags`— Untuk menandai instance dan volume saat peluncuran saat template peluncuran dengan spesifikasi tag disediakan.

Buat hook siklus hidup () `autoscaling:PutLifecycleHook`

- `iam:PassRole`— Untuk meneruskan peran IAM ke layanan. Diperlukan saat peran IAM disediakan.

Lampirkan grup target VPC Lattice () `autoscaling:AttachTrafficSources`

- `vpc-lattice:RegisterTargets`— Untuk secara otomatis mendaftarkan instance dengan grup target.

Lepaskan grup target VPC Lattice () `autoscaling:DetachTrafficSources`

- `vpc-lattice:DeregisterTargets`— Untuk secara otomatis membatalkan pendaftaran instance dengan grup target.

Buat konfigurasi peluncuran (`autoscaling>CreateLaunchConfiguration`)

- `ec2:DescribeImages`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceAttribute`
- `ec2:DescribeKeyPairs`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSpotInstanceRequests`
- `ec2:DescribeVpcClassicLink`
- `iam:PassRole`— Untuk meneruskan peran IAM ke instans EC2 saat diluncurkan. Diperlukan saat konfigurasi peluncuran menentukan profil instance (wadah untuk peran IAM).

AWS kebijakan terkelola untuk Auto Scaling Amazon EC2

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, pemutakhiran 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 [Kebijakan yang dikelola AWS](#) dalam Panduan Pengguna IAM.

Kebijakan terkelola Auto Scaling Amazon EC2

Anda dapat melampirkan kebijakan terkelola berikut ke identitas AWS Identity and Access Management (IAM) Anda (pengguna atau peran). Setiap kebijakan menyediakan akses ke semua atau beberapa tindakan API untuk Amazon EC2 Auto Scaling.

- [AutoScalingFullAccess](#)— Memberikan akses penuh ke Amazon EC2 Auto Scaling untuk identitas IAM yang memerlukan akses Auto Scaling Amazon EC2 penuh dari atau SDK, tetapi bukan akses. AWS CLI AWS Management Console
- [AutoScalingReadOnlyAccess](#)— Memberikan akses hanya-baca ke Amazon EC2 Auto Scaling untuk identitas IAM yang hanya melakukan panggilan ke atau SDK. AWS CLI
- [AutoScalingConsoleFullAccess](#)— Memberikan akses penuh ke Amazon EC2 Auto Scaling menggunakan. AWS Management Console Kebijakan ini berfungsi saat Anda menggunakan konfigurasi peluncuran, tetapi tidak saat Anda menggunakan templat peluncuran.
- [AutoScalingConsoleReadOnlyAccess](#)— Memberikan akses hanya-baca ke Amazon EC2 Auto Scaling menggunakan. AWS Management Console Kebijakan ini berfungsi saat Anda menggunakan konfigurasi peluncuran, tetapi tidak saat Anda menggunakan templat peluncuran.

Saat Anda menggunakan templat peluncuran dari konsol, Anda perlu memberikan izin tambahan khusus untuk meluncurkan templat, yang dibahas di [Luncurkan dukungan templat](#). Konsol Auto Scaling Amazon EC2 memerlukan izin `ec2` untuk tindakan sehingga dapat menampilkan informasi tentang templat peluncuran dan instance peluncuran menggunakan templat peluncuran.

Kebijakan terkelola AutoScalingServiceRolePolicy AWS

Anda tidak dapat [AutoScalingServiceRolePolicy](#) melampirkan identitas IAM Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang memungkinkan Auto Scaling Amazon EC2 meluncurkan dan menghentikan instans. Untuk informasi selengkapnya, lihat [Peran terkait layanan untuk Amazon EC2 Auto Scaling](#).

Pembaruan Auto Scaling Amazon EC2 ke kebijakan terkelola AWS

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Auto Scaling Amazon EC2 sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman riwayat Dokumen Auto Scaling Amazon EC2.

Perubahan	Deskripsi	Tanggal
Auto Scaling Amazon EC2 menambahkan izin ke peran terkait layanannya	AutoScalingServiceRolePolicy Kebijakan ini sekarang memberikan izin untuk memanggil tindakan Amazon GetSecurityGroupsForVpc EC2 API agar semua grup keamanan agar VPC dapat meningkatkan validasi, dan tindakan GetInstanceTypesFromInstanceRequirements Amazon EC2 API untuk mendapatkan informasi tentang jenis instans mana yang memenuhi serangkaian persyaratan instans tertentu. Untuk informasi selengkapnya, lihat Peran terkait layanan untuk Amazon EC2 Auto Scaling .	Februari 29, 2024

Perubahan	Deskripsi	Tanggal
<p>Auto Scaling Amazon EC2 menambahkan izin ke peran terkait layanannya</p>	<p>AutoScalingService RolePolicy Kebijakan ini sekarang memberikan izin ke layanan untuk mengakses tindakan API yang diperlukan untuk integrasi dengan VPC Lattice.</p> <ul style="list-style-type: none"> • <code>GetTargetGroup</code> dan <code>ListTargetGroup</code> tindakan. Diperlukan untuk mengambil informasi tentang kelompok target VPC Lattice. • <code>RegisterTargets</code> dan <code>DeregisterTargets</code> tindakan. Diperlukan untuk mendaftarkan dan membatalkan pendaftaran instance dari grup target VPC Lattice. • <code>ListTargets</code> . Memungkinkan Auto Scaling Amazon EC2 untuk mengambil informasi kesehatan untuk instans yang terdaftar ke grup target VPC Lattice. <p>Untuk informasi selengkapnya, lihat Peran terkait layanan untuk Amazon EC2 Auto Scaling.</p>	<p>6 Desember 2022</p>

Perubahan	Deskripsi	Tanggal
Auto Scaling Amazon EC2 menambahkan izin ke peran terkait layanannya	Untuk mendukung pengguna n AWS Systems Manager Parameter sebagai alias untuk ID AMI saat membuat template peluncuran, <code>AutoScalingServiceRolePolicy</code> kebijakan sekarang memberikan izin untuk memanggil tindakan AWS Systems Manager GetParametersAPI . Untuk informasi selengkapnya, lihat Peran terkait layanan untuk Amazon EC2 Auto Scaling .	28 Maret 2022
Auto Scaling Amazon EC2 menambahkan izin ke peran terkait layanannya	Untuk mendukung penskalaan prediktif, <code>AutoScalingServiceRolePolicy</code> kebijakan sekarang menyertakan izin untuk memanggil tindakan CloudWatch GetMetricDataAPI . Untuk informasi selengkapnya, lihat Peran terkait layanan untuk Amazon EC2 Auto Scaling .	19 Mei 2021
Auto Scaling Amazon EC2 mulai melacak perubahan	Auto Scaling Amazon EC2 mulai melacak perubahan untuk AWS kebijakan yang dikelola.	19 Mei 2021

Peran terkait layanan untuk Amazon EC2 Auto Scaling

Auto Scaling Amazon EC2 menggunakan peran terkait layanan untuk izin yang diperlukan untuk memanggil orang lain atas nama Anda. Layanan AWS Peran terkait layanan adalah jenis unik peran IAM yang ditautkan langsung ke sebuah. Layanan AWS

Peran terkait layanan menyediakan cara aman untuk mendelegasikan izin ke pihak lain Layanan AWS karena hanya layanan tertaut yang dapat mengambil peran terkait layanan. Untuk informasi selengkapnya, lihat [Menggunakan peran terkait layanan](#) dalam Panduan Pengguna IAM. Peran terkait layanan juga memungkinkan semua panggilan API terlihat. AWS CloudTrail Hal ini membantu dengan persyaratan pemantauan dan audit karena Anda dapat melacak semua tindakan yang dilakukan Auto Scaling Amazon EC2 atas nama Anda. Untuk informasi selengkapnya, lihat [Log panggilan API Auto Scaling Amazon EC2 dengan AWS CloudTrail](#).

Bagian berikut ini menjelaskan cara membuat dan mengelola peran otomatis Amazon EC2 Auto Scaling. Mulailah dengan mengonfigurasi izin untuk mengizinkan identitas IAM (seperti pengguna atau peran) membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat [Menggunakan peran terkait layanan](#) dalam Panduan Pengguna IAM.

Daftar Isi

- [Gambaran Umum](#)
- [Izin yang diberikan oleh peran tertaut layanan](#)
- [Buat peran terkait layanan \(otomatis\)](#)
- [Buat peran terkait layanan \(manual\)](#)
- [Edit peran terkait layanan](#)
- [Hapus peran terkait layanan](#)
- [Wilayah yang Didukung untuk peran terkait layanan Amazon EC2 Auto Scaling](#)

Gambaran Umum

Ada dua jenis peran terkait layanan Amazon EC2 Auto Scaling:

- Peran default yang ditautkan layanan untuk akun Anda, bernama. `AWSServiceRoleForAutoScaling` Peran ini secara otomatis ditetapkan ke grup Auto Scaling kecuali Anda menentukan peran terkait layanan yang berbeda.
- ***Peran terkait layanan dengan akhiran khusus yang Anda tentukan saat membuat peran, misalnya, `AWSServiceRoleForAutoScaling _ mysuffix`.***

Izin peran terkait layanan akhiran kustom identik dengan peran terhubung ke layanan default. Dalam kedua kasus, Anda tidak dapat mengedit peran, dan Anda juga tidak dapat menghapusnya jika peran tersebut masih digunakan oleh grup Auto Scaling. Satu-satunya perbedaan adalah akhiran nama peran.

Anda dapat menentukan salah satu peran saat mengedit kebijakan AWS Key Management Service utama agar instans yang diluncurkan oleh Amazon EC2 Auto Scaling dienkripsi dengan kunci yang dikelola pelanggan. Namun, jika Anda berencana untuk memberikan akses terperinci ke kunci terkelola pelanggan tertentu, Anda harus menggunakan peran terkait layanan akhiran khusus. Dengan peran layanan terkait akhiran yang disesuaikan, Anda akan mendapatkan:

- Kontrol lebih besar atas kunci yang dikelola pelanggan
- Kemampuan untuk melacak grup Auto Scaling mana yang melakukan panggilan API di log Anda CloudTrail

Jika Anda membuat kunci terkelola pelanggan yang tidak dapat diakses oleh semua pengguna, ikuti langkah-langkah berikut untuk mengizinkan penggunaan peran terkait layanan akhiran kustom:

1. Buat peran terkait layanan dengan akhiran khusus. Untuk informasi selengkapnya, lihat [Buat peran terkait layanan \(manual\)](#).
2. Berikan akses peran terkait layanan ke kunci yang dikelola pelanggan. Untuk informasi selengkapnya tentang kebijakan kunci yang memungkinkan kunci digunakan oleh peran terkait layanan, lihat [Kebijakan AWS KMS kunci yang diperlukan untuk digunakan dengan volume terenkripsi](#)
3. Berikan pengguna akses ke peran terkait layanan yang Anda buat. Untuk informasi selengkapnya tentang cara membuat kebijakan IAM, lihat [Kontrol peran terkait layanan mana yang dapat diteruskan \(menggunakan\) PassRole](#). Jika pengguna mencoba menentukan peran terkait layanan tanpa izin untuk meneruskan peran tersebut ke layanan, mereka akan menerima pesan kesalahan.

Izin yang diberikan oleh peran tertaut layanan

Auto Scaling Amazon EC2 menggunakan `AWSServiceRoleForAutoScaling` nama peran terkait layanan atau peran terkait layanan akhiran khusus Anda.

Peran terkait layanan memercayai layanan berikut untuk mengambil peran tersebut:

- `autoscaling.amazonaws.com`

Peran menggunakan [AutoScalingServiceRolePolicy](#) kebijakan, yang mencakup izin berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2InstanceManagement",
      "Effect": "Allow",
      "Action": [
        "ec2:AttachClassicLinkVpc",
        "ec2:CancelSpotInstanceRequests",
        "ec2:CreateFleet",
        "ec2:CreateTags",
        "ec2>DeleteTags",
        "ec2:Describe*",
        "ec2:DetachClassicLinkVpc",
        "ec2:GetInstanceTypesFromInstanceRequirements",
        "ec2:GetSecurityGroupsForVpc",
        "ec2:ModifyInstanceAttribute",
        "ec2:RequestSpotInstances",
        "ec2:RunInstances",
        "ec2:StartInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": "*"
    },
    {
      "Sid": "EC2InstanceProfileManagement",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "ec2.amazonaws.com*"
        }
      }
    },
    {
      "Sid": "EC2SpotManagement",
      "Effect": "Allow",
      "Action": [
```

```

    "iam:CreateServiceLinkedRole"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "iam:AWSServiceName": "spot.amazonaws.com"
    }
  }
},
{
  "Sid": "ELBManagement",
  "Effect": "Allow",
  "Action": [
    "elasticloadbalancing:Register*",
    "elasticloadbalancing:Deregister*",
    "elasticloadbalancing:Describe*"
  ],
  "Resource": "*"
},
{
  "Sid": "CWManagement",
  "Effect": "Allow",
  "Action": [
    "cloudwatch:DeleteAlarms",
    "cloudwatch:DescribeAlarms",
    "cloudwatch:GetMetricData",
    "cloudwatch:PutMetricAlarm"
  ],
  "Resource": "*"
},
{
  "Sid": "SNSManagement",
  "Effect": "Allow",
  "Action": [
    "sns:Publish"
  ],
  "Resource": "*"
},
{
  "Sid": "EventBridgeRuleManagement",
  "Effect": "Allow",
  "Action": [
    "events:PutRule",
    "events:PutTargets",

```



```

    "events:RemoveTargets",
    "events>DeleteRule",
    "events:DescribeRule"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "events:ManagedBy": "autoscaling.amazonaws.com"
    }
  }
},
{
  "Sid": "SystemsManagerParameterManagement",
  "Effect": "Allow",
  "Action": [
    "ssm:GetParameters"
  ],
  "Resource": "*"
},
{
  "Sid": "VpcLatticeManagement",
  "Effect": "Allow",
  "Action": [
    "vpc-lattice:DeregisterTargets",
    "vpc-lattice:GetTargetGroup",
    "vpc-lattice:ListTargets",
    "vpc-lattice:ListTargetGroups",
    "vpc-lattice:RegisterTargets"
  ],
  "Resource": "*"
}
]
}

```

Peran memiliki izin untuk melakukan semua hal berikut:

- `ec2`— Buat, jelaskan, modifikasi, mulai/hentikan, dan akhiri instans EC2.
- `iam`— [Lulus peran IAM](#) ke instans EC2 sehingga aplikasi yang berjalan pada instans dapat mengakses kredensial sementara untuk peran tersebut.
- `iam`— Buat peran `AWSServiceRoleForEC2Spot` terkait layanan untuk memungkinkan Auto Scaling Amazon EC2 meluncurkan Instans Spot atas nama Anda.

- `elasticloadbalancing`— Daftarkan dan deregister instans dengan Elastic Load Balancing dan periksa kesehatan target yang terdaftar.
- `cloudwatch`— Membuat, mendeskripsikan, memodifikasi, dan menghapus CloudWatch alarm untuk kebijakan penskalaan dan mengambil metrik yang digunakan untuk penskalaan prediktif.
- `sns`— Publikasikan pemberitahuan ke Amazon SNS saat instance diluncurkan atau dihentikan.
- `events`— Buat, jelaskan, perbarui, dan hapus EventBridge aturan atas nama Anda.
- `ssm`— Baca parameter dari Parameter Store saat menggunakan parameter Systems Manager sebagai alias untuk ID AMI dalam template peluncuran.
- `vpc-lattice`— Daftarkan dan deregister instance dengan VPC Lattice dan periksa kesehatan target yang terdaftar.

Buat peran terkait layanan (otomatis)

Auto Scaling Amazon EC2 membuat `AWSServiceRoleForAutoScaling` peran terkait layanan untuk Anda saat pertama kali membuat grup Auto Scaling, kecuali jika Anda membuat peran terkait layanan akhiran khusus secara manual dan menentukannya saat membuat grup.

Important

Anda harus memiliki izin IAM untuk membuat peran terkait layanan. Jika tidak, pembuatan otomatis gagal. Untuk informasi lebih lanjut, lihat [Izin peran yang ditautkan dengan layanan](#) dalam Panduan Pengguna IAM dan [Buat peran tertaut layanan](#) dalam panduan ini.

Amazon EC2 Auto Scaling mulai mendukung peran terkait layanan pada Maret 2018. Jika Anda membuat grup Auto Scaling sebelum itu, Amazon EC2 Auto Scaling akan `AWSServiceRoleForAutoScaling` membuat peran tersebut di akun Anda. Untuk informasi selengkapnya, lihat [Peran baru muncul di saya Akun AWS](#) di Panduan Pengguna IAM.

Buat peran terkait layanan (manual)

Untuk membuat peran terkait layanan (konsol)

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Dalam panel navigasi, pilih Roles (Peran), lalu Create role (Buat peran).
3. Untuk Pilih entitas tepercaya, pilih AWS layanan.

4. Untuk Pilih layanan yang akan menggunakan peran ini, pilih EC2 Auto Scaling dan kasus penggunaan EC2 Auto Scaling.
5. Pilih Berikutnya: Izin, Berikutnya: Tag, dan kemudian Berikutnya: Ulasan. Catatan: Anda tidak dapat melampirkan tag ke peran terkait layanan selama pembuatan.
6. ***Pada halaman Tinjauan, biarkan nama Peran kosong untuk membuat peran terkait layanan dengan nama `AWSServiceRoleForAutoScaling`, atau masukkan akhiran untuk membuat peran terkait layanan dengan akhiran nama `_
AWSServiceRoleForAutoScaling`***
7. (Opsional) Untuk Deskripsi peran, edit deskripsi untuk peran terkait layanan.
8. Pilih Buat peran.

Untuk membuat peran terkait layanan (AWS CLI)

***Gunakan perintah `create-service-linked-role` CLI berikut untuk membuat peran terkait layanan untuk Auto Scaling Amazon EC2 dengan akhiran nama `_
AWSServiceRoleForAutoScaling`***

```
aws iam create-service-linked-role --aws-service-name autoscaling.amazonaws.com --  
custom-suffix suffix
```

Output dari perintah ini mencakup ARN dari peran terkait layanan, yang dapat Anda gunakan untuk memberikan akses peran terkait layanan ke kunci terkelola pelanggan Anda.

```
{  
  "Role": {  
    "RoleId": "ABCDEF0123456789ABCDEF",  
    "CreateDate": "2018-08-30T21:59:18Z",  
    "RoleName": "AWSServiceRoleForAutoScaling_suffix",  
    "Arn": "arn:aws:iam::123456789012:role/aws-service-role/  
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling_suffix",  
    "Path": "/aws-service-role/autoscaling.amazonaws.com/",  
    "AssumeRolePolicyDocument": {  
      "Version": "2012-10-17",  
      "Statement": [  
        {  
          "Action": [  
            "sts:AssumeRole"  
          ],  
        }  
      ],  
    }  
  }  
}
```

```
        "Principal": {
            "Service": [
                "autoscaling.amazonaws.com"
            ]
        },
        "Effect": "Allow"
    }
}
}
```

Untuk informasi lebih lanjut, lihat [Membuat peran terkait layanan](#) dalam Panduan Pengguna IAM.

Edit peran terkait layanan

Anda tidak dapat mengedit peran terhubung layanan yang dibuat untuk Amazon EC2 Auto Scaling. Setelah Anda membuat peran terkait layanan, Anda tidak dapat mengubah nama peran atau izinnya. Namun, Anda dapat mengedit deskripsi peran. Untuk informasi lebih lanjut, lihat [Mengedit peran terkait layanan](#) dalam Panduan Pengguna IAM.

Hapus peran terkait layanan

Jika Anda tidak menggunakan grup Auto Scaling, sebaiknya hapus peran terkait layanan. Menghapus peran mencegah Anda memiliki entitas yang tidak digunakan atau dipantau dan dipelihara secara aktif.

Anda dapat menghapus peran terkait layanan hanya setelah terlebih dahulu menghapus sumber daya dependen terkait. Hal ini melindungi Anda dari secara tidak sengaja mencabut izin Amazon EC2 Auto Scaling ke sumber daya Anda. Jika peran terkait layanan digunakan dengan beberapa grup Auto Scaling, Anda harus menghapus semua grup Auto Scaling yang menggunakan peran terkait layanan sebelum Anda dapat menghapusnya. Untuk informasi selengkapnya, lihat [Hapus infrastruktur Auto Scaling Anda](#).

Anda dapat menggunakan IAM untuk menghapus peran yang terkait layanan. Untuk informasi lebih lanjut, lihat [Menghapus peran terkait layanan](#) dalam Panduan Pengguna IAM.

Jika Anda menghapus peran `AWSServiceRoleForAutoScaling` terkait layanan, Auto Scaling Amazon EC2 akan membuat peran tersebut lagi saat Anda membuat grup Auto Scaling dan tidak menentukan peran terkait layanan lainnya.

Wilayah yang Didukung untuk peran terkait layanan Amazon EC2 Auto Scaling

Auto Scaling Amazon EC2 mendukung penggunaan peran terkait layanan di semua tempat layanan tersedia Wilayah AWS .

Contoh kebijakan Amazon EC2 Auto Scaling berbasis identitas

Secara default, pengguna baru di Anda tidak Akun AWS memiliki izin untuk melakukan apa pun. Administrator IAM harus membuat dan menetapkan kebijakan IAM yang memberikan izin identitas IAM (seperti pengguna atau peran) untuk melakukan tindakan API Auto Scaling Amazon EC2.

Untuk mempelajari cara membuat kebijakan IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan di tab JSON](#) dalam Panduan Pengguna IAM.

Berikut adalah contoh kebijakan izin.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "autoscaling:CreateAutoScalingGroup",
      "autoscaling:UpdateAutoScalingGroup",
      "autoscaling>DeleteAutoScalingGroup"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": { "autoscaling:ResourceTag/purpose": "testing" }
    }
  },
  {
    "Effect": "Allow",
    "Action": "autoscaling:Describe*",
    "Resource": "*"
  }
]
```

Kebijakan contoh ini memberikan izin untuk membuat, memperbarui, dan menghapus grup Auto Scaling, tetapi hanya jika grup menggunakan tag. **purpose=testing** Karena Describe tindakan tidak mendukung izin tingkat sumber daya, Anda harus menentukannya dalam pernyataan terpisah tanpa kondisi. Untuk meluncurkan instance dengan template peluncuran, pengguna juga harus

memiliki `ec2:RunInstances` izin. Untuk informasi selengkapnya, lihat [Luncurkan dukungan templat](#).

Note

Anda dapat membuat kebijakan IAM kustom Anda sendiri untuk mengizinkan atau menolak izin identitas IAM (pengguna atau peran) untuk melakukan tindakan Auto Scaling Amazon EC2. Anda dapat melampirkan kebijakan kustom ini ke identitas IAM yang memerlukan izin yang ditentukan. Contoh berikut menunjukkan izin untuk beberapa kasus penggunaan umum. Beberapa tindakan Amazon EC2 Auto Scaling API memungkinkan Anda untuk menyertakan grup Auto Scaling tertentu dalam kebijakan Anda yang dapat dibuat atau dimodifikasi oleh tindakan tersebut. Anda dapat membatasi sumber daya target untuk tindakan ini dengan menentukan masing-masing ARN grup Auto Scaling. Namun, sebagai praktik terbaik, kami menyarankan Anda menggunakan kebijakan berbasis tag yang mengizinkan (atau menolak) tindakan pada grup Auto Scaling dengan tag tertentu.

Topik

- [Kontrol ukuran grup Auto Scaling yang dapat dibuat](#)
- [Kontrol kunci tag dan nilai tag mana yang dapat digunakan](#)
- [Kontrol grup Auto Scaling mana yang dapat dihapus](#)
- [Kontrol kebijakan penskalaan yang dapat dihapus](#)
- [Kontrol akses ke tindakan penyegaran instans](#)
- [Buat peran tertaut layanan](#)
- [Kontrol peran terkait layanan mana yang dapat diteruskan \(menggunakan\) PassRole](#)

Kontrol ukuran grup Auto Scaling yang dapat dibuat

Kebijakan berikut memberikan izin untuk membuat dan memperbarui semua grup Auto Scaling dengan **`environment=development`** tag, selama pemohon tidak menentukan ukuran minimum kurang **1** dari atau ukuran maksimum yang lebih besar dari **10**. Jika memungkinkan, gunakan tag untuk membantu Anda mengontrol akses ke grup Auto Scaling di akun Anda.

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

```

    "Effect": "Allow",
    "Action": [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": { "autoscaling:ResourceTag/environment": "development" },
        "NumericGreaterThanEqualsIfExists": { "autoscaling:MinSize": 1 },
        "NumericLessThanEqualsIfExists": { "autoscaling:MaxSize": 10 }
    }
}

```

Atau, jika Anda tidak menggunakan tag untuk mengontrol akses ke grup Auto Scaling, Anda dapat menggunakan ARN untuk mengidentifikasi grup Auto Scaling yang berlaku untuk kebijakan IAM.

Grup Auto Scaling memiliki ARN berikut.

```

"Resource": "arn:aws:autoscaling:region:account-id:autoScalingGroup:*:autoScalingGroupName/my-asg"

```

Anda juga dapat menentukan beberapa ARN dengan melampirkannya dalam daftar. Untuk informasi selengkapnya tentang menentukan ARN sumber daya Resource Auto Scaling Amazon EC2 di elemen, lihat [Sumber daya kebijakan untuk Auto Scaling Amazon EC2](#)

Kontrol kunci tag dan nilai tag mana yang dapat digunakan

Anda juga dapat menggunakan kondisi dalam kebijakan IAM untuk mengontrol kunci tag dan nilai tag yang dapat diterapkan ke grup Auto Scaling. Untuk memberikan izin untuk membuat atau menandai grup Auto Scaling hanya jika pemohon menentukan tag tertentu, gunakan kunci kondisi `aws:RequestTag` Agar hanya memperbolehkan kunci tag tertentu, gunakan kunci ketentuan `aws:TagKeys` dengan modifier `ForAllValues`.

Kebijakan berikut mengharuskan pemohon untuk menentukan tag dengan kunci **environment** dalam permintaan. Nilai `"?*"` akan mendorong adanya beberapa nilai untuk kunci tag. Untuk menggunakan wildcard, Anda harus menggunakan operator ketentuan `StringLike`.

```

{
  "Version": "2012-10-17",
  "Statement": [{

```

```

    "Effect": "Allow",
    "Action": [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:CreateOrUpdateTags"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": { "aws:RequestTag/environment": "?*" }
    }
}
]]
}

```

Kebijakan berikut menetapkan bahwa pemohon hanya dapat menandai grup Auto Scaling dengan tag **cost-center=cc123** dan, dan hanya **purpose** mengizinkan **purpose=webserver** tag dan (tidak ada tag lain yang dapat ditentukan). **cost-center**

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:CreateOrUpdateTags"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/purpose": "webserver",
            "aws:RequestTag/cost-center": "cc123"
        },
        "ForAllValues:StringEquals": { "aws:TagKeys": [purpose, cost-center] }
    }
  }]
}

```

Kebijakan berikut mengharuskan pemohon untuk menentukan setidaknya satu tag dalam permintaan, dan hanya mengizinkan **cost-center** dan **owner** kunci.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",

```



```

    "Action": [
      "autoscaling:CreateAutoScalingGroup",
      "autoscaling:CreateOrUpdateTags"
    ],
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": { "aws:TagKeys": ["cost-center", "owner"] }
    }
  }
}

```

Note

Untuk syarat, kunci syarat tidak bersifat peka terhadap huruf besar dan kecil dan nilai syarat bersifat peka huruf besar dan kecil. Oleh karena itu, untuk memaksakan sifat peka terhadap huruf besar atau kecil dari kunci tanda, gunakan kunci syarat `aws:TagKeys`, di mana kunci tanda ditetapkan sebagai nilai dalam syarat tersebut.

Kontrol grup Auto Scaling mana yang dapat dihapus

Kebijakan berikut memungkinkan penghapusan grup Auto Scaling hanya jika grup memiliki tag **environment=development**

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "autoscaling:DeleteAutoScalingGroup",
    "Resource": "*",
    "Condition": {
      "StringEquals": { "aws:ResourceTag/environment": "development" }
    }
  }]
}

```

Atau, jika Anda tidak menggunakan tombol kondisi untuk mengontrol akses ke grup Auto Scaling, Anda dapat menentukan ARN sumber daya dalam `Resource` elemen untuk mengontrol akses.

Kebijakan berikut memberikan izin kepada pengguna untuk menggunakan tindakan `DeleteAutoScalingGroup` API, tetapi hanya untuk grup Auto Scaling yang namanya dimulai **devteam-**

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "autoscaling:DeleteAutoScalingGroup",
    "Resource": "arn:aws:autoscaling:region:account-
id:autoScalingGroup:*:autoScalingGroupName/devteam-*"
  }]
}
```

Anda juga dapat menentukan beberapa ARN dengan melampirkannya dalam daftar. Menyertakan UUID memastikan bahwa akses diberikan ke grup Auto Scaling tertentu. UUID untuk grup baru berbeda dari UUID untuk grup yang dihapus dengan nama yang sama.

```
"Resource": [
  "arn:aws:autoscaling:region:account-
id:autoScalingGroup:uuid:autoScalingGroupName/devteam-1",
  "arn:aws:autoscaling:region:account-
id:autoScalingGroup:uuid:autoScalingGroupName/devteam-2",
  "arn:aws:autoscaling:region:account-
id:autoScalingGroup:uuid:autoScalingGroupName/devteam-3"
]
```

Kontrol kebijakan penskalaan yang dapat dihapus

Kebijakan berikut memberikan izin untuk menggunakan `DeletePolicy` tindakan tersebut guna menghapus kebijakan penskalaan. Namun, ini juga menggantikan tindakan jika grup Auto Scaling yang ditindaklanjuti memiliki tanda **environment=production**. Jika memungkinkan, gunakan tag untuk membantu Anda mengontrol akses ke grup Auto Scaling di akun Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "autoscaling:DeletePolicy",
    "Resource": "*"
  }]
}
```

```

},
{
  "Effect": "Deny",
  "Action": "autoscaling:DeletePolicy",
  "Resource": "*",
  "Condition": {
    "StringEquals": { "autoscaling:ResourceTag/environment": "production" }
  }
}
}

```

Kontrol akses ke tindakan penyegaran instans

Kebijakan berikut memberikan izin untuk memulai, memutar kembali, dan membatalkan penyegaran instans hanya jika grup Auto Scaling yang ditindaklanjuti memiliki tag. **environment=testing** Karena Describe tindakan tidak mendukung izin tingkat sumber daya, Anda harus menentukannya dalam pernyataan terpisah tanpa kondisi.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "autoscaling:StartInstanceRefresh",
      "autoscaling:CancelInstanceRefresh",
      "autoscaling:RollbackInstanceRefresh"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": { "autoscaling:ResourceTag/environment": "testing" }
    }
  },
  {
    "Effect": "Allow",
    "Action": "autoscaling:DescribeInstanceRefreshes",
    "Resource": "*"
  }
}
}

```

Untuk menentukan konfigurasi yang diinginkan dalam StartInstanceRefresh panggilan, pengguna mungkin memerlukan beberapa izin terkait, seperti:

- `ec2:RunInstances` — Untuk meluncurkan instans EC2 menggunakan template peluncuran, pengguna harus memiliki `ec2:RunInstances` izin dalam kebijakan IAM. Untuk informasi selengkapnya, lihat [Luncurkan dukungan templat](#).
- `ec2:CreateTags` — Untuk meluncurkan instans EC2 dari template peluncuran yang menambahkan tag ke instance dan volume pada pembuatan, pengguna harus memiliki `ec2:CreateTags` izin dalam kebijakan IAM. Untuk informasi selengkapnya, lihat [Izin yang diperlukan untuk menandai instance dan volume](#).
- `iam:PassRole` — Untuk meluncurkan instans EC2 dari template peluncuran yang berisi profil instance (wadah untuk peran IAM), pengguna juga harus memiliki `iam:PassRole` izin dalam kebijakan IAM. Untuk informasi selengkapnya dan contoh kebijakan IAM, lihat [Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2](#).
- `ssm:GetParameters` — Untuk meluncurkan instans EC2 dari template peluncuran yang menggunakan AWS Systems Manager parameter, pengguna juga harus memiliki `ssm:GetParameters` izin dalam kebijakan IAM. Untuk informasi selengkapnya, lihat [Gunakan AWS Systems Manager parameter alih-alih ID AMI di templat peluncuran](#).

Buat peran tertaut layanan

Auto Scaling Amazon EC2 memerlukan izin untuk membuat peran terkait layanan saat pertama kali pengguna mana pun dalam panggilan Anda melakukan tindakan API Auto Scaling Akun AWS Amazon EC2. Jika peran terhubung layanan tidak ada, Amazon EC2 Auto Scaling akan membuatnya di akun Anda. Peran terkait layanan memberikan izin ke Auto Scaling Amazon EC2 sehingga dapat memanggil orang lain atas nama Anda. Layanan AWS

Agar pembuatan peran otomatis berhasil, pengguna harus memiliki izin untuk tindakan `iam:CreateServiceLinkedRole` nyata.

```
"Action": "iam:CreateServiceLinkedRole"
```

Berikut ini adalah contoh kebijakan izin yang memungkinkan pengguna untuk membuat peran Amazon EC2 Auto Scaling service-linked untuk Amazon EC2 Auto Scaling.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
```

```

    "Resource": "arn:aws:iam::*:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling",
    "Condition": {
        "StringLike": { "iam:AWSServiceName": "autoscaling.amazonaws.com" }
    }
}]]
}

```

Kontrol peran terkait layanan mana yang dapat diteruskan (menggunakan) PassRole

Pengguna yang membuat atau memperbarui grup Auto Scaling dan menentukan peran terkait layanan akhiran kustom dalam permintaan memerlukan izin. `iam:PassRole`

Anda dapat menggunakan `iam:PassRole` izin untuk melindungi keamanan kunci yang dikelola AWS KMS pelanggan jika Anda memberikan akses peran terkait layanan yang berbeda ke kunci yang berbeda. Tergantung pada kebutuhan organisasi Anda, Anda mungkin memiliki kunci untuk tim pengembangan, satu lagi untuk tim QA, dan satu lagi untuk tim keuangan. Pertama, buat peran terkait layanan yang memiliki akses ke kunci yang diperlukan, misalnya, peran terkait layanan bernama `AWSServiceRoleForAutoScaling_devteamkeyaccess`. Kemudian, lampirkan kebijakan ke identitas IAM, seperti pengguna atau peran.

Kebijakan berikut memberikan izin untuk meneruskan

`AWSServiceRoleForAutoScaling_devteamkeyaccess` peran ke grup Auto Scaling mana pun yang namanya dimulai. **`devteam-`** Jika identitas IAM yang membuat grup Auto Scaling mencoba menentukan peran terkait layanan yang berbeda, mereka akan menerima kesalahan. Jika mereka memilih untuk tidak menentukan peran terkait layanan, `AWSServiceRoleForAutoScaling` peran default akan digunakan sebagai gantinya.

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "arn:aws:iam::account-id:role/aws-service-role/
autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling_devteamkeyaccess",
    "Condition": {
      "StringEquals": { "iam:PassedToService": [ "autoscaling.amazonaws.com" ] },
      "StringLike": { "iam:AssociatedResourceARN":
[ "arn:aws:autoscaling:region:account-
id:autoScalingGroup:*:autoScalingGroupName/devteam-*" ] }
    }
  ]
}

```

```
}]  
}
```

Untuk informasi lebih lanjut tentang peran layanan terkait akhiran yang disesuaikan, lihat [Peran terkait layanan untuk Amazon EC2 Auto Scaling](#).

Pencegahan confused deputy lintas layanan

Masalah deputy yang bingung adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memaksa entitas yang lebih istimewa untuk melakukan tindakan.

Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses.

Untuk mencegah hal ini, AWS sediakan alat yang membantu Anda melindungi data Anda untuk semua layanan dengan prinsip layanan yang telah diberikan akses ke sumber daya di akun Anda. Sebaiknya gunakan kunci konteks kondisi [aws:SourceAccount](#) global [aws:SourceArn](#) dan global dalam kebijakan kepercayaan untuk peran layanan Auto Scaling Amazon EC2. Kunci ini membatasi izin yang diberikan oleh Amazon EC2 Auto Scaling layanan lain ke sumber daya.

Nilai untuk `SourceAccount` bidang `SourceArn` dan ditetapkan saat Auto Scaling Amazon EC2 AWS Security Token Service menggunakan `AWS STS()` untuk mengambil peran atas nama Anda.

Untuk menggunakan `aws:SourceArn` atau kunci kondisi `aws:SourceAccount` global, tetapkan nilainya ke Nama Sumber Daya Amazon (ARN) atau akun sumber daya yang disimpan oleh Amazon EC2 Auto Scaling. Bila memungkinkan, gunakan `aws:SourceArn`, yang lebih spesifik. Tetapkan nilai ke ARN atau pola ARN dengan wildcard (*) untuk bagian ARN yang tidak diketahui. Jika Anda tidak tahu ARN sumber daya, gunakan `aws:SourceAccount` sebagai gantinya.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan global di Auto Scaling Amazon EC2 untuk mencegah masalah deputy yang membingungkan.

Contoh: Menggunakan `aws:SourceArn` dan `aws:SourceAccount` mengkondisikan kunci

Peran yang diasumsikan layanan untuk melakukan tindakan atas nama Anda disebut [peran layanan](#). Jika Anda ingin membuat kait siklus hidup yang mengirim notifikasi ke tempat lain selain Amazon EventBridge, Anda harus membuat peran layanan untuk mengizinkan Auto Scaling Amazon EC2 mengirim notifikasi ke topik Amazon SNS atau antrean Amazon SQS atas nama Anda. Jika hanya ingin satu grup Auto Scaling dikaitkan dengan akses lintas layanan, Anda dapat menentukan kebijakan kepercayaan peran layanan sebagai berikut.

Contoh kebijakan trust ini menggunakan pernyataan kondisi untuk membatasi AssumeRole kemampuan pada peran layanan hanya pada tindakan yang memengaruhi grup Auto Scaling yang ditentukan di akun yang ditentukan. `aws:SourceAccountKondisi` `aws:SourceArn` dan dievaluasi secara independen. Setiap permintaan untuk menggunakan peran layanan harus memenuhi kedua kondisi tersebut.

Sebelum menggunakan kebijakan ini, ganti Region, ID akun, UUID, dan nama grup dengan nilai yang valid dari akun Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ConfusedDeputyPreventionExamplePolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "autoscaling.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn":
            "arn:aws:autoscaling:region:account_id:autoScalingGroup:uuid:autoScalingGroupName/my-
asg"
        },
        "StringEquals": {
          "aws:SourceAccount": "account_id"
        }
      }
    }
  ]
}
```

Dalam contoh sebelumnya:

- `PrincipalElement` menentukan prinsip layanan dari layanan (`autoscaling.amazonaws.com`).
- `ActionElement` menentukan `sts:AssumeRole` tindakan.
- `ConditionElement` menentukan `aws:SourceArn` dan kunci kondisi `aws:SourceAccount` global. ARN sumber mencakup ID akun sehingga tidak perlu menggunakan `aws:SourceAccount` dengan `aws:SourceArn`.

Informasi tambahan

Untuk informasi selengkapnya, lihat [kunci konteks kondisi AWS global](#), [Masalah deputi yang membingungkan](#), dan [Memodifikasi kebijakan kepercayaan peran \(konsol\)](#) di Panduan Pengguna IAM.

Luncurkan dukungan templat

Amazon EC2 Auto Scaling mendukung penggunaan templat peluncuran Amazon EC2 dengan grup Auto Scaling Anda. Kami menyarankan Anda mengizinkan pengguna untuk membuat grup Auto Scaling dari template peluncuran, karena hal itu memungkinkan mereka untuk menggunakan fitur terbaru Amazon EC2 Auto Scaling dan Amazon EC2. Misalnya, pengguna harus menentukan template peluncuran untuk menggunakan [kebijakan instance campuran](#).

Anda dapat menggunakan `AmazonEC2FullAccess` kebijakan ini untuk memberi pengguna akses lengkap agar dapat bekerja dengan sumber daya Penskalaan Otomatis Amazon EC2, templat peluncuran, dan sumber daya EC2 lainnya di akun mereka. Atau, Anda dapat membuat kebijakan IAM kustom Anda sendiri untuk memberi pengguna izin halus untuk bekerja dengan templat peluncuran, seperti yang dijelaskan dalam topik ini.

Contoh kebijakan yang dapat Anda sesuaikan untuk penggunaan Anda sendiri

Berikut ini menunjukkan contoh kebijakan izin dasar yang dapat Anda sesuaikan untuk digunakan sendiri. Kebijakan memberikan izin untuk membuat, memperbarui, dan menghapus semua grup Auto Scaling, tetapi hanya jika grup menggunakan tag. **purpose=testing** Kemudian memberikan izin untuk semua `Describe` tindakan. Karena `Describe` tindakan tidak mendukung izin tingkat sumber daya, Anda harus menentukannya dalam pernyataan terpisah tanpa kondisi.

Identitas IAM (pengguna atau peran) dengan kebijakan ini memiliki izin untuk membuat atau memperbarui grup Auto Scaling menggunakan templat peluncuran karena mereka juga diberi izin untuk menggunakan tindakan. `ec2:RunInstances`


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup",
        "autoscaling>DeleteAutoScalingGroup"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": { "autoscaling:ResourceTag/purpose": "testing" }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:Describe*",
        "ec2:RunInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

Pengguna yang membuat atau memperbarui grup Auto Scaling mungkin memerlukan beberapa izin terkait, seperti:

- `ec2: CreateTags` — Untuk menambahkan tag ke instance dan volume pada pembuatan, pengguna harus memiliki `ec2:CreateTags` izin dalam kebijakan IAM. Untuk informasi selengkapnya, lihat [Izin yang diperlukan untuk menandai instance dan volume](#).
- `iam: PassRole` — Untuk meluncurkan instans EC2 dari template peluncuran yang berisi profil instance (wadah untuk peran IAM), pengguna juga harus memiliki `iam:PassRole` izin dalam kebijakan IAM. Untuk informasi selengkapnya dan contoh kebijakan IAM, lihat [Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2](#).
- `ssm: GetParameters` — Untuk meluncurkan instans EC2 dari template peluncuran yang menggunakan AWS Systems Manager parameter, pengguna juga harus memiliki `ssm:GetParameters` izin dalam kebijakan IAM. Untuk informasi selengkapnya, lihat [Gunakan AWS Systems Manager parameter alih-alih ID AMI di templat peluncuran](#).

Izin untuk tindakan yang harus diselesaikan saat meluncurkan instance dicentang saat pengguna berinteraksi dengan grup Auto Scaling. Untuk informasi selengkapnya, lihat [Validasi izin untuk dan ec2:RunInstancesiam:PassRole](#).

Contoh berikut menunjukkan pernyataan kebijakan yang dapat Anda gunakan untuk mengontrol akses yang pengguna IAM harus menggunakan template peluncuran.

Topik

- [Memerlukan template peluncuran yang memiliki tag tertentu](#)
- [Memerlukan template peluncuran dan nomor versi](#)
- [Memerlukan penggunaan layanan metadata instance versi 2 \(IMDSv2\)](#)
- [Batasi akses ke sumber daya Amazon EC2](#)
- [Izin yang diperlukan untuk menandai instance dan volume](#)
- [Izin template peluncuran tambahan](#)
- [Validasi izin untuk dan ec2:RunInstancesiam:PassRole](#)
- [Sumber daya terkait](#)

Memerlukan template peluncuran yang memiliki tag tertentu

Saat memberikan `ec2:RunInstances` izin, Anda dapat menentukan bahwa pengguna hanya dapat menggunakan templat peluncuran dengan tag tertentu atau ID tertentu untuk membatasi izin saat meluncurkan instance dengan templat peluncuran. Anda juga dapat mengontrol AMI dan sumber daya lain yang dapat direferensikan dan digunakan oleh siapa pun yang menggunakan templat peluncuran saat meluncurkan instance dengan menentukan izin tingkat sumber daya tambahan untuk panggilan tersebut. `RunInstances`

Contoh berikut membatasi izin untuk `ec2:RunInstances` tindakan untuk meluncurkan template yang terletak di Wilayah tertentu dan yang memiliki tag. **purpose=testing** Ini juga memberi pengguna akses ke sumber daya yang ditentukan dalam template peluncuran: AMI, jenis instance, volume, pasangan kunci, antarmuka jaringan, dan grup keamanan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
```

```

    "Resource": "arn:aws:ec2:region:account-id:launch-template/*",
    "Condition": {
      "StringEquals": { "aws:ResourceTag/purpose": "testing" }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
      "arn:aws:ec2:region::image/ami-*",
      "arn:aws:ec2:region:account-id:instance/*",
      "arn:aws:ec2:region:account-id:subnet/*",
      "arn:aws:ec2:region:account-id:volume/*",
      "arn:aws:ec2:region:account-id:key-pair/*",
      "arn:aws:ec2:region:account-id:network-interface/*",
      "arn:aws:ec2:region:account-id:security-group*"
    ]
  }
]
}

```

Untuk informasi selengkapnya tentang penggunaan kebijakan berbasis tag dengan templat peluncuran, lihat [Mengontrol akses ke templat peluncuran dengan izin IAM](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Memerlukan template peluncuran dan nomor versi

Anda juga dapat menggunakan izin IAM untuk menegaskan bahwa template peluncuran dan nomor versi template peluncuran harus ditentukan saat membuat atau memperbarui grup Auto Scaling.

Contoh berikut memungkinkan pengguna untuk membuat dan memperbarui grup Auto Scaling hanya jika template peluncuran dan nomor versi template peluncuran ditentukan. Jika pengguna dengan kebijakan ini menghilangkan nomor versi untuk menentukan versi templat `$Latest` atau `$Default` peluncuran, atau mencoba menggunakan konfigurasi peluncuran, tindakan akan gagal.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:CreateAutoScalingGroup",

```

```

        "autoscaling:UpdateAutoScalingGroup"
    ],
    "Resource": "*",
    "Condition": {
        "Bool": { "autoscaling:LaunchTemplateVersionSpecified": "true" }
    }
},
{
    "Effect": "Deny",
    "Action": [
        "autoscaling:CreateAutoScalingGroup",
        "autoscaling:UpdateAutoScalingGroup"
    ],
    "Resource": "*",
    "Condition": {
        "Null": { "autoscaling:LaunchConfigurationName": "false" }
    }
}
]
}

```

Memerlukan penggunaan layanan metadata instance versi 2 (IMDSv2)

Untuk keamanan ekstra, Anda dapat mengatur izin pengguna agar memerlukan penggunaan templat peluncuran yang memerlukan IMDSv2. Untuk informasi selengkapnya, lihat [Mengonfigurasi instance metadata service](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Contoh berikut menetapkan bahwa pengguna tidak dapat memanggil `ec2:RunInstances` tindakan kecuali instance juga memilih untuk meminta penggunaan ImDSv2 (ditunjukkan oleh `"ec2:MetadataHttpTokens":"required"`)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RequireImdsV2",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Condition": {
        "StringNotEquals": { "ec2:MetadataHttpTokens": "required" }
      }
    }
  ]
}

```

```
]
}
```

Tip

Untuk memaksa instance Auto Scaling pengganti diluncurkan yang menggunakan template peluncuran baru atau versi baru template peluncuran dengan opsi metadata instans yang dikonfigurasi, Anda dapat memulai penyegaran instans. Untuk informasi selengkapnya, lihat [Perbarui instans Auto Scaling](#).

Batasi akses ke sumber daya Amazon EC2

Contoh berikut mengontrol konfigurasi instans yang dapat diluncurkan pengguna dengan membatasi akses ke sumber daya Amazon EC2. Untuk menentukan izin tingkat sumber daya untuk sumber daya yang ditentukan dalam templat peluncuran, Anda harus menyertakan sumber daya dalam pernyataan tindakan. RunInstances

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:region:account-id:launch-template/*",
        "arn:aws:ec2:region::image/ami-04d5cc9b88example",
        "arn:aws:ec2:region:account-id:subnet/subnet-1a2b3c4d",
        "arn:aws:ec2:region:account-id:volume/*",
        "arn:aws:ec2:region:account-id:key-pair/*",
        "arn:aws:ec2:region:account-id:network-interface/*",
        "arn:aws:ec2:region:account-id:security-group/sg-903004f88example"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:region:account-id:instance/*",
      "Condition": {
        "StringEquals": { "ec2:InstanceType": ["t2.micro", "t2.small"] }
      }
    }
  ]
}
```

```

    }
  ]
}

```

Dalam contoh ini, ada dua pernyataan:

- Pernyataan pertama mengharuskan pengguna meluncurkan instance ke subnet (**subnet-1a2b3c4d**) tertentu, menggunakan grup keamanan tertentu (**sg-903004f88example**), dan menggunakan AMI (**ami-04d5cc9b88example**) tertentu. Ini juga memberi pengguna akses ke sumber daya yang ditentukan dalam template peluncuran: antarmuka jaringan, pasangan kunci, dan volume.
- Pernyataan kedua memungkinkan pengguna untuk meluncurkan instance hanya menggunakan tipe **t2.micro** dan **t2.small** instance, yang mungkin Anda lakukan untuk mengontrol biaya.

Namun, perhatikan bahwa saat ini tidak ada cara yang efektif untuk sepenuhnya mencegah pengguna yang memiliki izin untuk meluncurkan instance dengan template peluncuran meluncurkan jenis instance lainnya. Ini karena jenis instance yang ditentukan dalam template peluncuran dapat diganti untuk menggunakan tipe instance yang didefinisikan menggunakan pemilihan tipe instance berbasis atribut.

Untuk daftar lengkap izin tingkat sumber daya yang dapat Anda gunakan untuk mengontrol konfigurasi instans yang dapat diluncurkan pengguna, lihat [Kunci tindakan, sumber daya, dan kondisi untuk Amazon EC2](#) di Referensi Otorisasi Layanan.

Izin yang diperlukan untuk menandai instance dan volume

Contoh berikut memungkinkan pengguna untuk menandai instance dan volume pada pembuatan. Kebijakan ini diperlukan jika ada tag yang ditentukan dalam template peluncuran. Untuk informasi selengkapnya, lihat [Memberikan izin untuk menandai sumber daya selama pembuatan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:region:account-id:*/*",
      "Condition": {

```

```

        "StringEquals": { "ec2:CreateAction": "RunInstances" }
    }
}
]
}

```

Izin template peluncuran tambahan

Anda harus memberikan izin kepada pengguna konsol untuk `ec2:DescribeLaunchTemplateVersions` tindakan `ec2:DescribeLaunchTemplates` dan tindakan. Tanpa izin ini, data template peluncuran tidak dapat dimuat di wizard grup Auto Scaling, dan pengguna tidak dapat melangkah melalui wizard untuk meluncurkan instance menggunakan template peluncuran. Anda dapat menentukan tindakan tambahan ini dalam Action elemen pernyataan kebijakan IAM.

Validasi izin untuk dan **ec2:RunInstancesiam:PassRole**

Pengguna dapat menentukan versi template peluncuran mana yang digunakan grup Auto Scaling mereka. Bergantung pada izin mereka, ini bisa berupa versi bernomor tertentu, atau `$Default` versi `$Latest` atau dari template peluncuran. Jika yang terakhir, berhati-hatilah. Ini dapat mengesampingkan izin untuk `ec2:RunInstances` dan `iam:PassRole` yang ingin Anda batasi.

Bagian ini menjelaskan skenario penggunaan template peluncuran versi terbaru atau default dengan grup Auto Scaling.

Saat pengguna memanggil `CreateAutoScalingGroup`, `UpdateAutoScalingGroup`, atau `StartInstanceRefresh` API, Auto Scaling Amazon EC2 memeriksa izin mereka terhadap versi template peluncuran yang merupakan versi terbaru atau default pada saat itu sebelum melanjutkan permintaan. Ini memvalidasi izin untuk tindakan yang akan diselesaikan saat meluncurkan instance, seperti tindakan dan tindakan. `ec2:RunInstances` `iam:PassRole` Untuk mencapai hal ini, kami mengeluarkan panggilan dry run Amazon [RunInstances](#)EC2 untuk memvalidasi apakah pengguna memiliki izin yang diperlukan untuk tindakan tersebut, tanpa benar-benar membuat permintaan. Ketika respons dikembalikan, itu dibaca oleh Amazon EC2 Auto Scaling. Jika izin pengguna tidak mengizinkan tindakan yang diberikan, Auto Scaling Amazon EC2 akan gagal dalam permintaan dan mengembalikan kesalahan kepada pengguna yang berisi informasi tentang izin yang hilang.

[Setelah verifikasi awal dan permintaan selesai, setiap kali instans diluncurkan, Amazon EC2 Auto Scaling meluncurkannya dengan versi terbaru atau default, meskipun telah berubah, menggunakan izin peran terkait layanannya.](#) Ini berarti bahwa pengguna yang menggunakan template peluncuran

berpotensi memperbaruinya untuk meneruskan peran IAM ke instance meskipun mereka tidak memiliki `iam:PassRole` izin.

Gunakan tombol `autoscaling:LaunchTemplateVersionSpecified` kondisi jika Anda ingin membatasi siapa yang memiliki akses untuk mengonfigurasi grup untuk menggunakan `$Default` versi `$Latest` atau. Ini memastikan bahwa grup Auto Scaling hanya menerima versi bernomor tertentu saat pengguna memanggil API dan. `CreateAutoScalingGroup` `UpdateAutoScalingGroup` Untuk contoh yang menunjukkan cara menambahkan kunci kondisi ini ke kebijakan IAM, lihat [Memerlukan template peluncuran dan nomor versi](#).

Untuk grup Auto Scaling yang dikonfigurasi untuk menggunakan `$Latest` atau `$Default` meluncurkan versi templat, pertimbangkan untuk membatasi siapa yang dapat membuat dan mengelola versi template peluncuran, termasuk `ec2:ModifyLaunchTemplate` tindakan yang memungkinkan pengguna menentukan versi templat peluncuran default. Untuk informasi selengkapnya, lihat [Mengontrol izin pembuatan versi](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Sumber daya terkait

Untuk mempelajari lebih lanjut tentang izin untuk melihat, membuat, dan menghapus templat peluncuran dan meluncurkan versi templat, lihat [Mengontrol akses untuk meluncurkan templat dengan izin IAM](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk informasi selengkapnya tentang izin tingkat sumber daya yang dapat Anda gunakan untuk mengontrol akses ke `RunInstances` panggilan, lihat [Tindakan, sumber daya, dan kunci kondisi untuk Amazon EC2 di Referensi Otorisasi Layanan](#).

Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2

Aplikasi yang berjalan di instans Amazon EC2 memerlukan kredensial untuk mengakses lainnya. Layanan AWS Untuk memberikan kredensial ini dengan cara yang aman, gunakan peran IAM. Peran tersebut menyediakan izin sementara yang dapat digunakan aplikasi saat mengakses sumber daya lain AWS . Izin peran menentukan apa yang dapat dilakukan aplikasi.

Untuk instance dalam grup Auto Scaling, Anda harus membuat templat peluncuran atau memulai konfigurasi dan memilih profil instance untuk dikaitkan dengan instance. Profil instance adalah kontainer untuk peran IAM yang memungkinkan Amazon EC2 melewati peran IAM ke suatu instance saat instance diluncurkan. Pertama, buat peran IAM yang memiliki semua izin yang diperlukan untuk mengakses sumber daya. AWS Kemudian, buat profil instance dan tugaskan peran ke profil tersebut.

Note

Sebagai praktik terbaik, kami sangat menyarankan Anda membuat peran sehingga memiliki izin minimum ke yang lain Layanan AWS yang dibutuhkan aplikasi Anda.

Daftar Isi

- [Prasyarat](#)
- [Membuat templat peluncuran](#)
- [Lihat juga](#)

Prasyarat

Buat peran IAM yang dapat dijalankan aplikasi Anda di Amazon EC2. Pilih izin yang sesuai sehingga aplikasi yang kemudian diberikan peran tersebut dapat melakukan panggilan API khusus yang diperlukannya.

Jika Anda menggunakan konsol IAM alih-alih AWS SDK AWS CLI atau salah satu, konsol akan membuat profil instance secara otomatis dan memberinya nama yang sama dengan peran yang sesuai dengannya.

Untuk membuat peran IAM (konsol)

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Pada panel navigasi di sebelah kiri, pilih Peran.
3. Pilih Buat peran.
4. Untuk Pilih entitas tepercaya, pilih AWS layanan.
5. Untuk kasus penggunaan Anda, pilih EC2 dan kemudian pilih Berikutnya.
6. Jika memungkinkan, pilih kebijakan yang akan digunakan untuk kebijakan izin atau pilih Buat kebijakan untuk membuka tab browser baru dan membuat kebijakan baru dari awal. Untuk informasi selengkapnya, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM. Setelah Anda membuat kebijakan, tutup tab tersebut dan kembali ke tab asli Anda. Pilih kotak centang di samping kebijakan izin yang ingin dimiliki pengguna identitas web.
7. (Opsional) Tetapkan [batas izin](#). Ini adalah fitur lanjutan yang tersedia untuk peran layanan. Untuk informasi selengkapnya, lihat [Batas izin untuk entitas IAM](#) di Panduan Pengguna IAM.
8. Pilih Berikutnya.

9. Pada halaman Nama, tinjau, dan buat, untuk nama Peran, masukkan nama peran untuk membantu Anda mengidentifikasi tujuan peran ini. Nama ini harus unik di dalam diri Anda Akun AWS. Karena AWS sumber daya lain mungkin mereferensikan peran, Anda tidak dapat mengedit nama peran setelah dibuat.
10. Tinjau peran lalu pilih Buat peran.

Izin IAM

Gunakan kebijakan berbasis identitas IAM untuk mengontrol akses ke peran IAM baru Anda.

`iam:PassRole` diperlukan pada identitas IAM (pengguna atau peran) yang membuat atau memperbarui grup Auto Scaling menggunakan template peluncuran yang menentukan profil instance.

Contoh kebijakan berikut memberikan izin untuk hanya meneruskan peran IAM yang namanya dimulai. **gateam-**

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id:role/gateam-*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "ec2.amazonaws.com",
            "ec2.amazonaws.com.cn"
          ]
        }
      }
    }
  ]
}
```

Important

Untuk informasi tentang cara Auto Scaling Amazon EC2 memvalidasi izin untuk tindakan `iam:PassRole` untuk grup Auto Scaling yang menggunakan templat peluncuran, lihat.

[Validasi izin untuk dan `ec2:RunInstancesiam:PassRole`](#)

Membuat templat peluncuran

Saat Anda membuat template peluncuran menggunakan AWS Management Console, di bagian Detail lanjutan, pilih peran dari profil instans IAM. Untuk informasi selengkapnya, lihat [Buat template peluncuran menggunakan pengaturan lanjutan](#).

Saat Anda membuat template peluncuran menggunakan [create-launch-template](#) perintah dari AWS CLI, tentukan nama profil instance peran IAM Anda seperti yang ditunjukkan pada contoh berikut.

```
aws ec2 create-launch-template --launch-template-name my-lt-with-instance-profile --  
version-description version1 \  
--launch-template-data  
'{"ImageId": "ami-04d5cc9b88example", "InstanceType": "t2.micro", "IamInstanceProfile":  
{"Name": "my-instance-profile"}}'
```

Lihat juga

Untuk informasi selengkapnya guna membantu Anda mulai mempelajari dan menggunakan peran IAM untuk Amazon EC2, lihat:

- [Peran IAM untuk Amazon EC2](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux
- [Menggunakan profil instans](#) dan [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2](#) dalam Panduan Pengguna IAM

Validasi kepatuhan untuk Amazon EC2 Auto Scaling


Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan khusus, lihat [Layanan AWS di Scope oleh Program](#) Program Kepatuhan yang Anda minati. Untuk informasi umum, lihat [Program Kepatuhan 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 berdasarkan sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Mulai Cepat Keamanan dan Kepatuhan](#) – Panduan deployment ini membahas pertimbangan arsitektur dan menyediakan langkah-langkah untuk melakukan deployment lingkungan dasar di AWS yang menjadi fokus keamanan dan kepatuhan.

- [Merancang Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) – Laporan resmi ini menjelaskan cara perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

 Note

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

- [Sumber Daya Kepatuhan AWS](#) – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [Panduan Kepatuhan Pelanggan AWS](#) – Pahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan kontrol keamanan di banyak kerangka kerja (termasuk National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), dan International Organization for Standardization (ISO)).
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan Developer AWS Config – Layanan AWS Config menilai seberapa baik konfigurasi sumber daya Anda dalam mematuhi praktik-praktik internal, pedoman industri, dan regulasi internal.
- [AWS Security Hub](#) – Layanan AWS ini memberikan pandangan komprehensif tentang status keamanan Anda di dalam AWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi sumber daya AWS Anda dan memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [Referensi kontrol Security Hub](#).
- [AWS Audit Manager](#) – Layanan AWS ini akan membantu Anda untuk terus-menerus mengaudit penggunaan AWS untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap regulasi dan standar industri.

Kepatuhan PCI DSS

Auto Scaling Amazon EC2 mendukung pemrosesan, penyimpanan, dan transmisi data kartu kredit oleh merchant atau penyedia layanan, dan telah divalidasi sesuai dengan Standar Keamanan Data (DSS) Industri Kartu Pembayaran (PCI). Untuk informasi tentang DSS PCI selengkapnya, termasuk cara meminta salinan AWS PCI Compliance Package, lihat [DSS PCI Level 1](#).

Untuk informasi tentang mencapai kepatuhan PCI DSS untuk AWS beban kerja Anda, lihat panduan kepatuhan berikut:

- [Standar Keamanan Data Industri Kartu Pembayaran \(PCI DSS\) 3.2.1 pada AWS](#)

Amazon EC2 Auto Scaling dan endpoint VPC antarmuka

Anda dapat meningkatkan postur keamanan VPC Anda dengan mengonfigurasi Auto Scaling Amazon EC2 untuk menggunakan titik akhir VPC antarmuka. Titik akhir antarmuka didukung oleh AWS PrivateLink, teknologi yang memungkinkan Anda mengakses API Auto Scaling Amazon EC2 secara pribadi dengan membatasi semua lalu lintas jaringan antara VPC dan Amazon EC2 Auto Scaling ke jaringan. AWS Dengan titik akhir antarmuka, Anda juga tidak memerlukan sebuah gerbang internet, perangkat NAT, atau gerbang pribadi virtual.

Anda tidak diharuskan untuk mengkonfigurasi AWS PrivateLink, tetapi disarankan. [Untuk informasi selengkapnya tentang AWS PrivateLink dan titik akhir VPC, lihat Apa itu? AWS PrivateLink](#) dalam AWS PrivateLink Guide.

Topik

- [Buat endpoint VPC antarmuka](#)
- [Buat kebijakan endpoint VPC](#)

Buat endpoint VPC antarmuka

Buat endpoint untuk Amazon EC2 Auto Scaling menggunakan nama layanan berikut:

```
com.amazonaws.region.autoscaling
```

Untuk informasi selengkapnya, lihat [Mengakses AWS layanan menggunakan titik akhir VPC antarmuka di Panduan](#).AWS PrivateLink

Anda tidak perlu mengubah pengaturan Amazon EC2 Auto Scaling. Auto Scaling Amazon EC2 memanggil layanan AWS lain menggunakan titik akhir layanan atau titik akhir VPC antarmuka pribadi, mana pun yang digunakan.

Buat kebijakan endpoint VPC

Anda dapat melampirkan kebijakan ke endpoint VPC Anda untuk mengontrol akses ke Amazon EC2 Auto Scaling API. Kebijakan menentukan:

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

Contoh berikut menunjukkan kebijakan endpoint VPC yang menolak izin setiap orang untuk menghapus kebijakan penskalaan melalui endpoint. Kebijakan contoh juga memberikan izin kepada semua orang untuk melakukan semua tindakan lainnya.

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

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

Memecahkan Masalah Auto Scaling Amazon EC2

Amazon EC2 Auto Scaling memberikan kesalahan deskriptif dan spesifik untuk membantu Anda memecahkan masalah. Anda dapat menemukan pesan kesalahan dalam deskripsi aktivitas penskalaan.

Topik

- [Mengambil pesan kesalahan dari aktivitas penskalaan](#)
- [Matikan aktivitas penskalaan](#)
- [Sumber daya pemecahan masalah tambahan](#)
- [Memecahkan Masalah Auto Scaling Amazon EC2: Kegagalan peluncuran instans EC2](#)
- [Memecahkan Masalah Auto Scaling Amazon EC2: Masalah AMI](#)
- [Memecahkan Masalah Auto Scaling Amazon EC2: Masalah penyeimbang beban](#)
- [Memecahkan Masalah Auto Scaling Amazon EC2: Luncurkan templat](#)
- [Memecahkan Masalah Auto Scaling Amazon EC2: Pemeriksaan Kesehatan](#)

Mengambil pesan kesalahan dari aktivitas penskalaan

Untuk mengambil pesan kesalahan dari deskripsi aktivitas penskalaan, gunakan perintah. [describe-scaling-activities](#) Anda memiliki catatan aktivitas penskalaan yang berasal dari 6 minggu yang lalu. Aktivitas penskalaan diurutkan berdasarkan waktu mulai, dengan aktivitas penskalaan terbaru terdaftar terlebih dahulu.

Note

Aktivitas penskalaan juga ditampilkan dalam riwayat aktivitas di konsol Auto Scaling Amazon EC2 di tab Aktivitas untuk grup Auto Scaling.

Untuk melihat aktivitas penskalaan untuk grup Auto Scaling tertentu, gunakan perintah berikut.

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg
```

Berikut ini adalah contoh respons, di mana `StatusCode` berisi status aktivitas saat ini dan `StatusMessage` berisi pesan kesalahan.

```
{
  "Activities": [
    {
      "ActivityId": "3b05dbf6-037c-b92f-133f-38275269dc0f",
      "AutoScalingGroupName": "my-asg",
      "Description": "Launching a new EC2 instance: i-003a5b3ffe1e9358e. Status Reason: Instance failed to complete user's Lifecycle Action: Lifecycle Action with token e85eb647-4fe0-4909-b341-a6c42d8aba1f was abandoned: Lifecycle Action Completed with ABANDON Result",
      "Cause": "At 2021-01-11T00:35:52Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 1. At 2021-01-11T00:35:53Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 1.",
      "StartTime": "2021-01-11T00:35:55.542Z",
      "EndTime": "2021-01-11T01:06:31Z",
      "StatusCode": "Cancelled",
      "StatusMessage": "Instance failed to complete user's Lifecycle Action: Lifecycle Action with token e85eb647-4fe0-4909-b341-a6c42d8aba1f was abandoned: Lifecycle Action Completed with ABANDON Result",
      "Progress": 100,
      "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-west-2b\"...}",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:283179a2-f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
    },
    ...
  ]
}
```

Untuk deskripsi bidang dalam output, lihat [Aktivitas](#) di Referensi API Auto Scaling Amazon EC2.

Untuk melihat aktivitas penskalaan untuk grup yang dihapus

Untuk melihat aktivitas penskalaan setelah grup Auto Scaling dihapus, tambahkan opsi `--include-deleted-groups` ke perintah [describe-scaling-activities](#) sebagai berikut.

```
aws autoscaling describe-scaling-activities --auto-scaling-group-name my-asg --include-deleted-groups
```

Berikut ini adalah contoh respons, dengan aktivitas penskalaan untuk grup yang dihapus.


```
{
  "Activities": [
    {
      "ActivityId": "e1f5de0e-f93e-1417-34ac-092a76fba220",
      "AutoScalingGroupName": "my-asg",
      "Description": "Launching a new EC2 instance. Status Reason: Your Spot request price of 0.001 is lower than the minimum required Spot request fulfillment price of 0.0031. Launching EC2 instance failed.",
      "Cause": "At 2021-01-13T20:47:24Z a user request update of AutoScalingGroup constraints to min: 1, max: 5, desired: 3 changing the desired capacity from 0 to 3. At 2021-01-13T20:47:27Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 3.",
      "StartTime": "2021-01-13T20:47:30.094Z",
      "EndTime": "2021-01-13T20:47:30Z",
      "StatusCode": "Failed",
      "StatusMessage": "Your Spot request price of 0.001 is lower than the minimum required Spot request fulfillment price of 0.0031. Launching EC2 instance failed.",
      "Progress": 100,
      "Details": "{\"Subnet ID\":\"subnet-5ea0c127\",\"Availability Zone\":\"us-west-2b\"...}",
      "AutoScalingGroupState": "Deleted",
      "AutoScalingGroupARN": "arn:aws:autoscaling:us-west-2:123456789012:autoScalingGroup:283179a2-f3ce-423d-93f6-66bb518232f7:autoScalingGroupName/my-asg"
    },
    ...
  ]
}
```

Matikan aktivitas penskalaan

Anda memiliki opsi berikut jika perlu menyelidiki masalah tanpa gangguan dari kebijakan penskalaan atau tindakan terjadwal:

- Mencegah semua kebijakan penskalaan dan tindakan terjadwal membuat perubahan pada kapasitas yang diinginkan grup dengan menanggihkan AlarmNotification dan ScheduledActions proses. Untuk informasi selengkapnya, lihat [Menanggihkan dan melanjutkan proses Auto Scaling Amazon EC2](#).

- Nonaktifkan kebijakan penskalaan individual sehingga tidak mengubah kapasitas yang diinginkan grup sebagai respons terhadap perubahan pemuatan. Untuk informasi selengkapnya, lihat [Menonaktifkan kebijakan penskalaan untuk grup Auto Scaling](#).
- Perbarui kebijakan penskalaan pelacakan target individual agar hanya memperkecil skala (menambah kapasitas) dengan menonaktifkan bagian penskalaan kebijakan. Metode ini mencegah kapasitas yang diinginkan grup menyusut tetapi memungkinkannya ditingkatkan ketika beban meningkat. Untuk informasi selengkapnya, lihat [Kebijakan penskalaan pelacakan target untuk Amazon EC2 Auto Scaling](#).

Sumber daya pemecahan masalah tambahan

Halaman berikut memberikan informasi tambahan untuk mengatasi masalah dengan Auto Scaling Amazon EC2.

- [Memverifikasi aktivitas penskalaan untuk grup Auto Scaling](#)
- [Lihat grafik pemantauan di konsol Auto Scaling Amazon EC2](#)
- [Health memeriksa instans dalam grup Auto Scaling](#)
- [Pertimbangan dan batasan untuk kait siklus hidup](#)
- [Selesaikan tindakan siklus hidup](#)
- [Menyediakan konektivitas jaringan untuk instans Auto Scaling menggunakan Amazon VPC](#)
- [Hapus instans sementara dari grup Auto Scaling](#)
- [Menonaktifkan kebijakan penskalaan untuk grup Auto Scaling](#)
- [Menangguhkan dan melanjutkan proses Auto Scaling Amazon EC2](#)
- [Kontrol instans Auto Scaling mana yang akan dihentikan selama penskalaan](#)
- [Hapus infrastruktur Auto Scaling Anda](#)
- [Kuota untuk Auto Scaling Amazon EC2](#)

AWS Sumber daya berikut juga dapat membantu:

- [Topik Auto Scaling Amazon EC2 di Pusat Pengetahuan AWS](#)
- [Pertanyaan Auto Scaling Amazon EC2 di Re: posting AWS](#)
- [Posting Auto Scaling Amazon EC2 di Blog Komputasi AWS](#)

- [Pemecahan masalah CloudFormation di Panduan Pengguna AWS CloudFormation](#)

Pemecahan masalah sering membutuhkan permintaan berulang dan penemuan oleh seorang ahli atau dari komunitas pembantu. Jika Anda terus mengalami masalah setelah mencoba saran di bagian ini, hubungi AWS Support (di, klik Support AWS Management Console, Support Center) atau ajukan pertanyaan di [AWS re:Post](#) menggunakan tag Amazon EC2 Auto Scaling.

Memecahkan Masalah Auto Scaling Amazon EC2: Kegagalan peluncuran instans EC2

Halaman ini memberikan informasi tentang instance EC2 Anda yang gagal diluncurkan, kemungkinan penyebab, dan langkah-langkah yang dapat Anda ambil untuk memecahkan masalah.

Untuk mengambil data pesan kesalahan, lihat [Mengambil pesan kesalahan dari aktivitas penskalaan](#).

Saat instance EC2 Anda gagal diluncurkan, Anda mungkin mendapatkan satu atau beberapa pesan kesalahan berikut:

Masalah peluncuran

- [Konfigurasi yang diminta saat ini tidak didukung.](#)
- [Grup keamanan <name of the security group> tidak ada. Peluncuran instance EC2 gagal.](#)
- [Pasangan kunci <key pair associated with your EC2 instance> tidak ada. Peluncuran instance EC2 gagal.](#)
- [Jenis instance yang Anda minta \(<instance type>\) tidak didukung dalam Zona Ketersediaan yang Anda minta \(<instance Availability Zone>\)...](#)
- [Harga permintaan Spot Anda sebesar 0,015 lebih rendah dari harga pemenuhan permintaan Spot minimum 0,0735...](#)
- [Nama perangkat tidak <device name>valid/Unggahan nama perangkat tidak valid. Peluncuran instance EC2 gagal.](#)
- [Value \(<name associated with the instance storage device>\) untuk parameter VirtualName tidak valid... Peluncuran instance EC2 gagal.](#)
- [EBS memblokir pemetaan perangkat tidak didukung untuk AMI instance-store.](#)
- [Grup penempatan tidak boleh digunakan dengan contoh tipe '<instance type>'. Peluncuran instance EC2 gagal.](#)

- [Klien. InternalError: Kesalahan klien saat diluncurkan.](#)
- [Saat ini kami tidak memiliki <instance type>kapasitas yang cukup di Availability Zone yang Anda minta... Peluncuran instance EC2 gagal.](#)
- [Reservasi yang diminta tidak memiliki kapasitas yang kompatibel dan tersedia untuk permintaan ini. Peluncuran instance EC2 gagal.](#)
- [Reservasi Blok Kapasitas Anda <reservation id>belum aktif. Peluncuran instance EC2 gagal.](#)
- [Tidak ada kapasitas Spot yang tersedia yang sesuai dengan permintaan Anda. Peluncuran instance EC2 gagal.](#)
- [<number of instances> instance telah berjalan. Peluncuran instance EC2 gagal.](#)

Konfigurasi yang diminta saat ini tidak didukung.

Penyebab: Beberapa opsi dalam templat peluncuran atau konfigurasi peluncuran mungkin tidak kompatibel dengan jenis instans, atau konfigurasi instans mungkin tidak didukung di AWS Wilayah atau Zona Ketersediaan yang Anda minta.

Solusi: Coba konfigurasi instance yang berbeda. Untuk mencari jenis instans yang memenuhi persyaratan Anda, lihat [Menemukan jenis instans Amazon EC2](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk panduan lebih lanjut untuk mengatasi masalah ini, periksa hal berikut:

- Pastikan Anda telah memilih AMI yang didukung oleh jenis instans Anda. Misalnya, jika tipe instans menggunakan prosesor AWS Graviton berbasis ARM alih-alih prosesor Intel Xeon, Anda memerlukan AMI yang kompatibel dengan ARM. Untuk informasi selengkapnya tentang memilih jenis instans yang kompatibel, lihat [Kompatibilitas untuk mengubah jenis instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Uji apakah jenis instans tersedia di Wilayah dan Zona Ketersediaan yang Anda minta. Jenis instans generasi terbaru mungkin belum tersedia di Wilayah atau Zona Ketersediaan tertentu. Jenis instans generasi lama mungkin tidak tersedia di Wilayah dan Zona Ketersediaan yang lebih baru. Untuk mencari jenis contoh yang ditawarkan berdasarkan lokasi (Region atau Availability Zone), gunakan [describe-instance-type-offerings](#) perintah. Untuk informasi selengkapnya, lihat [Menemukan jenis instans Amazon EC2](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Jika Anda menggunakan Instans Khusus atau Host Khusus, pastikan Anda telah memilih jenis instans yang didukung sebagai Dedicated Instance atau Dedicated Host.

Grup keamanan <name of the security group> tidak ada. Peluncuran instance EC2 gagal.

Penyebab: Grup keamanan yang ditentukan dalam templat peluncuran atau konfigurasi peluncuran Anda mungkin telah dihapus.

Solusi:

1. Gunakan [describe-security-groups](#) perintah untuk mendapatkan daftar grup keamanan yang terkait dengan akun Anda.
2. Dari daftar, pilih grup keamanan yang akan digunakan. Untuk membuat grup keamanan sebagai gantinya, gunakan [create-security-group](#) perintah.
3. Buat template peluncuran baru atau konfigurasi peluncuran.
4. Perbarui grup Auto Scaling Anda dengan templat peluncuran baru atau konfigurasi peluncuran menggunakan perintah. [update-auto-scaling-group](#)

Pasangan kunci <key pair associated with your EC2 instance> tidak ada. Peluncuran instance EC2 gagal.

Penyebab: Key pair yang digunakan saat meluncurkan instance mungkin telah dihapus.

Solusi:

1. Gunakan [describe-key-pairs](#) perintah untuk mendapatkan daftar pasangan kunci yang tersedia untuk Anda.
2. Dari daftar, pilih pasangan kunci yang akan digunakan. Untuk membuat key pair sebagai gantinya, gunakan [create-key-pair](#) perintah.
3. Buat template peluncuran baru atau konfigurasi peluncuran.
4. Perbarui grup Auto Scaling Anda dengan templat peluncuran baru atau konfigurasi peluncuran menggunakan perintah. [update-auto-scaling-group](#)

Jenis instance yang Anda minta (<instance type>) tidak didukung dalam Zona Ketersediaan yang Anda minta (<instance Availability Zone>)...

Pesan galat: Jenis instans yang Anda minta (<instance type>) tidak didukung di Availability Zone (<instance Availability Zone>) yang Anda minta... Peluncuran instans EC2 gagal.

Penyebab: Availability Zone yang ditentukan dalam grup Auto Scaling tidak mendukung jenis instans yang Anda pilih.

Solusi:

1. Verifikasi Availability Zones yang mendukung jenis instans yang Anda pilih menggunakan [describe-instance-type-offerings](#) perintah atau dari konsol Amazon EC2 dengan memeriksa nilai Availability Zones pada panel jaringan halaman Jenis instans.
2. Perbarui atau hapus subnet untuk zona yang tidak didukung dalam pengaturan grup Auto Scaling Anda menggunakan perintah. [update-auto-scaling-group](#) Untuk informasi selengkapnya, lihat [Menambahkan dan menghapus Availability Zones](#).

Harga permintaan Spot Anda sebesar 0,015 lebih rendah dari harga pemenuhan permintaan Spot minimum 0,0735...

Penyebab: Harga maksimum Spot dalam permintaan Anda lebih rendah dari harga Spot untuk jenis instans yang Anda pilih.

Solusi: Kirim permintaan baru dengan harga maksimum Spot yang lebih tinggi (mungkin harga On-Demand). Sebelumnya, harga Spot yang Anda bayarkan didasarkan pada penawaran. Hari ini, Anda membayar harga Spot saat ini. Dengan menetapkan harga maksimum Anda lebih tinggi, ini memberi layanan Amazon EC2 Spot kesempatan yang lebih baik untuk meluncurkan dan mempertahankan jumlah kapasitas yang Anda butuhkan.

Nama perangkat tidak <device name>valid/Unggahan nama perangkat tidak valid. Peluncuran instance EC2 gagal.

Penyebab 1: Pemetaan perangkat blokir di templat peluncuran atau konfigurasi peluncuran mungkin berisi nama perangkat blok yang tidak tersedia atau saat ini tidak didukung.

Solusi:

1. Verifikasi nama perangkat mana yang tersedia untuk konfigurasi instans spesifik Anda. Untuk detail selengkapnya tentang penamaan [perangkat, lihat Nama perangkat di instans Linux](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
2. Buat instans Amazon EC2 secara manual yang bukan bagian dari grup Auto Scaling dan selidiki masalahnya. Jika konfigurasi penamaan perangkat blok bertentangan dengan nama di Amazon

- Machine Image (AMI), instance akan gagal selama peluncuran. Untuk informasi selengkapnya, lihat [Memblokir pemetaan perangkat](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
3. Setelah mengonfirmasi bahwa instans berhasil diluncurkan, gunakan perintah [describe-volume](#) untuk melihat bagaimana volume diekspos ke instance.
 4. Buat templat peluncuran baru atau konfigurasi peluncuran menggunakan nama perangkat yang tercantum dalam deskripsi volume.
 5. Perbarui grup Auto Scaling Anda dengan templat peluncuran baru atau konfigurasi peluncuran menggunakan perintah. [update-auto-scaling-group](#)

Value (<name associated with the instance storage device>) untuk parameter VirtualName tidak valid... Peluncuran instance EC2 gagal.

Penyebab: Format yang ditentukan untuk nama virtual yang terkait dengan perangkat blok tidak benar.

Solusi:

1. Buat template peluncuran baru atau konfigurasi peluncuran dengan menentukan nama perangkat di `virtualName` parameter. Untuk informasi tentang format nama perangkat, lihat [Penamaan perangkat di instance Linux](#) di Panduan Pengguna Amazon EC2 untuk Instance Linux.
2. Perbarui grup Auto Scaling Anda dengan templat peluncuran baru atau konfigurasi peluncuran menggunakan perintah. [update-auto-scaling-group](#)

EBS memblokir pemetaan perangkat tidak didukung untuk AMI instance-store.

Penyebab: Pemetaan perangkat blok yang ditentukan dalam templat peluncuran atau konfigurasi peluncuran tidak didukung pada instans Anda.

Solusi:

1. Buat templat peluncuran baru atau konfigurasi peluncuran dengan pemetaan perangkat blok yang didukung oleh jenis instans Anda. Untuk informasi lebih lanjut, lihat [Pemetaan perangkat blok](#) di Panduan Pengguna Amazon EC2 untuk Instance Linux.
2. Perbarui grup Auto Scaling Anda dengan templat peluncuran baru atau konfigurasi peluncuran menggunakan perintah. [update-auto-scaling-group](#)

Grup penempatan tidak boleh digunakan dengan contoh tipe '<instance type>'. Peluncuran instance EC2 gagal.

Penyebab: Grup penempatan klaster Anda berisi jenis instance yang tidak valid.

Solusi:

1. Untuk informasi tentang tipe contoh valid yang didukung oleh grup penempatan, lihat [Grup penempatan](#) di Panduan Pengguna Amazon EC2 untuk Instance Linux.
2. Ikuti petunjuk yang diuraikan di dalam [Grup penempatan](#) untuk membuat grup penempatan baru.
3. Atau, buat template peluncuran baru atau konfigurasi peluncuran dengan jenis instans yang didukung.
4. Perbarui grup Auto Scaling Anda dengan grup penempatan baru, templat peluncuran, atau konfigurasi peluncuran menggunakan perintah. [update-auto-scaling-group](#)

Klien. InternalError: Kesalahan klien saat diluncurkan.

Masalah: Auto Scaling Amazon EC2 mencoba meluncurkan instance yang memiliki volume EBS terenkripsi, tetapi peran terkait layanan tidak memiliki akses ke kunci terkelola pelanggan yang digunakan untuk AWS KMS mengenkripsinya. Untuk informasi selengkapnya, lihat [Kebijakan AWS KMS kunci yang diperlukan untuk digunakan dengan volume terenkripsi](#).

Penyebab 1: Anda memerlukan kebijakan kunci yang memberikan izin untuk menggunakan kunci yang dikelola pelanggan untuk peran terkait layanan yang tepat.

Solusi 1: Izinkan peran terkait layanan menggunakan kunci yang dikelola pelanggan sebagai berikut:

1. Tentukan peran terkait layanan yang akan digunakan untuk grup Auto Scaling ini.
2. Perbarui kebijakan kunci pada kunci yang dikelola pelanggan dan izinkan peran terkait layanan menggunakan kunci yang dikelola pelanggan.
3. Perbarui grup Auto Scaling untuk menggunakan peran terkait layanan.

Untuk contoh kebijakan kunci yang memungkinkan peran terkait layanan menggunakan kunci terkelola pelanggan, lihat. [Contoh 1: Bagian kebijakan utama yang memungkinkan akses ke kunci yang dikelola pelanggan](#)

Penyebab 2: Jika kunci yang dikelola pelanggan dan grup Auto Scaling berada di AWS akun yang berbeda, Anda perlu mengonfigurasi akses lintas akun ke kunci yang dikelola pelanggan untuk memberikan izin untuk menggunakan kunci yang dikelola pelanggan ke peran terkait layanan yang tepat.

Solusi 2: Izinkan peran terkait layanan di akun eksternal untuk menggunakan kunci yang dikelola pelanggan di akun lokal sebagai berikut:

1. Perbarui kebijakan kunci pada kunci yang dikelola pelanggan untuk memungkinkan akun grup Auto Scaling mengakses kunci yang dikelola pelanggan.
2. Tentukan pengguna IAM atau peran dalam akun grup Auto Scaling yang dapat membuat izin.
3. Tentukan peran terkait layanan yang akan digunakan untuk grup Auto Scaling ini.
4. Buat hibah untuk kunci yang dikelola pelanggan dengan peran terkait layanan yang tepat sebagai prinsipal penerima hibah.
5. Perbarui grup Auto Scaling untuk menggunakan peran terkait layanan.

Untuk informasi selengkapnya, lihat [Contoh 2: Bagian kebijakan utama yang memungkinkan akses lintas akun ke kunci yang dikelola pelanggan](#).

Solusi 3: Gunakan kunci yang dikelola pelanggan di AWS akun yang sama dengan grup Auto Scaling.

1. Salin dan enkripsi ulang snapshot dengan kunci terkelola pelanggan lain yang termasuk dalam akun yang sama dengan grup Auto Scaling.
2. Izinkan peran terkait layanan menggunakan kunci terkelola pelanggan baru. Lihat langkah-langkah untuk Solusi 1.

Saat ini kami tidak memiliki <instance type>kapasitas yang cukup di Availability Zone yang Anda minta... Peluncuran instance EC2 gagal.

Pesan galat: Saat ini kami tidak memiliki <instance type>kapasitas yang cukup di Availability Zone yang Anda minta (<requested Availability Zone>). Sistem kami akan mengupayakan penyediaan kapasitas tambahan. Saat ini Anda mendapat kapasitas <instance type> dengan tidak menentukan Zona Ketersediaan dalam permintaan atau pilihan Anda <list of Availability Zones that currently supports the instance type>. Peluncuran instance EC2 gagal.

Penyebab: Saat ini, jenis instans yang diminta dan kombinasi Availability Zone tidak didukung.

Solusi: Untuk mengatasi masalah ini, coba yang berikut ini:

- Tunggu beberapa menit hingga Auto Scaling Amazon EC2 menemukan kapasitas untuk jenis instans ini di Availability Zone lain yang diaktifkan.
- Perluas grup Auto Scaling Anda ke Availability Zone tambahan. Untuk informasi selengkapnya, lihat [Menambahkan dan menghapus Availability Zones](#).
- Ikuti praktik terbaik menggunakan beragam jenis instance sehingga Anda tidak bergantung pada satu jenis instance tertentu. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).

Reservasi yang diminta tidak memiliki kapasitas yang kompatibel dan tersedia untuk permintaan ini. Peluncuran instance EC2 gagal.

Penyebab 1: Anda telah mencapai batas jumlah instans yang dapat diluncurkan dengan Reservasi Kapasitas targeted Sesuai Permintaan.

Solusi 1: Tingkatkan jumlah instans yang dapat Anda luncurkan dengan Reservasi Kapasitas targeted Sesuai Permintaan, atau gunakan grup Reservasi Kapasitas sehingga apa pun di luar kapasitas cadangan akan diluncurkan sebagai kapasitas Sesuai Permintaan reguler. Untuk informasi selengkapnya, lihat [Gunakan Reservasi Kapasitas Sesuai Permintaan untuk memesan kapasitas di Availability Zone tertentu](#).

Penyebab 2: Anda telah mencapai batas jumlah instance yang dapat Anda luncurkan dengan Blok Kapasitas.

Dengan Blok Kapasitas, Anda dibatasi oleh jumlah kapasitas yang awalnya dibeli. Jika Anda mengalami jumlah peluncuran yang lebih tinggi dari yang diantisipasi dan menggunakan semua kapasitas yang Anda miliki, ini menyebabkan peluncuran gagal. Contoh penghentian melalui proses pembersihan yang panjang sebelum dihentikan sepenuhnya. Selama waktu ini, mereka tidak dapat digunakan kembali. Ini juga dapat menyebabkan peluncuran gagal. Untuk informasi selengkapnya, lihat [Gunakan Blok Kapasitas untuk beban kerja pembelajaran mesin](#).

Solusi 2: Untuk mengatasi masalah ini, coba yang berikut ini:

- Simpan permintaan apa adanya. Jika instans Blok Kapasitas berakhir, Anda harus menunggu beberapa menit agar instans selesai dihentikan dan kapasitas tersedia kembali. Auto Scaling Amazon EC2 terus secara otomatis membuat permintaan peluncuran hingga kapasitas tersedia.

- Pastikan Anda membeli kapasitas yang cukup untuk mengakomodasi beban kerja puncak Anda sehingga Anda tidak sering mengalami kesalahan ini.

Reservasi Blok Kapasitas Anda <reservation id>belum aktif. Peluncuran instance EC2 gagal.

Penyebab: Blok Kapasitas yang ditentukan belum aktif.

Solusi: Ikuti pendekatan yang disarankan untuk Blok Kapasitas dan gunakan penskalaan terjadwal. Melakukan hal itu membantu Anda memastikan Anda meningkatkan kapasitas yang diinginkan dari grup Auto Scaling Anda hanya ketika reservasi aktif dan menguranginya sebelum reservasi selesai.

Tidak ada kapasitas Spot yang tersedia yang sesuai dengan permintaan Anda. Peluncuran instance EC2 gagal.

Penyebab: Saat ini, tidak ada kapasitas cadangan yang cukup untuk memenuhi permintaan Anda untuk Instans Spot.

Solusi: Untuk mengatasi masalah ini, coba yang berikut ini:

- Tunggu beberapa menit; kapasitas bisa sering bergeser. Auto Scaling Amazon EC2 terus secara otomatis membuat permintaan peluncuran hingga kapasitas tersedia.
- Perluas grup Auto Scaling Anda ke Availability Zone tambahan. Untuk informasi selengkapnya, lihat [Menambahkan dan menghapus Availability Zones](#).
- Ikuti praktik terbaik menggunakan beragam jenis instance sehingga Anda tidak bergantung pada satu jenis instance tertentu. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).

<number of instances> instance telah berjalan. Peluncuran instance EC2 gagal.

Penyebab: Anda telah mencapai batas jumlah instans yang dapat Anda luncurkan di Wilayah. Saat Anda membuat AWS akun, kami menetapkan batas default pada jumlah instans yang dapat Anda jalankan per wilayah.

Solusi: Untuk mengatasi masalah ini, coba yang berikut ini:

- Jika batas Anda saat ini tidak memadai untuk kebutuhan Anda, Anda dapat meminta peningkatan kuota per wilayah. Untuk informasi selengkapnya, lihat [kuota layanan Amazon EC2](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Kirim permintaan baru dengan jumlah instance yang berkurang (yang dapat Anda tingkatkan pada tahap selanjutnya).

Memecahkan Masalah Auto Scaling Amazon EC2: Masalah AMI

Halaman ini memberikan informasi tentang masalah yang terkait dengan AMI Anda, penyebab potensial, dan langkah-langkah yang dapat Anda ambil untuk menyelesaikan masalah.

Untuk mengambil data pesan kesalahan, lihat [Mengambil pesan kesalahan dari aktivitas penskalaan](#).

Saat instance EC2 Anda gagal diluncurkan karena masalah dengan AMI Anda, Anda mungkin mendapatkan satu atau beberapa pesan kesalahan berikut,

Masalah AMI

- [ID AMI <ID of your AMI> tidak ada. Peluncuran instance EC2 gagal.](#)
- [AMI <AMI ID> tertunda, dan tidak dapat dijalankan. Peluncuran instance EC2 gagal.](#)
- [<device name>Nama perangkat tidak valid. Peluncuran instance EC2 gagal.](#)
- [Arsitektur 'arm64' dari jenis instance yang ditentukan tidak cocok dengan arsitektur 'x86_64' dari AMI yang ditentukan... Peluncuran instans EC2 gagal.](#)
- [AMI '<AMI ID>' dinonaktifkan, dan tidak dapat dijalankan. Peluncuran instance EC2 gagal.](#)

Important

AWS mendukung berbagi AMI secara pribadi dengan AWS akun lain dengan memodifikasi izin AMI. Jika AMI dibuat pribadi tanpa dibagikan, ini dapat mengakibatkan kesalahan otorisasi saat meluncurkan instance baru. Untuk informasi selengkapnya tentang berbagi AMI pribadi, lihat [Berbagi AMI dengan AWS akun tertentu](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

ID AMI <ID of your AMI> tidak ada. Peluncuran instance EC2 gagal.

- Penyebab: AMI mungkin telah dihapus setelah membuat template peluncuran atau konfigurasi peluncuran.
- Solusi:
 1. Buat template peluncuran baru atau konfigurasi peluncuran menggunakan AMI yang valid.
 2. Perbarui grup Auto Scaling Anda dengan templat peluncuran baru atau konfigurasi peluncuran menggunakan perintah. [update-auto-scaling-group](#)

AMI <AMI ID> tertunda, dan tidak dapat dijalankan. Peluncuran instance EC2 gagal.

Penyebab: Anda mungkin baru saja membuat AMI Anda (dengan mengambil snapshot dari instance yang sedang berjalan atau cara lain), dan itu mungkin belum tersedia.

Solusi: Anda harus menunggu AMI Anda tersedia dan kemudian membuat template peluncuran atau konfigurasi peluncuran Anda.

<device name>Nama perangkat tidak valid. Peluncuran instance EC2 gagal.

Penyebab: Saat melampirkan volume EBS ke instans EC2, Anda harus memberikan nama perangkat yang valid untuk volume tersebut. AMI yang dipilih harus mendukung nama perangkat ini.

Solusi:

1. Buat templat peluncuran baru atau konfigurasi peluncuran dan tentukan nama perangkat yang benar untuk AMI Anda. Konvensi penamaan yang direkomendasikan bervariasi berdasarkan jenis virtualisasi AMI. Untuk informasi selengkapnya, lihat [Nama perangkat](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
2. Perbarui grup Auto Scaling Anda dengan templat peluncuran baru atau konfigurasi peluncuran menggunakan perintah. [update-auto-scaling-group](#)

Arsitektur 'arm64' dari jenis instance yang ditentukan tidak cocok dengan arsitektur 'x86_64' dari AMI yang ditentukan... Peluncuran instans EC2 gagal.

Penyebab 1: Jika arsitektur AMI dan jenis instans yang digunakan dalam template peluncuran atau konfigurasi peluncuran tidak sama, Anda akan mendapatkan error saat Amazon EC2 Auto Scaling mencoba meluncurkan instance menggunakan konfigurasi instans yang tidak kompatibel.

Solusi 1:

1. Verifikasi arsitektur AMI Anda menggunakan [perintah deskripsikan gambar](#) atau dari konsol Amazon EC2 dengan memeriksa nilai Arsitektur pada panel detail halaman Amazon Machine Images (AMI).
2. Temukan jenis instans yang memiliki arsitektur yang sama dengan AMI Anda menggunakan [describe-instance-types](#) perintah atau dari konsol Amazon EC2 dengan mencentang kolom Arsitektur di layar Jenis instans. Untuk informasi selengkapnya tentang memilih jenis instans yang kompatibel, lihat [Kompatibilitas untuk mengubah jenis instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
3. Buat template peluncuran baru atau konfigurasi peluncuran menggunakan tipe instans yang memiliki arsitektur yang sama dengan AMI Anda.
4. Perbarui grup Auto Scaling Anda dengan templat peluncuran baru atau konfigurasi peluncuran menggunakan perintah. [update-auto-scaling-group](#)

Penyebab 2: Auto Scaling Amazon EC2 mencoba meluncurkan jenis instans yang ditentukan dalam kebijakan instans campuran untuk grup Auto Scaling Anda, tetapi jenis instans tidak memiliki arsitektur yang sama dengan AMI yang ditentukan dalam template peluncuran Anda.

Solusi 1: Jangan sertakan tipe instance yang memiliki arsitektur berbeda dalam kebijakan instance campuran Anda.

1. Verifikasi arsitektur AMI Anda menggunakan [perintah deskripsikan gambar](#) atau dari konsol Amazon EC2 dengan memeriksa nilai Arsitektur pada panel detail halaman Amazon Machine Images (AMI).
2. Verifikasi arsitektur setiap jenis instans yang ingin Anda sertakan dalam kebijakan instance campuran menggunakan [describe-instance-types](#) perintah atau dari konsol Amazon EC2 dengan mencentang kolom Arsitektur di layar Jenis instans. Untuk informasi selengkapnya tentang memilih

jenis instans yang kompatibel, lihat [Kompatibilitas untuk mengubah jenis instans](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

3. Perbarui atau hapus jenis instans yang tidak kompatibel dari grup Auto Scaling menggunakan [update-auto-scaling-group](#) perintah.

Solusi 2: Untuk meluncurkan instans Arm (Graviton2) dan x86_64 (Intel) di grup Auto Scaling yang sama, Anda harus menggunakan templat peluncuran yang didukung oleh AMI yang kompatibel dengan ARM dan AMI yang kompatibel dengan Intel x86, masing-masing, agar sesuai dengan jenis instans dalam kebijakan instans campuran Anda.

1. Verifikasi arsitektur AMI di template peluncuran yang ada menggunakan [perintah deskripsikan gambar](#) atau dari konsol Amazon EC2 dengan memeriksa nilai Arsitektur pada panel detail halaman Amazon Machine Images (AMI).
2. Buat template peluncuran baru menggunakan AMI yang cocok dengan arsitektur lain yang ingin Anda gunakan.
3. Perbarui grup Auto Scaling Anda untuk mengganti template peluncuran yang ada dan tentukan template peluncuran baru untuk setiap jenis instance yang kompatibel menggunakan perintah. [update-auto-scaling-group](#) Untuk informasi selengkapnya, lihat [Gunakan template peluncuran yang berbeda untuk jenis instans](#).

AMI '<AMI ID>' dinonaktifkan, dan tidak dapat dijalankan. Peluncuran instance EC2 gagal.

Penyebab: Anda mencoba meluncurkan instance dari AMI yang telah dinonaktifkan. Untuk informasi selengkapnya, lihat [Menonaktifkan AMI](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Solusi:

1. Buat template peluncuran baru atau konfigurasi peluncuran dan tentukan AMI yang tidak dinonaktifkan.
2. Perbarui grup Auto Scaling Anda dengan templat peluncuran baru atau konfigurasi peluncuran menggunakan perintah. [update-auto-scaling-group](#)

Memecahkan Masalah Auto Scaling Amazon EC2: Masalah penyeimbang beban

Halaman ini memberikan informasi tentang masalah yang disebabkan oleh load balancer yang terkait dengan grup Auto Scaling, penyebab potensial, dan langkah-langkah yang dapat Anda ambil untuk memecahkan masalah tersebut.

Untuk mengambil data pesan kesalahan, lihat [Mengambil pesan kesalahan dari aktivitas penskalaan](#).

Ketika instance EC2 Anda gagal diluncurkan karena masalah dengan load balancer yang terkait dengan grup Auto Scaling Anda, Anda mungkin mendapatkan satu atau beberapa pesan kesalahan berikut.

Masalah load balancer

- [Satu atau lebih kelompok sasaran tidak ditemukan. Validasi konfigurasi load balancer gagal.](#)
- [<your load balancer>Tidak dapat menemukan Load Balancer. Validasi konfigurasi load balancer gagal.](#)
- [Tidak ada Load Balancer AKTIF yang bernama <load balancer name>. Pembaruan konfigurasi load balancer gagal.](#)
- [Instance EC2 <instance ID> tidak ada dalam VPC. Pembaruan konfigurasi load balancer gagal.](#)

Note

Anda dapat menggunakan Reachability Analyzer untuk memecahkan masalah konektivitas dengan memeriksa apakah instance di grup Auto Scaling dapat dijangkau melalui penyeimbang beban. Untuk mempelajari berbagai masalah kesalahan konfigurasi jaringan yang terdeteksi secara otomatis oleh Reachability Analyzer, lihat kode penjelasan Reachability Analyzer di Panduan [Pengguna Reachability Analyzer](#).

Satu atau lebih kelompok sasaran tidak ditemukan. Validasi konfigurasi load balancer gagal.

Masalah: Saat grup Auto Scaling meluncurkan instans, Amazon EC2 Auto Scaling mencoba memvalidasi bahwa sumber daya Elastic Load Balancing yang terkait dengan grup Auto Scaling ada. Ketika grup target tidak dapat ditemukan, aktivitas penskalaan gagal, dan Anda mendapatkan One

or more target groups not found. Validating load balancer configuration failed. kesalahan.

Penyebab 1: Grup target yang dilampirkan ke grup Auto Scaling Anda telah dihapus.

Solusi 1: [Anda dapat membuat grup Auto Scaling baru tanpa grup target atau menghapus grup target yang tidak digunakan dari grup Auto Scaling menggunakan konsol Auto Scaling Amazon EC2 atau perintah `groups.detach-load-balancer-target`](#)

Penyebab 2: Grup target ada, tetapi ada masalah saat mencoba menentukan ARN grup target saat membuat grup Auto Scaling. Sumber daya tidak dibuat dalam urutan yang benar.

Solusi 2: Buat grup Auto Scaling baru dan tentukan grup target di bagian akhir.

<your load balancer>Tidak dapat menemukan Load Balancer. Validasi konfigurasi load balancer gagal.

Masalah: Saat grup Auto Scaling meluncurkan instans, Amazon EC2 Auto Scaling mencoba memvalidasi bahwa sumber daya Elastic Load Balancing yang terkait dengan grup Auto Scaling ada. Jika Classic Load Balancer tidak dapat ditemukan, aktivitas penskalaan gagal, dan Anda mendapatkan kesalahan. `Cannot find Load Balancer <your load balancer>. Validating load balancer configuration failed.`

Penyebab 1: Classic Load Balancer telah dihapus.

Solusi 1: Anda dapat membuat grup Auto Scaling baru tanpa penyeimbang beban atau menghapus penyeimbang beban yang tidak digunakan dari grup Auto Scaling dengan menggunakan konsol Auto Scaling Amazon EC2 atau perintah. [detach-load-balancers](#)

Penyebab 2: Classic Load Balancer ada, tetapi ada masalah saat mencoba menentukan nama penyeimbang beban saat membuat grup Auto Scaling. Sumber daya tidak dibuat dalam urutan yang benar.

Solusi 2: Buat grup Auto Scaling baru dan tentukan nama penyeimbang beban di bagian akhir.

Tidak ada Load Balancer AKTIF yang bernama <load balancer name>. Pembaruan konfigurasi load balancer gagal.

Penyebab: Penyeimbang beban yang ditentukan mungkin telah dihapus.

Solusi: Anda dapat membuat penyeimbang beban baru dan kemudian membuat grup Auto Scaling baru atau membuat grup Auto Scaling baru tanpa penyeimbang beban.

Instance EC2 <instance ID> tidak ada dalam VPC. Pembaruan konfigurasi load balancer gagal.

Penyebab: Instance yang ditentukan tidak ada di VPC.

Solusi: Anda dapat menghapus penyeimbang beban yang terkait dengan instance atau membuat grup Auto Scaling baru.

Memecahkan Masalah Auto Scaling Amazon EC2: Luncurkan templat

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat mencoba menentukan templat peluncuran untuk grup Auto Scaling Anda.

Tidak dapat meluncurkan instance

Jika Anda tidak dapat meluncurkan instance apa pun dengan templat peluncuran yang sudah ditentukan, periksa berikut ini untuk pemecahan masalah umum: [Memecahkan Masalah Auto Scaling Amazon EC2: Kegagalan peluncuran instans EC2](#)

Anda harus menggunakan templat peluncuran yang valid sepenuhnya (nilai tidak valid)

Masalah: Saat Anda mencoba menentukan template peluncuran untuk grup Auto Scaling, Anda mendapatkan kesalahan. You must use a valid fully-formed launch template Anda mungkin mengalami kesalahan ini karena nilai dalam template peluncuran hanya divalidasi ketika grup Auto Scaling yang menggunakan template peluncuran dibuat atau diperbarui.

Penyebab 1: Jika Anda menerima You must use a valid fully-formed launch template kesalahan, maka ada masalah yang menyebabkan Auto Scaling Amazon EC2 menganggap sesuatu tentang template peluncuran tidak valid. Ini adalah kesalahan umum yang dapat memiliki beberapa penyebab berbeda.

Solusi 1: Coba langkah-langkah berikut untuk memecahkan masalah:

1. Perhatikan bagian kedua dari pesan kesalahan untuk menemukan informasi lebih lanjut. Setelah `You must use a valid fully-formed launch template` kesalahan, lihat pesan kesalahan yang lebih spesifik yang mengidentifikasi masalah yang perlu Anda atasi.
2. Jika Anda tidak dapat menemukan penyebabnya, uji template peluncuran Anda dengan perintah [run-instance](#). Gunakan `--dry-run` opsi, seperti yang ditunjukkan pada contoh berikut. Ini memungkinkan Anda mereproduksi masalah dan dapat memberikan wawasan tentang penyebabnya.

```
aws ec2 run-instances --launch-template LaunchTemplateName=my-template,Version='1' --dry-run
```

3. Jika nilai tidak valid, verifikasi bahwa sumber daya yang ditentukan ada dan itu benar. Misalnya, saat Anda menentukan key pair Amazon EC2, sumber daya harus ada di akun Anda dan di Wilayah tempat Anda membuat atau memperbarui grup Auto Scaling Anda.
4. Jika informasi yang diharapkan tidak ada, verifikasi pengaturan Anda dan sesuaikan templat peluncuran sesuai kebutuhan.
5. Setelah membuat perubahan, jalankan kembali perintah [run-instance](#) dengan `--dry-run` opsi untuk memverifikasi bahwa template peluncuran Anda menggunakan nilai yang valid.

Untuk informasi selengkapnya, lihat [Membuat template peluncuran untuk grup Auto Scaling](#).

Anda tidak diizinkan untuk menggunakan template peluncuran (izin tidak memadai)

Masalah: Saat Anda mencoba menentukan template peluncuran untuk grup Auto Scaling, Anda mendapatkan kesalahan. `You are not authorized to use launch template`

Penyebab 1: Jika Anda mencoba menggunakan template peluncuran, dan kredensial IAM yang Anda gunakan tidak memiliki izin yang memadai, Anda menerima kesalahan bahwa Anda tidak diizinkan untuk menggunakan template peluncuran.

Solusi 1: Untuk mengatasi masalah ini, coba yang berikut ini:

- Verifikasi bahwa kredensial IAM yang Anda gunakan untuk membuat permintaan memiliki izin untuk memanggil tindakan API EC2 yang Anda butuhkan, termasuk tindakan. `ec2:RunInstances` Jika Anda menentukan tag apa pun di template peluncuran, Anda juga harus memiliki izin untuk menggunakan `ec2:CreateTags` tindakan tersebut.

- Atau, verifikasi bahwa kredensial IAM yang Anda gunakan untuk membuat permintaan ditetapkan kebijakan. `AmazonEC2FullAccess` Kebijakan AWS terkelola ini memberikan akses penuh ke semua sumber daya Amazon EC2 dan layanan terkait, termasuk Auto Scaling Amazon EC2, CloudWatch dan Elastic Load Balancing.

Untuk informasi selengkapnya tentang izin yang diperlukan untuk menggunakan templat peluncuran, termasuk contoh kebijakan IAM, lihat [Mengontrol akses untuk meluncurkan templat dengan izin IAM](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux. Untuk contoh kebijakan IAM lainnya, lihat [Luncurkan dukungan templat](#).

Penyebab 2: Jika Anda mencoba menggunakan template peluncuran yang menentukan profil instance, Anda harus memiliki izin IAM untuk meneruskan peran IAM yang terkait dengan profil instance.

Solusi 2: Verifikasi bahwa kredensial IAM yang Anda gunakan untuk membuat permintaan memiliki `iam:PassRole` izin yang benar untuk meneruskan peran yang ditentukan ke layanan Auto Scaling Amazon EC2. Untuk informasi selengkapnya dan contoh kebijakan IAM, lihat [Peran IAM untuk aplikasi yang berjalan di instance Amazon EC2](#). Untuk topik pemecahan masalah lebih lanjut terkait profil contoh, lihat [Pemecahan Masalah Amazon EC2 dan IAM](#) di Panduan Pengguna IAM.

Penyebab 3: Jika Anda mencoba menggunakan templat peluncuran yang menentukan AMI di tempat lain Akun AWS, dan AMI bersifat pribadi dan tidak dibagikan dengan yang Akun AWS Anda gunakan, Anda menerima kesalahan bahwa Anda tidak diizinkan untuk menggunakan templat peluncuran.

Solusi 3: Verifikasi bahwa izin pada AMI menyertakan akun yang Anda gunakan. Untuk informasi selengkapnya, lihat [Berbagi AMI dengan spesifik Akun AWS](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Memecahkan Masalah Auto Scaling Amazon EC2: Pemeriksaan Kesehatan

Halaman ini memberikan informasi tentang instans EC2 Anda yang berakhir karena pemeriksaan kesehatan. Ini menjelaskan penyebab potensial, dan langkah-langkah yang dapat Anda ambil untuk menyelesaikan masalah.

Untuk mengambil data pesan kesalahan, lihat [Mengambil pesan kesalahan dari aktivitas penskalaan](#).

Masalah pemeriksaan kesehatan

- [Sebuah instance dikeluarkan dari layanan sebagai tanggapan atas kegagalan pemeriksaan status instans EC2](#)
- [Sebuah instance dikeluarkan dari layanan sebagai tanggapan terhadap reboot terjadwal EC2](#)
- [Sebuah contoh dikeluarkan dari layanan sebagai tanggapan terhadap pemeriksaan kesehatan EC2 yang mengindikasikan telah dihentikan atau dihentikan](#)
- [Sebuah contoh diambil dari layanan sebagai tanggapan atas kegagalan pemeriksaan kesehatan sistem ELB](#)

Note

Anda dapat diberi tahu saat Auto Scaling Amazon EC2 menghentikan instans di grup Auto Scaling, termasuk saat penyebab penghentian instans bukan akibat aktivitas penskalaan. Untuk informasi selengkapnya, lihat [Opsis pemberitahuan Amazon SNS untuk Auto Scaling Amazon EC2](#).

Bagian berikut menjelaskan kesalahan pemeriksaan kesehatan yang paling umum dan penyebab yang akan Anda temui. Jika Anda memiliki masalah lain, lihat artikel Pusat AWS Pengetahuan berikut untuk bantuan pemecahan masalah tambahan:

- [Mengapa Auto Scaling Amazon EC2 menghentikan instans?](#)
- [Mengapa Auto Scaling Amazon EC2 tidak menghentikan instans yang tidak sehat?](#)

Sebuah instance dikeluarkan dari layanan sebagai tanggapan atas kegagalan pemeriksaan status instans EC2

Masalah: Instans Auto Scaling gagal dalam pemeriksaan status Amazon EC2.

Penyebab 1: Jika ada masalah yang menyebabkan Amazon EC2 menganggap instans di grup Auto Scaling Anda terganggu, Auto Scaling Amazon EC2 secara otomatis mengganti instans yang mengalami gangguan sebagai bagian dari pemeriksaan kesehatannya. Pemeriksaan status dibangun di Amazon EC2 sehingga pemeriksaan tidak dapat dinonaktifkan atau dihapus. Ketika pemeriksaan status instans gagal, Anda biasanya harus mengatasi masalah sendiri dengan membuat perubahan konfigurasi instance sampai aplikasi Anda tidak lagi menunjukkan masalah apa pun.

Solusi 1: Untuk mengatasi masalah ini, ikuti langkah-langkah berikut:

1. Buat instans Amazon EC2 secara manual yang bukan bagian dari grup Auto Scaling dan selidiki masalahnya. Untuk bantuan umum dalam menyelidiki instans yang mengalami gangguan, lihat [Memecahkan masalah instans dengan pemeriksaan status gagal di](#) Panduan Pengguna Amazon EC2 untuk Instans Linux dan Pemecahan Masalah Instans Windows di Panduan Pengguna Amazon EC2 untuk [Instans Windows](#).
2. Setelah mengonfirmasi bahwa instans berhasil diluncurkan dan sehat, terapkan konfigurasi instans baru yang bebas kesalahan ke grup Auto Scaling.
3. Hapus instans yang Anda buat untuk menghindari tagihan yang sedang berlangsung ke AWS akun Anda.

Penyebab 2: Ada ketidakcocokan antara masa tenggang pemeriksaan kesehatan dan waktu startup instance.

Solusi 2: Edit masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling Anda ke periode waktu yang sesuai untuk aplikasi Anda. Instans yang diluncurkan dalam grup Auto Scaling memerlukan waktu pemanasan yang cukup (masa tenggang) untuk mencegah penghentian dini karena penggantian pemeriksaan kesehatan. Untuk informasi selengkapnya, lihat [Mengatur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling](#).

Sebuah instance dikeluarkan dari layanan sebagai tanggapan terhadap reboot terjadwal EC2

Masalah: Instans Auto Scaling diganti ketika acara terjadwal menunjukkan masalah dengan instance.

Penyebab: Auto Scaling Amazon EC2 menggantikan instans dengan pemeliharaan terjadwal atau acara pensiun di masa mendatang.

Solusi: Peristiwa ini tidak sering terjadi. Jika Anda memerlukan sesuatu untuk terjadi pada instance yang sedang dihentikan, atau pada instance yang dimulai, Anda dapat menggunakan kait siklus hidup. Hook ini memungkinkan Anda melakukan tindakan kustom saat Amazon EC2 Auto Scaling meluncurkan atau menghentikan instans. Untuk informasi selengkapnya, lihat [Hook siklus aktif Amazon EC2 Auto Scaling](#).

Jika Anda tidak ingin instance diganti karena acara yang dijadwalkan, Anda dapat menanggukhan proses pemeriksaan kesehatan untuk grup Auto Scaling. Untuk informasi selengkapnya, lihat [Menanggukhan dan melanjutkan proses Auto Scaling Amazon EC2](#).

Sebuah contoh dikeluarkan dari layanan sebagai tanggapan terhadap pemeriksaan kesehatan EC2 yang mengindikasikan telah dihentikan atau dihentikan

Masalah: Instans Auto Scaling yang telah dihentikan, reboot, atau dihentikan diganti.

Penyebab 1: Pengguna secara manual menghentikan, me-reboot, atau menghentikan instance.

Solusi 1: Jika pemeriksaan kesehatan gagal karena pengguna menghentikan, me-reboot, atau menghentikan instans secara manual, ini disebabkan oleh cara kerja pemeriksaan kesehatan Auto Scaling Amazon EC2. Contohnya harus sehat dan dapat dijangkau. Jika Anda perlu me-reboot instans di grup Auto Scaling Anda, sebaiknya Anda menempatkan instans dalam keadaan siaga terlebih dahulu. Untuk informasi selengkapnya, lihat [Hapus instans sementara dari grup Auto Scaling](#).

Perhatikan bahwa ketika Anda menghentikan instans secara manual, kait siklus hidup terminasi dan deregistrasi Elastic Load Balancing (dan pengurusan koneksi) harus diselesaikan sebelum instans benar-benar dihentikan.

Penyebab 2: Auto Scaling Amazon EC2 mencoba mengganti Instans Spot setelah layanan Spot Amazon EC2 mengganggu instans, karena harga Spot naik di atas harga maksimum atau kapasitas Anda tidak lagi tersedia.

Solusi 2: Tidak ada jaminan bahwa Instans Spot ada untuk memenuhi permintaan pada titik waktu tertentu. Namun, Anda dapat mencoba yang berikut ini:

- Gunakan harga maksimum Spot yang lebih tinggi (mungkin harga On-Demand). Dengan menetapkan harga maksimum Anda lebih tinggi, ini memberi layanan Amazon EC2 Spot kesempatan yang lebih baik untuk meluncurkan dan mempertahankan jumlah kapasitas yang Anda butuhkan.
- Tingkatkan jumlah kumpulan kapasitas berbeda tempat Anda dapat meluncurkan instance dengan menjalankan beberapa jenis instans di beberapa Availability Zone. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#).
- Jika Anda menggunakan beberapa jenis instans, pertimbangkan untuk mengaktifkan fitur Capacity Rebalancing. Ini berguna jika Anda ingin layanan Amazon EC2 Spot mencoba meluncurkan Instans Spot baru sebelum instance yang sedang berjalan dihentikan. Untuk informasi selengkapnya, lihat [Gunakan Rebalancing Kapasitas untuk menangani interupsi Spot Amazon EC2](#).

Penyebab 3: Dengan Blok Kapasitas, Amazon EC2 menghentikan instans apa pun yang masih berjalan 30 menit sebelum waktu akhir Blok Kapasitas. Penghentian mendadak ini menyebabkan grup Auto Scaling Anda mencoba meluncurkan instans baru untuk mempertahankan kapasitas yang diinginkan, bahkan saat Blok Kapasitas berakhir.

Solusi 3: Untuk mengatasi masalah ini, coba yang berikut ini:

- Kurangi kapasitas yang diinginkan dari grup Auto Scaling untuk mencegahnya mencoba meluncurkan instance baru. Untuk informasi selengkapnya, lihat [Skala manual untuk Amazon EC2 Auto Scaling](#).
- Pastikan Anda menskalakan di grup Auto Scaling 30 menit sebelum waktu akhir Blok Kapasitas sehingga Anda tidak sering mengalami kesalahan ini. Pastikan semua kait siklus hidup telah selesai 30 menit sebelum waktu akhir Blok Kapasitas. Untuk informasi selengkapnya, lihat [Gunakan Blok Kapasitas untuk beban kerja pembelajaran mesin](#).

Sebuah contoh diambil dari layanan sebagai tanggapan atas kegagalan pemeriksaan kesehatan sistem ELB

Masalah: Instans Auto Scaling mungkin lolos pemeriksaan status EC2. Tetapi mereka mungkin gagal dalam pemeriksaan kesehatan Elastic Load Balancing untuk kelompok target atau Classic Load Balancer yang dengannya grup Auto Scaling terdaftar.

Penyebab: Jika grup Auto Scaling mengandalkan pemeriksaan kesehatan yang disediakan oleh Elastic Load Balancing, Auto Scaling Amazon EC2 menentukan status kesehatan instans Anda dengan memeriksa hasil pemeriksaan status EC2 dan pemeriksaan kesehatan Elastic Load Balancing. Penyeimbang beban melakukan pemeriksaan kesehatan dengan mengirimkan permintaan ke setiap instance dan menunggu respons yang benar, atau dengan membuat koneksi dengan instance. Sebuah instance mungkin gagal dalam pemeriksaan kesehatan Elastic Load Balancing karena aplikasi yang berjalan pada instance memiliki masalah yang menyebabkan penyeimbang beban mempertimbangkan instance di luar layanan. Untuk informasi selengkapnya, lihat [Health memeriksa instans dalam grup Auto Scaling](#).

Solusi 1: Untuk lulus pemeriksaan kesehatan Elastic Load Balancing:

- Catat kode keberhasilan yang diharapkan penyeimbang beban, dan verifikasi bahwa aplikasi Anda dikonfigurasi dengan benar untuk mengembalikan kode ini pada keberhasilan.
- Verifikasi bahwa grup keamanan untuk penyeimbang beban dan grup Auto Scaling Anda telah dikonfigurasi dengan benar.

- Verifikasi bahwa pengaturan pemeriksaan kesehatan grup target Anda telah dikonfigurasi dengan benar. Anda menentukan pengaturan pemeriksaan kesehatan untuk penyeimbang beban Anda per grup target.
- Pertimbangkan untuk menambahkan hook siklus hidup peluncuran ke grup Auto Scaling untuk memastikan bahwa aplikasi pada instans siap menerima lalu lintas sebelum terdaftar ke penyeimbang beban di akhir hook siklus hidup.
- Atur masa tenggang pemeriksaan kesehatan untuk grup Auto Scaling Anda ke periode waktu yang cukup lama untuk mendukung jumlah pemeriksaan kesehatan yang berhasil berturut-turut yang diperlukan sebelum Elastic Load Balancing menganggap instans yang baru diluncurkan sehat.
- Verifikasi bahwa penyeimbang beban dikonfigurasi di Availability Zone yang sama dengan grup Auto Scaling Anda.

Untuk informasi selengkapnya, lihat topik berikut:

- [Pemeriksaan kesehatan untuk grup target Anda](#) di Panduan Pengguna untuk Penyeimbang Beban Aplikasi
- [Pemeriksaan kesehatan untuk grup target Anda](#) di Panduan Pengguna untuk Network Load Balancers
- [Pemeriksaan kesehatan untuk grup target Anda](#) di Panduan Pengguna untuk Penyeimbang Beban Gateway
- [Konfigurasi pemeriksaan kesehatan untuk Classic Load Balancer Anda](#) di Panduan Pengguna untuk Classic Load Balancer

Solusi 2: Perbarui grup Auto Scaling untuk menonaktifkan pemeriksaan kesehatan Elastic Load Balancing.

Informasi terkait

Sumber daya terkait berikut dapat membantu Anda ketika bekerja dengan layanan ini.

Sumber daya	Deskripsi
Referensi API Auto Scaling Amazon EC2	Dokumentasi untuk setiap operasi API menunjukkan parameter permintaan dan respons XMLserta menyediakan tautan ke topik referensi SDK khusus bahasa.
penskalaan otomatis di Referensi Perintah AWS CLI	Deskripsi AWS CLI perintah yang dapat Anda gunakan untuk bekerja dengan grup Auto Scaling.
AWS Tools for PowerShellCmdlet Referensi	AWSAlat untuk PowerShell memungkinkan Anda melakukan operasi skrip pada AWS sumber daya Anda dari baris PowerShell perintah.
Buat grup Auto Scaling dengan AWS CloudFormation	Sumber daya AWS::AutoScaling::AutoScalingGrup AWS::AutoScaling::AutoScaling memungkinkan Anda membuat, memodelkan, dan mengelola grup Auto Scaling Anda tanpa tindakan manual.
Titik akhir dan kuota Auto Scaling Amazon EC2 di Referensi Umum AWS	Informasi tentang wilayah dan titik akhir Auto Scaling Amazon EC2.
Halaman Produk	Halaman web utama untuk informasi tentang Amazon EC2 Auto Scaling.
AWS re:Post	AWSlayanan tanya jawab terkelola (Tanya Jawab) yang menawarkan jawaban yang bersumber dari banyak orang dan ditinjau oleh ahli untuk pertanyaan teknis Anda.

Sumber daya	Deskripsi
Membuat AMI di Panduan Pengguna Amazon EC2 untuk Instans Linux	Pelajari cara membuat Amazon Machine Image (AMI) dari instance yang disesuaikan.
Cara menyambung ke instans Linux Anda di Panduan Pengguna Amazon EC2 untuk Instans Linux	Pelajari cara terhubung ke instans Linux yang Anda luncurkan.
Cara menyambung ke instans Windows Anda di Panduan Pengguna Amazon EC2 untuk Instans Windows	Pelajari cara menyambung ke instance Windows yang Anda luncurkan.
Membuat alarm penagihan untuk memantau perkiraan AWS tagihan Anda di CloudWatch Panduan Pengguna Amazon	Pelajari cara memantau perkiraan biaya Anda menggunakan CloudWatch.
Panduan Pengguna Penskalaan Otomatis Aplikasi	Pelajari cara mengonfigurasi penskalaan otomatis untuk sumber daya yang dapat diskalakan untuk Amazon Web Services di luar Amazon EC2.

Sumber daya umum berikut tersedia untuk membantu Anda mempelajari lebih lanjut AWS.

- [Kelas dan Lokakarya](#) – Tautan ke kursus khusus dan berbasis peran, selain laboratorium mandiri, untuk membantu mempertajam keterampilan AWS Anda dan mendapatkan pengalaman praktis.
- [Pusat Developer AWS](#) – Jelajahi tutorial, unduh peralatan, dan pelajari tentang acara developer AWS.
- [Alat Developer AWS](#) – Tautan ke alat, SDK, kit alat IDE, dan alat baris perintah developer untuk mengembangkan serta mengelola aplikasi AWS.
- [Memulai Pusat Sumber Daya](#) – Pelajari cara menyiapkan Akun AWS, bergabung dengan komunitas AWS, dan meluncurkan aplikasi pertama Anda.
- [Tutorial Hands-On](#) - Ikuti step-by-step tutorial untuk meluncurkan aplikasi pertama Anda. AWS
- [Laporan Resmi AWS](#) – Tautan ke daftar laporan resmi teknis AWS yang komprehensif, yang mencakup topik seperti arsitektur, keamanan, dan ekonomi serta ditulis oleh Arsitek Solusi AWS atau ahli teknis lainnya.

- [Pusat AWS Support](#) – Hub untuk membuat dan mengelola kasus AWS Support Anda. Juga mencakup tautan ke sumber daya yang bermanfaat lainnya, seperti forum, FAQ teknis, status kondisi layanan, dan AWS Trusted Advisor.
- [AWS Support](#)— Halaman web utama untuk informasi tentang AWS Support, saluran dukungan respons cepat untuk membantu Anda membangun dan menjalankan aplikasi di cloud. one-on-one
- [Hubungi Kami](#) – Titik kontak pusat untuk pertanyaan tentang tandaihan AWS, akun, peristiwa, penyalahgunaan, dan masalah lainnya.
- [Persyaratan Situs AWS](#) – Informasi detail tentang hak cipta dan merek dagang kami; akun, lisensi, dan akses situs Anda; serta topik lainnya.

Riwayat dokumen

Tabel berikut ini menjelaskan tambahan penting untuk dokumentasi Auto Scaling Amazon EC2, mulai Juli 2018. Untuk pemberitahuan tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

Perubahan	Deskripsi	Tanggal
Pembaruan IAM keamanan	Kebijakan AutoScalingServiceRolePolicy terkelola sekarang memberikan izin tambahan ke Amazon EC2 <code>ec2:GetSecurityGroupsForVpc</code> (dan). <code>ec2:GetInstanceTypesFromInstanceRequirements</code>	Februari 29, 2024
Hibernasi kolam hangat didukung dalam tambahan Wilayah AWS	Anda sekarang dapat hibernasi contoh di kolam hangat di dua Wilayah tambahan: AWS GovCloud (AS-Timur) dan AWS GovCloud (AS-Barat). Untuk informasi selengkapnya tentang kolam hangat, lihat Kolam hangat untuk Penskalaan Otomatis Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2 .	Februari 26, 2024
Hibernasi kolam hangat didukung dalam tambahan Wilayah AWS	Anda sekarang dapat hibernasi contoh di kolam hangat di dua Wilayah tambahan: Eropa (Zurich) dan Timur Tengah (UEA). Untuk informasi selengkapnya tentang kolam hangat,	Februari 21, 2024

lihat [Kolam hangat untuk Penskalaan Otomatis Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2](#).

[Support untuk penggunaan parameter lintas akun](#)

Anda sekarang dapat menggunakan AWS Systems Manager parameter yang dibagikan dari yang lain Akun AWS dengan Amazon EC2 Auto Scaling. Untuk informasi selengkapnya, lihat [Menggunakan AWS Systems Manager parameter alih-alih ID AMI di templat peluncuran](#) di Panduan Pengguna Auto Scaling Amazon EC2.

Februari 21, 2024

[Opsi perlindungan harga Spot baru](#)

Sekarang Anda dapat menentukan ambang perlindungan harga untuk Instans Spot sebagai persentase dari harga Sesuai Permintaan saat Anda menggunakan pemilihan jenis instans berbasis atribut. Untuk informasi selengkapnya, lihat [Perlindungan harga](#) di Panduan Pengguna Auto Scaling Amazon EC2.

Januari 29, 2024

[Kebijakan pemeliharaan instans](#)

Sekarang Anda dapat menggunakan kebijakan pemeliharaan instans untuk menentukan apakah instance diluncurkan sebelum atau setelah instance yang ada dihentikan selama peristiwa yang menyebabkan instance Anda diganti, termasuk penyegaran instans. Untuk informasi selengkapnya, lihat [Kebijakan pemeliharaan instans](#) di Panduan Pengguna Auto Scaling Amazon EC2.

15 November 2023

[Blok Kapasitas untuk ML](#)

Sekarang Anda dapat meluncurkan instance ke dalam Blok Kapasitas dengan menentukan ID reservasi Blok Kapasitas saat Anda membuat templat peluncuran. Dengan Blok Kapasitas, Anda dapat memesan instans GPU di masa mendatang untuk mendukung beban kerja machine learning (ML) berdurasi pendek. Untuk informasi selengkapnya, lihat [Menggunakan Blok Kapasitas untuk beban kerja pembelajaran mesin](#) di Panduan Pengguna Auto Scaling Amazon EC2.

31 Oktober 2023

[Fitur penyegaran instans baru](#)

Anda sekarang dapat mengonfigurasi penyegaran instance untuk mengatur statusnya menjadi gagal dan secara opsional memutar kembali ketika mendeteksi bahwa CloudWatch alarm tertentu telah masuk ke status. ALARM Untuk informasi selengkapnya, lihat [Membatalkan perubahan dengan rollback](#) di Panduan Pengguna Auto Scaling Amazon EC2.

31 Juli 2023

[Perubahan panduan](#)

Topik baru tentang peluncuran Instans Sesuai Permintaan ke dalam Reservasi Kapasitas telah ditambahkan ke panduan ini. Untuk informasi selengkapnya, lihat [Menggunakan Reservasi Kapasitas Sesuai Permintaan untuk memesan kapasitas di Availability Zone tertentu](#) di Panduan Pengguna Auto Scaling Amazon EC2.

28 Juli 2023

[Perubahan panduan](#)

Topik baru tentang memigrasi AWS CloudFormation tumpukan Anda dari konfigurasi peluncuran ke templat peluncuran telah ditambahkan ke panduan. Untuk informasi selengkapnya, lihat [Memigrasi AWS CloudFormation tumpukan dari konfigurasi peluncuran untuk meluncurkan templat](#) di Panduan Pengguna Auto Scaling Amazon EC2.

18 April 2023

[Support untuk operasi API baru](#)

Rilis ini menambahkan tiga operasi API baru, `AttachTrafficSources`, `DetachTrafficSources`, dan `DescribeTrafficSources`. Juga, bidang baru `TrafficSources` telah ditambahkan ke hasil `DescribeAutoScalingGroups` operasi. Status aktivitas baru `WaitingForConnectionDrain` telah ditambahkan ke hasil `DescribeScalingActivities` operasi. Auto Scaling Amazon EC2 juga mendukung nilai baru `VPC_LATTICE` untuk bidang `HealthCheckType` `CreateAutoScalingGroup` dan `UpdateAutoScalingGroup` operasi. `DescribeAutoScalingGroups` Untuk informasi selengkapnya, lihat Referensi API [Auto Scaling Amazon EC2](#).

31 Maret 2023

[Dukungan untuk Amazon VPC Lattice](#)

Ini adalah rilis ketersediaan umum dari VPC Lattice untuk Amazon EC2 Auto Scaling. Untuk informasi selengkapnya, lihat [Merutekan lalu lintas ke grup Auto Scaling Anda dengan grup target VPC Lattice di Panduan Pengguna Auto Scaling Amazon EC2.](#)

31 Maret 2023

[Perubahan panduan](#)

Bagian dengan AWS CLI contoh untuk bekerja dengan Elastic Load Balancing sekarang mencakup contoh baru dan yang diperbarui. Untuk informasi selengkapnya, lihat [Contoh untuk bekerja dengan Elastic Load Balancing dengan AWS Command Line Interface \(AWS CLI\) di Panduan Pengguna Auto Scaling Amazon EC2.](#)

31 Maret 2023

[Support untuk penskalaan prediktif dalam tambahan Wilayah AWS](#)

Anda sekarang dapat membuat kebijakan penskalaan prediktif di Wilayah Timur Tengah (UEA) dan AWS GovCloud (AS-Timur). Untuk informasi selengkapnya, lihat [Penskalaan prediktif untuk Auto Scaling Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2.](#)

16 Maret 2023

Fitur penyegaran instans baru

Sekarang Anda dapat memilih untuk menghentikan atau mengabaikan instance saat siaga dan mengganti atau mengabaikan instance yang dilindungi dari skala masuk, daripada menunggu instans tersebut dapat diganti. Anda juga dapat memutar kembali perubahan dari penyegaran instans yang gagal. Sebagai bagian dari pembaruan ini, dokumentasi telah diperluas untuk menyertakan topik untuk mengembalikan penyegaran instans, membatalkan penyegaran instance, dan memahami nilai default untuk parameter penyegaran instans yang dapat dikonfigurasi. Untuk informasi selengkapnya, lihat [Mengganti instans Auto Scaling berdasarkan refresh instans](#) dalam Panduan Pengguna Amazon EC2 Auto Scaling.

Februari 10, 2023

[Support untuk menggunakan AWS Systems Manager parameter untuk ID AMI](#)

Anda sekarang dapat menggunakan parameter Systems Manager alih-alih ID AMI di template peluncuran Anda. Untuk informasi selengkapnya, lihat [Menggunakan AWS Systems Manager parameter alih-alih ID AMI di templat peluncuran](#) di Panduan Pengguna Auto Scaling Amazon EC2.

19 Januari 2023

[Rekomendasi penskalaan prediktif](#)

Anda sekarang bisa mendapatkan rekomendasi untuk mengevaluasi dan memilih kebijakan penskalaan prediktif yang tepat dari konsol Auto Scaling Amazon EC2. Untuk informasi selengkapnya, lihat [Mengevaluasi kebijakan penskalaan prediktif Anda](#) di Panduan Pengguna Auto Scaling Amazon EC2.

18 Januari 2023

[Prakiraan penskalaan prediktif](#)

Prakiraan yang dihasilkan oleh penskalaan prediktif sekarang diperbarui setiap enam jam, bukan setiap hari. Untuk informasi selengkapnya, lihat [Penskalaan prediktif untuk Auto Scaling Amazon EC2 di Panduan Pengguna](#) Auto Scaling Amazon EC2.

Januari 6, 2023

[Support untuk matematika
CloudWatch metrik](#)

Sekarang Anda dapat menggunakan matematika saat membuat kebijakan penskalaan pelacakan target. Dengan matematika metrik, Anda dapat menanyakan beberapa CloudWatch metrik dan menggunakan ekspresi matematika untuk membuat deret waktu baru berdasarkan metrik ini. Untuk informasi selengkapnya, lihat [Membuat kebijakan penskalaan pelacakan target untuk Auto Scaling Amazon EC2 menggunakan matematika metrik di Panduan Pengguna Auto Scaling Amazon EC2](#).

Desember 8, 2022

[Pembaruan terhadap izin
peran yang ditautkan layanan
IAM](#)

AutoScalingService RolePolicy Kebijakan ini sekarang memberikan izin tambahan untuk Auto Scaling Amazon EC2. Untuk informasi selengkapnya, lihat [kebijakan AWS terkelola untuk Auto Scaling Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2](#).

6 Desember 2022

[Strategi alokasi Spot Baru](#)

Anda sekarang dapat menggunakan strategi alokasi harga dan kapasitas yang dioptimalkan untuk meminta Instans Spot dari kumpulan Spot yang paling kecil kemungkinannya untuk terganggu dan memiliki harga serendah mungkin. Untuk informasi selengkapnya, lihat [Strategi alokasi](#) di Panduan Pengguna Auto Scaling Amazon EC2.

10 November 2022

[Support untuk penskalaan prediktif di Wilayah Asia Pasifik \(Jakarta\)](#)

Anda sekarang dapat membuat kebijakan penskalaan prediktif di Wilayah Asia Pasifik (Jakarta). Untuk informasi selengkapnya, lihat [Penskalaan prediktif untuk Auto Scaling Amazon EC2 di Panduan](#) Pengguna Auto Scaling Amazon EC2.

13 Oktober 2022

[Support untuk metrik khusus untuk penskalaan prediktif di konsol](#)

Anda sekarang dapat menggunakan metrik khusus saat membuat kebijakan penskalaan prediktif dari konsol Auto Scaling Amazon EC2. Untuk informasi selengkapnya, lihat [Penskalaan prediktif untuk Auto Scaling Amazon EC2 di Panduan](#) Pengguna Auto Scaling Amazon EC2.

13 Oktober 2022

[CloudWatch pemantauan untuk metrik penskalaan prediktif](#)

Anda sekarang dapat mengakses data pemantauan untuk penskalaan prediktif menggunakan CloudWatch. CloudWatch ini memungkinkan Anda menggunakan matematika metrik untuk membuat deret waktu baru yang menampilkan keakuratan data perkiraan. Untuk informasi selengkapnya, lihat [Memantau metrik penskalaan prediktif dengan CloudWatch di Panduan Pengguna Auto Scaling Amazon EC2](#).

Juli 7, 2022

[Support untuk penskalaan prediktif di Wilayah Asia Pasifik \(Osaka\)](#)

Anda sekarang dapat membuat kebijakan penskalaan prediktif di Wilayah Asia Pasifik (Osaka). Untuk informasi selengkapnya, lihat [Penskalaan prediktif untuk Auto Scaling Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2](#).

6 Juli 2022

[Hibernasi kolam hangat didukung di Wilayah tambahan](#)

Anda sekarang dapat melakukan hibernasi di kolam hangat di empat Wilayah tambahan: Afrika (Cape Town), Asia Pasifik (Jakarta), Asia Pasifik (Osaka), dan Eropa (Milan). Untuk informasi selengkapnya tentang kolam hangat, lihat [Kolam hangat untuk Penskalaan Otomatis Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2](#).

Juli 5, 2022

[Perbarui untuk pemeriksaan kesehatan](#)

Saat melakukan pemeriksaan kesehatan, Auto Scaling Amazon EC2 kini membantu Anda meminimalkan waktu henti yang dapat terjadi karena masalah sementara atau pemeriksaan kesehatan yang salah konfigurasi. Untuk informasi selengkapnya, lihat [Cara Auto Scaling Amazon EC2 meminimalkan waktu henti di Panduan Pengguna Auto Scaling Amazon EC2](#).

Mei 21, 2022

[Pemanasan contoh default](#)

Sekarang Anda dapat menyatukan semua pengaturan pemanasan dan cooldown untuk grup Auto Scaling dan mengoptimalkan kinerja kebijakan penskalaan yang menskalakan secara terus menerus dengan mengaktifkan pemanasan instans default. Untuk informasi selengkapnya, lihat [Menyetel pemanasan instans default untuk grup Auto Scaling di Panduan Pengguna Auto Scaling Amazon EC2](#).

19 April 2022

[Perubahan panduan](#)

Bab baru tentang integrasi dengan AWS layanan lain telah ditambahkan ke panduan ini. Untuk informasi selengkapnya, lihat [AWS layanan yang terintegrasi dengan Auto Scaling Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2](#).

29 Maret 2022

[Pembaruan terhadap izin peran yang ditautkan layanan IAM](#)

AutoScalingService RolePolicy Kebijakan ini sekarang memberikan izin baca tambahan ke Auto Scaling Amazon EC2. Untuk informasi selengkapnya, lihat [kebijakan AWS terkelola untuk Auto Scaling Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2](#).

28 Maret 2022

[Metadata instance menyediakan status siklus hidup target](#)

Anda dapat mengambil status siklus hidup target instans Auto Scaling dari metadata instance. Untuk informasi selengkapnya, lihat [Mengambil status siklus hidup target melalui metadata instans di Panduan Pengguna Auto Scaling Amazon EC2](#).

24 Maret 2022

[Support untuk fungsionalitas kolam hangat baru](#)

Anda sekarang dapat hibernasi instance di kolam hangat untuk menghentikan instance tanpa menghapus konten memori (RAM) mereka. Anda sekarang juga dapat mengembalikan instance ke kolam hangat dalam skala, alih-alih selalu mengakhiri kapasitas instans yang Anda perlukan nanti. Untuk informasi selengkapnya, lihat [Kolam hangat untuk Penskalaan Otomatis Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2](#).

Februari 24, 2022

[Perubahan panduan](#)

Konsol Auto Scaling Amazon EC2 telah diperbarui dengan opsi tambahan untuk membantu Anda memulai penyegaran instans dengan pencocokan lewati diaktifkan dan konfigurasi yang diinginkan ditentukan. Untuk informasi selengkapnya, lihat [Memulai atau membatalkan penyegaran instans \(konsol\)](#) di Panduan Pengguna Auto Scaling Amazon EC2.

3 Februari, 2022

[Metrik khusus untuk kebijakan penskalaan prediktif](#)

Sekarang Anda dapat memilih apakah akan menggunakan metrik khusus saat membuat kebijakan penskalaan prediktif. Anda juga dapat menggunakan matematika metrik untuk menyesuaikan metrik yang Anda sertakan dalam kebijakan lebih lanjut. Untuk informasi selengkapnya, lihat [Konfigurasi kebijakan penskalaan prediktif lanjutan menggunakan metrik](#) khusus.

24 November 2021

[Strategi alokasi On-Demand baru](#)

Sekarang Anda dapat memilih apakah akan meluncurkan Instans Sesuai Permintaan berdasarkan harga (jenis instans dengan harga terendah terlebih dahulu) saat membuat grup Auto Scaling yang menggunakan kebijakan instance campuran. Untuk informasi selengkapnya, lihat [Strategi alokasi](#) di Panduan Pengguna Auto Scaling Amazon EC2.

27 Oktober 2021

[Pemilihan tipe instans berbasis atribut](#)

Auto Scaling Amazon EC2 menambahkan dukungan untuk pemilihan jenis instans berbasis atribut. Alih-alih memilih jenis instance secara manual, Anda dapat mengekspresikan persyaratan instance Anda sebagai sekumpulan atribut, seperti vCPU, memori, dan penyimpanan. Untuk selengkapnya, lihat [Membuat grup Auto Scaling menggunakan pemilihan jenis instans berbasis atribut](#) di Panduan Pengguna Auto Scaling Amazon EC2.

27 Oktober 2021

[Support untuk memfilter grup berdasarkan tag](#)

Sekarang Anda dapat memfilter grup Auto Scaling menggunakan filter tag saat Anda mengambil informasi tentang grup Auto Scaling menggunakan perintah `describe-auto-scaling-groups`. Untuk informasi selengkapnya, lihat [Menggunakan tag untuk memfilter grup Auto Scaling di Panduan Pengguna Auto Scaling Amazon EC2](#).

Oktober 14, 2021

[Perubahan panduan](#)

Konsol Auto Scaling Amazon EC2 telah diperbarui untuk membantu Anda membuat kebijakan penghentian khusus. AWS Lambda Dokumentasi konsol telah direvisi sesuai. Untuk informasi selengkapnya, lihat [Menggunakan kebijakan penghentian yang berbeda \(konsol\)](#).

Oktober 14, 2021

[Support untuk menyalin konfigurasi peluncuran untuk meluncurkan template](#)

Anda sekarang dapat menyalin semua konfigurasi peluncuran di AWS Wilayah ke templat peluncuran baru dari konsol Auto Scaling Amazon EC2. Untuk informasi selengkapnya, lihat [Menyalin konfigurasi peluncuran untuk meluncurkan templat](#) di Panduan Pengguna Auto Scaling Amazon EC2.

9 Agustus 2021

[Memperluas fungsionalitas penyegaran instance](#)

Anda sekarang dapat menyertakan pembaruan, seperti versi baru dari template peluncuran, saat mengganti instance dengan menambahkan konfigurasi yang Anda inginkan ke `start-instance-refresh` perintah. Anda juga dapat melewatkan penggantian instance yang sudah memiliki konfigurasi yang Anda inginkan dengan mengaktifkan pencocokan lewati. Untuk informasi selengkapnya, lihat [Mengganti instans Auto Scaling berdasarkan refresh instans](#) dalam Panduan Pengguna Amazon EC2 Auto Scaling.

5 Agustus 2021

[Support untuk kebijakan penghentian kustom](#)

Anda sekarang dapat membuat kebijakan penghentian kustom dengan AWS Lambda. Untuk informasi selengkapnya, lihat [Membuat kebijakan penghentian kustom dengan Lambda](#). Dokumentasi untuk menentukan kebijakan penghentian telah diperbarui sesuai dengan itu.

29 Juli 2021

Perubahan panduan	Konsol Auto Scaling Amazon EC2 telah diperbarui dan disempurnakan dengan fitur tambahan untuk membantu Anda membuat tindakan terjadwal dengan zona waktu yang ditentukan. Dokumentasi untuk penskalaan Terjadwal telah direvisi sesuai.	3 Juni 2021
volume gp3 dalam konfigurasi peluncuran	Anda sekarang dapat menentukan volume gp3 di pemetaan perangkat blok untuk konfigurasi peluncuran.	2 Juni 2021
Support untuk penskalaan prediktif	Sekarang Anda dapat menggunakan penskalaan prediktif untuk secara proaktif menskalakan grup Auto Scaling Amazon EC2 Anda menggunakan kebijakan penskalaan. Untuk informasi selengkapnya, lihat Penskalaan prediktif untuk Auto Scaling Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2 . Dengan pembaruan ini, kebijakan AutoScalingServiceRolePolicy terkelola sekarang menyertakan izin untuk memanggil tindakan <code>cloudwatch:GetMetricData</code> API.	19 Mei 2021

[Perubahan panduan](#)

Anda sekarang dapat mengakses contoh template untuk kait siklus hidup dari GitHub Untuk informasi lebih lanjut, lihat [Kait siklus hidup Amazon EC2 Auto Scaling](#) di Panduan Pengguna Amazon EC2 Auto Scaling.

9 April 2021

[Support untuk kolam hangat](#)

Sekarang Anda dapat menyeimbangkan kinerja (meminimalkan start dingin) dan biaya (menghentikan kapasitas instans penyediaan berlebih) untuk aplikasi dengan waktu boot pertama yang lama dengan menambahkan kolam hangat ke grup Auto Scaling. Untuk informasi selengkapnya, lihat [Kolam hangat untuk Penskalaan Otomatis Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2](#).

8 April 2021

[Support untuk pos pemeriksaan](#)

Sekarang Anda dapat menambahkan pos pemeriksaan ke penyegaran instans untuk mengganti instance secara bertahap dan melakukan verifikasi pada instans Anda di titik tertentu. Untuk informasi selengkapnya, lihat [Menambahkan pos pemeriksaan ke penyegaran instans](#) di Panduan Pengguna Auto Scaling Amazon EC2.

18 Maret 2021

[Perubahan panduan](#)

Dokumentasi yang disempurnakan akan digunakan EventBridge dengan peristiwa Auto Scaling Amazon EC2 dan kait siklus hidup. Untuk informasi selengkapnya, lihat [Menggunakan Auto Scaling Amazon EC2 EventBridge dengan dan Tutorial: Mengonfigurasi hook siklus hidup yang memanggil fungsi Lambda](#) di Panduan Pengguna Auto Scaling Amazon EC2.

18 Maret 2021

[Support untuk zona waktu lokal](#)

Anda sekarang dapat membuat tindakan terjadwal berulang di zona waktu lokal dengan menambahkan --time-zone opsi ke put-scheduled-update-group-action perintah. Jika zona waktu Anda mengamati Daylight Saving Time (DST), secara otomatis tindakan berulang menyesuaikan Daylight Saving Time. Untuk informasi selengkapnya, lihat [Penskalaan terjadwal](#) di Panduan Pengguna Auto Scaling Amazon EC2.

9 Maret 2021

[Memperluas fungsionalitas untuk kebijakan instance campuran](#)

Sekarang Anda dapat memprioritaskan jenis instans untuk kapasitas Spot saat menggunakan kebijakan instans campuran. Auto Scaling Amazon EC2 berupaya memenuhi prioritas dengan upaya terbaik tetapi mengoptimalkan kapasitas terlebih dahulu. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa tipe instans dan opsi pembelian](#) dalam Panduan Pengguna Amazon EC2 Auto Scaling.

8 Maret 2021

[Aktivitas penskalaan untuk grup yang dihapus](#)

Sekarang Anda dapat melihat aktivitas penskalaan untuk grup Auto Scaling yang dihapus dengan menambahkan opsi `--include-deleted-groups` ke `describe-scaling-activities` perintah. Untuk informasi selengkapnya, lihat [Memecahkan Masalah Penskalaan Otomatis Amazon EC2 di Panduan Pengguna Auto Scaling](#) Amazon EC2.

23 Februari 2021

[Perbaiki konsol](#)

Anda sekarang dapat membuat dan melampirkan Application Load Balancer atau Network Load Balancer dari konsol Amazon EC2 Auto Scaling. Untuk informasi selengkapnya, lihat [Membuat dan melampirkan Application Load Balancer atau Network Load Balancer \(konsol\) baru](#) di Panduan Pengguna Auto Scaling Amazon EC2.

24 November 2020

[Beberapa antarmuka jaringan](#)

Sekarang Anda dapat mengonfigurasi template peluncuran untuk grup Auto Scaling yang menentukan beberapa antarmuka jaringan. Untuk informasi selengkapnya, lihat [Antarmuka jaringan di VPC](#).

23 November 2020

Beberapa templat peluncuran	Beberapa template peluncuran sekarang dapat digunakan dengan grup Auto Scaling. Untuk informasi lebih lanjut, lihat Penentuan berbagai templat peluncuran untuk tipe instans di Panduan Pengguna Amazon EC2 Auto Scaling.	19 November 2020
Penyeimbang Beban Gateway	Panduan yang diperbarui untuk menunjukkan cara melampirkan Gateway Load Balancer ke grup Auto Scaling untuk memastikan bahwa instans peralatan yang diluncurkan oleh Amazon EC2 Auto Scaling didaftarkan secara otomatis dan dikeluarkan dari load balancer. Untuk informasi selengkapnya, lihat Jenis Elastic Load Balancing dan Memasang load balancer ke grup Auto Scaling Anda dalam Panduan Pengguna Auto Scaling Amazon EC2.	10 November 2020
Masa pakai instance maksimum	Sekarang Anda dapat mengurangi masa aktif instans maksimum menjadi satu hari (86.400 detik). Untuk informasi selengkapnya, lihat Mengganti instans Auto Scaling berdasarkan masa aktif instans maksimum dalam Panduan Pengguna Auto Scaling Amazon EC2.	9 November 2020

[Penyeimbangan Ulang Kapasitas](#)

Anda dapat mengonfigurasi grup Auto Scaling Anda untuk meluncurkan Spot Instance pengganti saat Amazon EC2 mengeluarkan rekomendasi penyeimbangan ulang. Untuk informasi selengkapnya, lihat [Amazon EC2 Auto Scaling Capacity Rebalancing](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

4 November 2020

[Layanan metadata instans versi 2](#)

Anda dapat mewajibkan penggunaan Layanan Metadata Instans Versi 2, yang merupakan metode yang berorientasi pada sesi untuk meminta metadata instans, saat menggunakan konfigurasi peluncuran. Untuk informasi selengkapnya, lihat [Mengonfigurasi opsi metadata instans](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

28 Juli 2020

[Perubahan panduan](#)

Berbagai peningkatan dan prosedur konsol baru dalam bagian [Mengendalikan instans Auto Scaling mana yang berakhir selama penyempitan](#), [Memantau instans dan grup Auto Scaling Anda](#), [Templat peluncuran](#), dan [Konfigurasi peluncuran](#) dari Panduan Pengguna Auto Scaling Amazon EC2.

28 Juli 2020

Penyegaran contoh	Mulai refresh instans untuk memperbarui semua instans dalam grup Auto Scaling Anda saat Anda membuat perubahan konfigurasi. Untuk informasi selengkapnya, lihat Mengganti instans Auto Scaling berdasarkan refresh instans dalam Panduan Pengguna Auto Scaling Amazon EC2.	16 Juni 2020
Perubahan panduan	Berbagai peningkatan dalam bagian Mengganti instans Auto Scaling berdasarkan masa aktif instans maksimum , Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian , Penskalaan berdasarkan Amazon SQS , dan Pemberian tag grup dan instans Auto Scaling dari Panduan Pengguna Auto Scaling Amazon EC2.	6 Mei 2020
Perubahan panduan	Berbagai peningkatan pada dokumentasi IAM. Untuk informasi lebih lanjut, lihat Dukungan templat peluncuran dan Contoh kebijakan berbasis identitas Amazon EC2 Auto Scaling dalam Panduan Pengguna Auto Scaling Amazon EC2.	4 Maret 2020

[Nonaktifkan kebijakan penskalaan](#)

Anda sekarang dapat menonaktifkan dan mengaktifkan kembali kebijakan penskalaan. Fitur ini memungkinkan Anda untuk menonaktifkan sementara kebijakan penskalaan selagi menyimpan detail konfigurasi sehingga Anda dapat mengaktifkan kembali kebijakan tersebut nanti. Untuk informasi selengkapnya, lihat [Menonaktifkan kebijakan penskalaan untuk kelompok Auto Scaling](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

18 Februari 2020

[Tambahkan fungsionalitas notifikasi](#)

Auto Scaling Amazon EC2 kini mengirimkan peristiwa ke grup Auto Scaling AWS Health Dashboard Anda saat grup Auto Scaling tidak dapat ditingkatkan karena grup keamanan atau templat peluncuran yang hilang. Untuk informasi selengkapnya, lihat [AWS Health Dashboard notifikasi untuk Penskalaan Otomatis Amazon EC2 di Panduan Pengguna Auto Scaling Amazon EC2](#).

12 Februari 2020

Perubahan panduan

Berbagai perbaikan dan koreksi dalam bagian [Cara kerja Amazon EC2 Auto Scaling dengan IAM](#), [Contoh kebijakan berbasis identitas Amazon EC2 Auto Scaling](#), [Kebijakan utama CMK yang diwajibkan untuk digunakan dengan volume terenkripsi](#), dan [Memantau instans dan grup Auto Scaling Anda](#) dari Panduan Pengguna Auto Scaling Amazon EC2.

10 Februari 2020

Perubahan panduan

Peningkatan dokumentasi untuk grup Auto Scaling yang menggunakan pembobotan instans. Pelajari cara menggunakan kebijakan penskalaan saat menggunakan “unit kapasitas” untuk mengukur kapasitas yang diinginkan. Untuk informasi selengkapnya, lihat [Cara kerja penskalaan kebijakan](#) dan [Jenis penyesuaian penskalaan](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

6 Februari 2020

Bab “Keamanan” baru

Bab [Keamanan](#) baru di Panduan Pengguna Auto Scaling Amazon EC2 membantu Anda memahami cara menerapkan [model tanggung jawab bersama](#) saat menggunakan Amazon EC2 Auto Scaling. Sebagai bagian dari pembaruan ini, bab panduan pengguna “Mengontrol Akses ke Sumber daya Amazon EC2 Auto Scaling Anda” telah diganti dengan bagian baru yang lebih berguna, [Manajemen identitas dan akses untuk Amazon EC2 Auto Scaling](#).

4 Februari 2020

Rekomendasi untuk jenis misalnya

AWS Compute Optimizer memberikan rekomendasi instans Amazon EC2 untuk membantu Anda meningkatkan kinerja, menghemat uang, atau keduanya. Untuk informasi selengkapnya, lihat [Mendapatkan rekomendasi untuk jenis instans](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

3 Desember 2019

[Host Khusus dan grup sumber daya host](#)

Panduan yang diperbarui untuk menunjukkan cara membuat templat peluncuran yang menentukan grup sumber daya host. Ini memungkinkan Anda membuat grup Auto Scaling dengan templat peluncuran yang menentukan BYOL AMI untuk digunakan di Host Khusus. Untuk informasi selengkapnya, lihat [Membuat templat peluncuran untuk grup Auto Scaling](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

3 Desember 2019

[Dukungan untuk titik akhir VPC Amazon VPC](#)

Sekarang Anda dapat membangun koneksi pribadi antara VPC dan Amazon EC2 Auto Scaling. Untuk informasi selengkapnya, lihat [Amazon EC2 Auto Scaling dan antarmuka titik akhir VPC](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

22 November 2019

[Masa pakai instance maksimum](#)

Anda sekarang dapat mengganti instans secara otomatis dengan menentukan panjang waktu maksimal yang dapat digunakan oleh suatu instans. Jika ada instans yang mendekati batas ini, Amazon EC2 Auto Scaling secara bertahap menggantikannya. Untuk informasi selengkapnya, lihat [Mengganti instans Auto Scaling berdasarkan masa aktif instans maksimum](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

19 November 2019

[Pembobotan contoh](#)

Untuk grup Auto Scaling dengan beberapa jenis instans, Anda kini dapat secara opsional menentukan jumlah unit kapasitas yang dikontribusikan masing-masing jenis instans pada kapasitas grup. Untuk informasi selengkapnya, lihat [Pembobotan instans untuk Amazon EC2 Auto Scaling](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

19 November 2019

Jumlah minimum jenis instans	Anda tidak perlu lagi menentukan jenis instans tambahan untuk grup Instans Spot, Sesuai Permintaan, dan Pesanan. Untuk semua grup Auto Scaling, jumlah minimal sekarang adalah satu jenis instans. Untuk informasi selengkapnya, lihat Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian dalam Panduan Pengguna Auto Scaling Amazon EC2.	16 September 2019
Support untuk strategi alokasi Spot baru	Amazon EC2 Auto Scaling kini mendukung strategi alokasi Spot baru “kapasitas yang dioptimalkan” yang memenuhi permintaan Anda menggunakan kumpulan Spot Instance yang dipilih secara optimal berdasarkan kapasitas Spot yang tersedia. Untuk informasi selengkapnya, lihat Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian dalam Panduan Pengguna Auto Scaling Amazon EC2.	12 Agustus 2019
Perubahan panduan	Peningkatan dokumentasi Amazon EC2 Auto Scaling dalam topik Peran yang terkait dengan layanan dan Kebijakan utama CMK yang diwajibkan untuk digunakan dengan volume terenkripsi .	1 Agustus 2019

[Support untuk peningkatan penandaan](#)

Amazon EC2 Auto Scaling kini menambahkan tag ke instans Amazon EC2 sebagai bagian dari panggilan API yang sama yang meluncurkan instans tersebut. Untuk informasi selengkapnya, lihat [Pemberian tag grup dan instans Auto Scaling](#).

26 Juli 2019

[Perubahan panduan](#)

Peningkatan dokumentasi Amazon EC2 Auto Scaling dalam topik [Menanggulangi dan melanjutkan proses penskalaan](#). Diperbarui [Contoh kebijakan yang dikelola pelanggan](#) untuk menyertakan kebijakan contoh yang memungkinkan pengguna hanya mengirimkan peran tertentu yang terkait layanan tambahan ke Amazon EC2 Auto Scaling.

13 Juni 2019

[Support untuk fitur Amazon EBS baru](#)

Tambahan dukungan untuk fitur EBS Amazon baru dalam topik templat peluncuran. Ubah status enkripsi volume EBS sembari memulihkan dari tangkapan layar. Untuk informasi selengkapnya, lihat [Membuat templat peluncuran untuk grup Auto Scaling](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

13 Mei 2019

[Perubahan panduan](#)

[Dokumentasi Auto Scaling Amazon EC2 yang disempurnakan di bagian berikut: Mengontrol instans Auto Scaling mana yang dihentikan selama penskalaan, grup Auto Scaling, grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian, dan penskalaan Dinamis untuk Auto Scaling Amazon EC2.](#)

12 Maret 2019

[Support untuk menggabungkan tipe instans dan opsi pembelian](#)

Penyediaan dan penskalaan otomatis instans di seluruh opsi pembelian (Instans Spot, Sesuai Permintaan, dan Pesanan) dan jenis instans dalam satu grup Auto Scaling. Untuk informasi selengkapnya, lihat [Grup Auto Scaling dengan beberapa jenis instans dan opsi pembelian](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

13 November 2018

[Topik yang diperbarui untuk penskalaan berdasarkan Amazon SQS](#)

Panduan yang diperbarui untuk menjelaskan bagaimana Anda dapat menggunakan metrik khusus untuk menskalakan grup Auto Scaling sebagai respons terhadap perubahan permintaan dari antrean Amazon SQS. Untuk informasi selengkapnya, lihat [Penskalaan berdasarkan Amazon SQS](#) dalam Panduan Pengguna Auto Scaling Amazon EC2.

26 Juli 2018

Tabel berikut ini menjelaskan perubahan penting pada dokumentasi Auto Scaling Amazon EC2 sebelum Juli 2018.

Fitur	Deskripsi	Tanggal rilis
Dukungan untuk kebijakan penskalaan pelacakan target	Atur penskalaan dinamis untuk aplikasi Anda hanya dalam beberapa langkah. Untuk informasi selengkapnya, lihat Kebijakan Penskalaan Pelacakan Target untuk Amazon EC2 Auto Scaling .	12 Juli 2017
Dukungan untuk izin tingkat sumber daya	Buat kebijakan IAM untuk mengontrol akses di tingkat sumber daya. Untuk informasi selengkapnya, lihat Mengontrol akses ke sumber daya Amazon EC2 Auto Scaling Anda .	15 Mei 2017
Peningkatan pemantauan	Metrik grup Auto Scaling tidak lagi mewajibkan Anda untuk mengaktifkan pemantauan terperinci. Sekarang Anda dapat mengaktifkan pengumpulan metrik grup dan melihat grafik metrik dari tab Pemantauan di konsol. Untuk informasi selengkapnya, lihat Memantau grup dan instans Auto Scaling menggunakan Amazon CloudWatch .	18 Agustus 2016

Fitur	Deskripsi	Tanggal rilis
Dukungan untuk Application Load Balancers	Lampirkan satu atau lebih kelompok target ke grup Auto Scaling baru atau yang sudah ada. Untuk informasi selengkapnya, lihat Melampirkan load balancer ke grup Auto Scaling Anda .	11 Agustus 2016
Peristiwa untuk pengait siklus hidup	Auto Scaling Amazon EC2 mengirimkan peristiwa EventBridge ke saat memanggil kait siklus hidup. Untuk informasi selengkapnya, lihat Mendapatkan EventBridge saat grup Auto Scaling Anda menskalakan .	24 Februari 2016
Perlindungan instans	Mencegah Amazon EC2 Auto Scaling memilih instans tertentu untuk pengakhiran saat penyempitan. Untuk informasi selengkapnya, lihat Perlindungan instans .	07 Desember 2015
Kebijakan penskalaan langkah	Buat kebijakan penskalaan yang memungkinkan Anda menskalakan berdasarkan ukuran pelanggaran alarm. Untuk informasi selengkapnya, lihat Jenis kebijakan penskalaan .	06 Juli 2015
Perbarui load balancer	Pasang load balancer ke atau lepaskan load balancer dari grup Auto Scaling yang ada. Untuk informasi selengkapnya, lihat Melampirkan load balancer ke grup Auto Scaling Anda .	11 Juni 2015
Support untuk ClassicLink	Tautkan instans EC2-Classic dalam grup Auto Scaling Anda ke VPC, mengaktifkan komunikasi antara instans EC2-Classic tertaut ini dan instans dalam VPC menggunakan alamat IP pribadi. Untuk informasi selengkapnya, lihat Menautkan instans EC2-Classic ke VPC .	19 Januari 2015
Kaitan siklus aktif	Tahan instans yang baru Anda luncurkan atau hentikan dalam status tertunda saat Anda melakukan tindakan pada instans. Untuk informasi selengkapnya, lihat kaitan siklus hidup Auto Scaling Amazon EC2 .	30 Juli 2014

Fitur	Deskripsi	Tanggal rilis
Lepaskan instans	Lepaskan instans dari grup Auto Scaling. Untuk informasi selengkapnya, lihat Lepaskan instans EC2 dari grup Auto Scaling Anda .	30 Juli 2014
Tempatkan instans pada status Siaga	Tempatkan instans yang berada dalam status InService menjadi status Standby. Untuk informasi selengkapnya, lihat Lepaskan sementara instans dari grup Auto Scaling Anda .	30 Juli 2014
Kelola tag	Kelola grup Auto Scaling Anda menggunakan AWS Management Console Untuk informasi selengkapnya, lihat Pemberian tag grup dan instans Auto Scaling .	01 Mei 2014
Dukungan untuk Instans Khusus	Luncurkan Instans Khusus dengan menentukan atribut tenansi penempatan saat Anda membuat konfigurasi peluncuran. Untuk informasi selengkapnya, lihat Tenansi penempatan instans .	23 April 2014
Buat grup atau konfigurasi peluncuran dari instans EC2	Buat grup Auto Scaling atau konfigurasi peluncuran menggunakan instans EC2. Untuk informasi tentang pembuatan konfigurasi peluncuran menggunakan instans EC2, lihat Membuat konfigurasi peluncuran menggunakan instans EC2 . Untuk informasi tentang membuat grup Auto Scaling menggunakan instans EC2, lihat Membuat grup Auto Scaling menggunakan instans EC2 .	02 Januari 2014
Pasang instans	Aktifkan penskalaan otomatis untuk instans EC2 dengan memasang instans pada grup Auto Scaling yang sudah ada. Untuk informasi selengkapnya, lihat Memasang instans EC2 ke grup Auto Scaling Anda .	02 Januari 2014
Lihat batas akun	Lihat batas sumber daya Auto Scaling untuk akun Anda. Untuk informasi selengkapnya, lihat Batas Auto Scaling .	02 Januari 2014

Fitur	Deskripsi	Tanggal rilis
Dukungan konsol untuk Amazon EC2 Auto Scaling	Akses Auto Scaling Amazon EC2 menggunakan file. AWS Management Console Untuk informasi selengkapnya, lihat Memulai dengan Amazon EC2 Auto Scaling .	10 Desember 2013
Tetapkan alamat IP publik	Tetapkan alamat IP publik pada instans yang diluncurkan ke VPC. Untuk informasi selengkapnya, lihat Meluncurkan instans Auto Scaling dalam VPC .	19 September 2013
Kebijakan pengakhiran instans	Tentukan kebijakan pengakhiran instans untuk Amazon EC2 Auto Scaling untuk digunakan saat mengakhiri instans EC2. Untuk informasi selengkapnya, lihat Mengendalikan instans Auto Scaling mana yang berakhir selama skala penyempitan .	17 September 2012
Dukungan untuk peran IAM	Luncurkan instans EC2 dengan profil instans IAM. Anda dapat menggunakan fitur ini untuk menetapkan peran IAM ke instans Anda, memungkinkan aplikasi Anda mengakses Amazon Web Services lainnya dengan aman. Untuk informasi selengkapnya, lihat Luncurkan instans Auto Scaling dengan peran IAM .	11 Juni 2012
Dukungan untuk Spot Instance	Luncurkan Instans Spot dengan konfigurasi peluncuran. Untuk informasi selengkapnya, lihat Meminta Instans Spot untuk aplikasi yang toleran terhadap kesalahan dan fleksibel.	7 Juni 2012
Beri tag grup dan instans	Beri tag grup Auto Scaling dan tentukan bahwa tag juga berlaku untuk instans EC2 yang diluncurkan setelah tag dibuat. Untuk informasi selengkapnya, lihat Pemberian tag grup dan instans Auto Scaling .	26 Januari 2012

Fitur	Deskripsi	Tanggal rilis
Dukungan untuk Amazon SNS	<p>Gunakan Amazon SNS untuk menerima pemberitahuan setiap kali Amazon EC2 Auto Scaling meluncurkan atau mengakhiri instans EC2. Untuk informasi selengkapnya, lihat Mendapatkan pemberitahuan SNS saat grup Auto Scaling Anda diskalakan.</p> <p>Amazon EC2 Auto Scaling juga menambahkan fitur baru berikut:</p> <ul style="list-style-type: none"> • Kemampuan untuk mengatur aktivitas penskalaan berulang menggunakan cron syntax. Untuk informasi selengkapnya, lihat operasi API PutScheduledUpdateGroupAction. • Pengaturan konfigurasi baru yang memungkinkan Anda melakukan skala tanpa menambahkan instance yang diluncurkan ke load balancer (LoadBalancer). Untuk informasi selengkapnya, lihat jenis data API ProcessType. • Bendera ForceDelete di operasi DeleteAutoScalingGroup yang meminta Amazon EC2 Auto Scaling untuk menghapus grup Auto Scaling dengan instans yang terkait tanpa menunggu pengakhiran instans terlebih dahulu. Untuk informasi selengkapnya, lihat operasi API DeleteAutoScalingGroup. 	20 Juli 2011
Tindakan penskalaan terjadwal	Menambahkan dukungan untuk tindakan penskalaan terjadwal. Untuk informasi selengkapnya, lihat Penskalaan Terjadwal untuk Amazon EC2 Auto Scaling .	2 Desember 2010
Dukungan untuk Amazon VPC	Dukungan tambahan untuk Amazon VPC. Untuk informasi selengkapnya, lihat Meluncurkan instans Auto Scaling dalam VPC .	2 Desember 2010
Dukungan untuk kluster HPC	Dukungan tambahan untuk kluster komputasi performa tinggi (HPC).	2 Desember 2010

Fitur	Deskripsi	Tanggal rilis
Dukungan untuk pemeriksaan kesehatan	Dukungan tambahan untuk menggunakan pemeriksaan kesehatan Elastic Load Balancing dengan instans EC2 yang dikelola Amazon EC2 Auto Scaling. Untuk informasi selengkapnya, lihat Health memeriksa instans dalam grup Auto Scaling .	2 Desember 2010
Support untuk CloudWatch alarm	Menghapus mekanisme pemicu yang lebih lama dan mendesain ulang Auto Scaling Amazon EC2 untuk menggunakan fitur alarm. CloudWatch Untuk informasi selengkapnya, lihat Penskalaan dinamis untuk Amazon EC2 Auto Scaling .	2 Desember 2010
Menangguhkan dan melanjutkan penskalaan	Dukungan tambahan untuk menangguhkan dan melanjutkan proses penskalaan.	2 Desember 2010
Dukungan untuk IAM	Dukungan tambahan untuk IAM. Untuk informasi selengkapnya, lihat Mengontrol akses ke sumber daya Amazon EC2 Auto Scaling Anda .	2 Desember 2010

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