



Panduan Pengguna

AWS Batch



AWS Batch: 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 AWS Batch?	1
Komponen AWS Batch	1
Tugas	1
Ketentuan Tugas	2
Antrean Tugas	2
Lingkungan Komputasi	2
Memulai	3
Dasbor	3
Antrian pekerjaan tunggal	3
CloudWatch Wawasan Kontainer	4
Log Job	4
Mengatur	6
Mendaftar Akun AWS	6
Membuat pengguna administratif	7
Buat peran IAM untuk lingkungan komputasi dan instance kontainer Anda	8
Membuat pasangan kunci	8
Buat VPC	11
Membuat grup keamanan	11
Instal AWS CLI	13
Memulai	14
Prasyarat	14
Memulai - Amazon EC2	14
Membuat lingkungan komputasi	14
Membuat antrean tugas	19
Buat definisi pekerjaan	20
Buat pekerjaan	24
Tinjau dan buat	24
Memulai - Fargate	24
Membuat lingkungan komputasi	25
Membuat antrean tugas	26
Buat definisi pekerjaan	26
Buat pekerjaan	30
Tinjau dan buat	30
AWS Batch di Amazon EKS	30

Prasyarat	31
Langkah 1: Mempersiapkan cluster Amazon EKS Anda AWS Batch	33
Langkah 2: Membuat lingkungan komputasi Amazon EKS	36
Langkah 3: Buat antrian pekerjaan dan lampirkan lingkungan komputasi	39
Langkah 4: Buat definisi pekerjaan	39
Langkah 5: Kirim pekerjaan	40
(Opsional) Kirim pekerjaan dengan penggantian	41
AWS Batch di Amazon EKS Private Clusters	42
Tugas	55
Mengirimkan pekerjaan	55
Status Job	58
Variabel Lingkungan Tugas	61
Percobaan kembali tugas secara otomatis	62
Dependensi Job	64
Batas waktu Job	64
Lowongan kerja Amazon EKS	66
Memetakan pekerjaan yang sedang berjalan ke pod dan node	66
Cara memetakan pod yang sedang berjalan kembali ke pekerjaannya	67
Tugas array	69
Contoh alur kerja pekerjaan array	72
Tutorial: Menggunakan indeks pekerjaan array	76
Tugas paralel multisimpul	82
Variabel-variabel lingkungan	83
Grup simpul	83
Siklus hidup Job	84
Hitung pertimbangan lingkungan	85
Lowongan kerja GPU	86
Untuk membuat pekerjaan berbasis GPU di sumber daya Amazon EKS	88
Untuk membuat Kubernetes cluster berbasis GPU di Amazon EKS	88
Untuk membuat definisi pekerjaan GPU Amazon EKS	90
Untuk menjalankan pekerjaan GPU di cluster Amazon EKS Anda	91
Cari dan filter AWS Batch pekerjaan	92
Log Job	93
Informasi Job	94
Ketentuan tugas	95
Membuat definisi pekerjaan simpul tunggal	95

Membuat definisi pekerjaan simpul tunggal pada sumber daya Amazon EC2	96
Membuat definisi pekerjaan simpul tunggal pada sumber daya AWS Fargate	102
Membuat definisi pekerjaan simpul tunggal di sumber daya Amazon EKS	108
Membuat ketentuan tugas paralel multisimpul	112
Membuat definisi pekerjaan paralel multi-node pada sumber daya Amazon EC2	112
Membuat definisi pekerjaan menggunakan ContainerProperties	120
Parameter definisi Job untuk ContainerProperties	127
Membuat definisi pekerjaan menggunakan EcsProperties	172
ContainerProperties versus definisi EcsProperties pekerjaan	172
Perubahan umum pada AWS Batch API	173
Definisi pekerjaan multi-kontainer untuk Amazon ECS	174
Definisi pekerjaan multi-kontainer untuk Amazon EKS	175
AWS Batch skenario pekerjaan menggunakan EcsProperties	175
Menggunakan driver log awslogs	181
Opsi driver log awslogs yang tersedia	182
Menentukan konfigurasi log dalam ketentuan tugas Anda	185
Menentukan data sensitif	186
Menggunakan Secrets Manager	187
Menggunakan Systems Manager Parameter Store	195
Otentikasi registri pribadi untuk pekerjaan	199
Izin IAM yang diperlukan untuk otentikasi registri pribadi	200
Menggunakan otentikasi registri pribadi	201
Volume Amazon EFS	202
Pertimbangan volume Amazon EFS	202
Menggunakan titik akses Amazon EFS	203
Menentukan sistem file Amazon EFS dalam ketentuan tugas Anda	204
Contoh ketentuan tugas	207
Menggunakan variabel lingkungan	207
Menggunakan substitusi parameter	208
Menguji fungsionalitas GPU	210
Tugas paralel multisimpul	210
Antrean tugas	212
Membuat antrean tugas	212
Membuat antrian pekerjaan Fargate	212
Membuat antrian pekerjaan Amazon EC2	213
Membuat antrian pekerjaan Amazon EKS	214

Templat antrian tugas	216
Parameter antrian tugas	217
Nama antrian tugas	217
Tindakan batas waktu status antrian pekerjaan	217
Prioritas	218
Kebijakan penjadwalan	218
Status	219
Urutan lingkungan komputasi	219
Tag	220
Penjadwalan Job	221
Bagikan pengidentifikasi	221
Penjadwalan saham yang adil	222
Lingkungan komputasi	224
Lingkungan komputasi terkelola	224
Pertimbangan saat membuat pekerjaan paralel multi-node	227
Lingkungan komputasi tidak terkelola	227
AMI sumber daya komputasi	228
Spesifikasi AMI sumber daya komputasi	230
Membuat AMI sumber daya komputasi	231
Menggunakan AMI beban kerja GPU	235
Penghentian Amazon Linux	240
Luncurkan dukungan templat	241
Data pengguna Amazon EC2 dalam templat peluncuran	243
Membuat lingkungan komputasi	247
Untuk membuat lingkungan komputasi terkelola menggunakan sumber daya AWS	
Fargate	248
Untuk membuat lingkungan komputasi terkelola menggunakan sumber daya EC2	250
Untuk membuat lingkungan komputasi tidak terkelola menggunakan sumber daya EC2	255
Untuk membuat lingkungan komputasi terkelola menggunakan sumber daya Amazon	
EKS	256
Templat lingkungan komputasi	260
Parameter lingkungan komputasi	261
Nama lingkungan komputasi	262
Tipe	262
Status	262
Sumber daya komputasi	263

Konfigurasi Amazon EKS	276
Peran layanan	276
Tag	278
Konfigurasi EC2	278
Strategi alokasi	279
Memperbarui lingkungan komputasi	280
Memperbarui ID AMI	284
Lingkungan komputasi Amazon EKS	285
Pilihan AMI default	285
Versi Kubernetes yang didukung	286
Memperbarui Kubernetes versi lingkungan komputasi	287
Tanggung jawab bersama dari Kubernetes node	287
Menjalankan DaemonSet pada node AWS Batch terkelola	288
Menyesuaikan dengan template peluncuran	289
Manajemen Memori	292
Menyimpan Memori Sistem	293
Melihat Memori Sumber Daya Komputasi	294
Pertimbangan memori dan vCPU untuk Amazon AWS Batch EKS	294
Kebijakan penjadwalan	300
Membuat kebijakan penjadwalan	300
Templat kebijakan penjadwalan	302
Parameter kebijakan penjadwalan	303
Nama kebijakan penjadwalan	303
Kebijakan saham yang adil	303
Tag	306
Lowongan kerja Orchestrate AWS Batch	307
Melihat detail mesin status	307
Mengedit mesin status	308
Menjalankan mesin keadaan	308
AWS Batch di AWS Fargate	309
Kapan saatnya menggunakan Fargate	309
Ketentuan tugas di Fargate	310
Antrean tugas di Fargate	312
Lingkungan komputasi di Fargate	312
AWS Batch di Amazon EKS	314
Elastic Fabric Adapter	317

Kebijakan, izin, dan IAM role	320
Struktur kebijakan	321
Sintaksis kebijakan	321
Tindakan untuk AWS Batch	322
Amazon Resource Names untuk AWS Batch	322
Izin pengujian	323
Izin tingkat sumber daya yang didukung	324
Kunci syarat	336
Contoh kebijakan	337
Akses hanya-baca	338
Membatasi pengguna, citra, hak istimewa, peran	338
Membatasi pengiriman tugas	340
Membatasi antrean tugas	341
Tolak tindakan saat mengkondisikan semua tombol cocok dengan string	341
Tolak tindakan ketika tombol kondisi apa pun cocok dengan string	342
Gunakan tombol <code>batch:ShareIdentifier</code> kondisi	344
Kebijakan terkelola AWS Batch	345
<code>AWSBatchFullAccess</code>	345
Membuat kebijakan IAM	346
Peran instans Amazon ECS	347
Peran armada spot Amazon EC2	350
Buat peran armada spot Amazon EC2 di AWS Management Console	350
Buat Peran Armada Spot Amazon EC2 dengan AWS CLI	352
EventBridge Peran IAM	353
EventBridge	356
AWS Batch Event	357
Peristiwa Perubahan Status Tugas	357
Acara yang diblokir antrean pekerjaan	359
Menggunakan Notifikasi AWS Pengguna dengan AWS Batch	361
AWS Batch pekerjaan sebagai EventBridge target	361
Membuat pekerjaan terjadwal	362
Membuat aturan dengan pola acara	365
Trafo masukan acara	367
Tutorial: Mendengarkan AWS Batch EventBridge	370
Prasyarat	370
Langkah 1: Buat Fungsi Lambda	370

Langkah 2: Daftarkan Aturan Peristiwa	371
Langkah 3: Uji Konfigurasi Anda	373
Tutorial: Mengirim Pemberitahuan Amazon Simple Notification Service untuk Peristiwa Tugas Gagal	374
Prasyarat	374
Langkah 1: Buat dan Berlangganan Topik Amazon SNS	374
Langkah 2: Daftarkan Aturan Peristiwa	374
Langkah 3: Uji Aturan Anda	376
Aturan alternatif: Antrian Pekerjaan Batch Diblokir	377
CloudWatch Log	378
Menambahkan kebijakan IAM CloudWatch Log	378
Instal dan konfigurasi CloudWatch agen	380
Lihat CloudWatch Log	380
Gunakan CloudWatch Log untuk memantau pekerjaan AWS Batch Amazon EKS	383
Prasyarat	383
Instal AWS untuk Fluent Bit	383
Aktifkan Fluent Bit untuk node AWS Batch	383
CloudWatch Wawasan Kontainer	385
Aktifkan Wawasan Kontainer	385
CloudTrail	387
AWS Batch Informasi di CloudTrail	387
Memahami Entri File Berkas Log AWS Batch	388
Membuat cloud pribadi virtual	391
Buat VPC	391
Langkah Berikutnya	392
Keamanan	393
Manajemen Identitas dan Akses	393
Audiens	394
Mengautentikasi dengan identitas	395
Mengelola akses menggunakan kebijakan	398
Bagaimana AWS Batch bekerja dengan IAM	401
IAM role eksekusi	408
Contoh kebijakan berbasis identitas	410
Pencegahan confused deputy lintas layanan	413
Pemecahan Masalah	416
Menggunakan Peran Tertaut Layanan	418

AWS kebijakan terkelola	426
Titik akhir VPC	440
Pertimbangan	441
Membuat sebuah titik akhir antarmuka	442
Membuat kebijakan titik akhir	443
Validasi Kepatuhan	444
Keamanan Infrastruktur	445
Menandai sumber daya Anda	447
Dasar-dasar tanda	447
Penandaan sumber daya Anda	448
Batasan tanda	449
Bekerja dengan tanda menggunakan konsol	450
Menambahkan tanda pada pembuatan sumber daya individu	450
Penambahan dan penghapusan tanda pada sumber daya individu	450
Cara menggunakan tanda dengan menggunakan CLI atau API	451
Kuota Layanan	453
Pemecahan Masalah	454
AWS Batch	455
INVALIDlingkungan komputasi	455
Pekerjaan terjebak dalam RUNNABLE status	457
Instans Spot tidak ditandai pada pembuatan	462
Instans Spot tidak menskalakan ke bawah	463
Tidak dapat mengambil rahasia Secrets Manager	465
Tidak dapat mengesampingkan persyaratan sumber daya definisi pekerjaan	465
Pesan galat saat Anda memperbarui <code>desiredvCpus</code> pengaturan	466
AWS Batch di Amazon EKS	467
INVALIDlingkungan komputasi	467
AWS Batch di Amazon EKS pekerjaan macet dalam RUNNABLE status	470
Verifikasi bahwa <code>aws-auth ConfigMap</code> sudah dikonfigurasi dengan benar	471
Izin atau binding RBAC tidak dikonfigurasi dengan benar	472
Praktik Terbaik	474
Kapan harus menggunakan AWS Batch	474
Daftar periksa untuk dijalankan pada skala	475
Optimalkan wadah dan AMI	476
Pilih sumber daya lingkungan komputasi yang tepat	477
Amazon EC2 Sesuai Permintaan atau Amazon EC2 Spot	478

Gunakan praktik terbaik Amazon EC2 Spot untuk AWS Batch	479
Kesalahan umum dan pemecahan masalah	480
Riwayat dokumen	484
.....	cdxci

Apa Itu AWS Batch?

AWS Batch membantu Anda menjalankan beban kerja komputasi batch di AWS Cloud Komputasi batch adalah cara yang umum bagi developer, ilmuwan, dan insinyur untuk mengakses sejumlah besar sumber daya komputasi, dan AWS Batch menghilangkan pekerjaan berat yang seragam dalam konfigurasi dan pengelolaan infrastruktur yang dibutuhkan, serupa dengan perangkat lunak komputasi batch tradisional. Layanan ini dapat secara efisien menyediakan sumber daya untuk menanggapi tugas yang dikirimkan guna menghilangkan kendala kapasitas, mengurangi biaya komputasi, dan memberikan hasil dengan cepat.

Sebagai layanan terkelola penuh, AWS Batch membantu Anda menjalankan beban kerja komputasi batch dengan skala apa pun. AWS Batch secara otomatis menyediakan sumber daya komputasi dan mengoptimalkan distribusi beban kerja berdasarkan kuantitas dan skala beban kerja. Dengan AWS Batch, Anda tidak perlu menginstal atau mengelola perangkat lunak komputasi batch sehingga Anda dapat memfokuskan waktu untuk menganalisis hasil dan memecahkan masalah.

Topik

- [Komponen AWS Batch](#)
- [Memulai](#)
- [Dasbor](#)

Komponen AWS Batch

AWS Batch menyederhanakan penjalanan tugas batch di beberapa Availability Zone dalam suatu Wilayah. Anda dapat membuat lingkungan komputasi AWS Batch dalam VPC baru atau yang sudah ada. Setelah lingkungan komputasi berjalan dan dikaitkan dengan antrean tugas, Anda dapat menentukan ketentuan tugas yang menentukan gambar kontainer Docker mana yang akan menjalankan tugas Anda. Gambar kontainer disimpan dan ditarik dari registri kontainer, yang mungkin ada di dalam atau di luar infrastruktur AWS.

Tugas

Unit kerja (seperti skrip shell, file Linux yang dapat dijalankan, atau gambar kontainer Docker) yang Anda kirimkan ke AWS Batch. Unit kerja memiliki nama, dan berjalan sebagai aplikasi dalam kontainer di AWS Fargate atau sumber daya Amazon EC2 di lingkungan komputasi Anda, dengan

menggunakan parameter yang Anda tentukan dalam ketentuan tugas. Tugas dapat mereferensikan tugas lain berdasarkan nama atau ID, dan dapat tergantung pada keberhasilan penyelesaian tugas lain. Untuk informasi selengkapnya, lihat [Tugas](#).

Ketentuan Tugas

Ketentuan tugas menentukan cara tugas akan dijalankan. Anda dapat membayangkan ketentuan tugas sebagai cetak biru untuk sumber daya dalam tugas Anda. Anda dapat menyediakan IAM role bagi tugas Anda untuk menyediakan akses ke sumber daya AWS. Anda juga menentukan persyaratan memori dan CPU. Ketentuan tugas juga dapat mengontrol properti kontainer, variabel lingkungan, dan titik pemasangan untuk penyimpanan tetap. Banyak spesifikasi dalam ketentuan tugas dapat diganti dengan menentukan nilai-nilai baru ketika mengirimkan tugas individu. Lihat informasi yang lebih lengkap di [Ketentuan tugas](#)

Antrean Tugas

Ketika mengirimkan tugas AWS Batch, Anda mengirimkannya ke antrean tugas tertentu, tempat tugas berada sampai dijadwalkan ke lingkungan komputasi. Anda mengaitkan satu atau beberapa lingkungan komputasi dengan antrean tugas. Anda juga dapat menetapkan nilai prioritas untuk lingkungan komputasi dan bahkan di antrean tugas itu sendiri. Misalnya, Anda dapat memiliki antrean prioritas tinggi yang menjadi tujuan pengiriman tugas sensitif waktu, dan antrean prioritas rendah untuk tugas yang dapat berjalan kapan saja ketika sumber daya komputasi lebih murah.

Lingkungan Komputasi

Lingkungan komputasi adalah seperangkat sumber daya komputasi terkelola atau tidak terkelola yang digunakan untuk menjalankan tugas. Dengan lingkungan komputasi terkelola, Anda dapat menentukan tipe komputasi yang diinginkan (Fargate atau EC2) pada beberapa tingkat detail. Anda dapat menyiapkan lingkungan komputasi yang menggunakan tipe instans EC2 tertentu, model tertentu seperti `c5.2xlarge` atau `m5.10xlarge`. Sebagai alternatif, Anda dapat memilih untuk menentukan bahwa Anda ingin menggunakan tipe instans terbaru. Anda juga dapat menentukan jumlah vCPUs minimum, yang diinginkan, dan maksimum untuk lingkungan tersebut, bersama dengan jumlah yang bersedia Anda bayar untuk Instans Spot sebagai persentase dari harga Instans Sesuai Permintaan dan set target subnet VPC. AWS Batch Secara efisien meluncurkan, mengelola, dan mengakhiri tipe komputasi sesuai kebutuhan. Anda juga dapat mengelola lingkungan komputasi Anda sendiri. Dengan demikian, Anda bertanggung jawab untuk menyiapkan dan menskalakan instans dalam sebuah kluster Amazon ECS yang AWS Batch buat untuk Anda. Untuk informasi selengkapnya, lihat [Lingkungan komputasi](#).

Memulai

Mulai AWS Batch dengan membuat ketentuan tugas, lingkungan komputasi, dan antrean tugas di konsol AWS Batch.

Wizard perjalanan pertama AWS Batch akan memberi Anda opsi untuk membuat lingkungan komputasi dan antrean tugas serta mengirimkan sampel tugas Hello World. Jika Anda sudah memiliki gambar Docker yang ingin Anda luncurkan di AWS Batch, Anda dapat membuat ketentuan tugas dengan gambar tersebut dan mengirimkannya ke antrean Anda. Untuk informasi selengkapnya, lihat [Memulai dengan AWS Batch](#).

Dasbor

Di AWS Batch dasbor, Anda dapat memantau pekerjaan terbaru, antrian pekerjaan, dan lingkungan komputasi. Secara default, widget dasbor berikut ditampilkan:

- Ikhtisar pekerjaan — Untuk informasi selengkapnya tentang AWS Batch lowongan kerja, lihat [Tugas](#).
- Ikhtisar antrian pekerjaan — Untuk informasi selengkapnya tentang antrian AWS Batch pekerjaan, lihat [Antrean tugas](#)
- Ikhtisar lingkungan komputasi — Untuk informasi selengkapnya tentang lingkungan AWS Batch komputasi, lihat [Lingkungan komputasi](#)

Anda dapat menyesuaikan widget yang ditampilkan di halaman Dasbor. Bagian berikut menjelaskan widget tambahan yang dapat Anda instal.

Antrian pekerjaan tunggal

Widget ini menampilkan informasi rinci tentang antrian pekerjaan tunggal.

Untuk menambahkan widget ini, ikuti langkah-langkah ini.

1. Buka [konsol AWS Batch](#).
2. Dari bilah navigasi, pilih Wilayah AWS yang Anda inginkan.
3. Di panel navigasi, pilih Dasbor.
4. Pilih Tambahkan widget.
5. Untuk antrian pekerjaan tunggal, pilih Tambah widget.

6. Untuk antrean Job, pilih antrean pekerjaan yang Anda inginkan.
7. Untuk status Job, pilih status pekerjaan yang ingin ditampilkan.
8. (Opsional) Matikan Tampilkan lingkungan komputasi yang terhubung jika Anda tidak ingin menampilkan properti untuk lingkungan komputasi.
9. Untuk properti Compute environment, pilih properti yang Anda inginkan.
10. Pilih Tambahkan.

CloudWatch Wawasan Kontainer

Widget ini menampilkan metrik agregat untuk lingkungan AWS Batch komputasi dan pekerjaan. Untuk informasi selengkapnya tentang Wawasan Kontainer ini, silakan lihat [CloudWatch Wawasan Kontainer](#).

Untuk menambahkan widget ini, ikuti langkah-langkah ini.

1. Buka [konsol AWS Batch](#).
2. Dari bilah navigasi, pilih Wilayah AWS yang Anda inginkan.
3. Di panel navigasi, pilih Dasbor.
4. Pilih Tambahkan widget.
5. Untuk wawasan Container, pilih Add widget.
6. Untuk lingkungan Compute, pilih lingkungan komputasi yang Anda inginkan.
7. Pilih Tambahkan.

Log Job

Widget ini menampilkan log yang berbeda dari pekerjaan Anda di satu lokasi yang nyaman. Untuk informasi selengkapnya tentang log pekerjaan, lihat [the section called “ Log Job”](#).

Untuk menambahkan widget ini, ikuti langkah-langkah ini.

1. Buka [konsol AWS Batch](#).
2. Dari bilah navigasi, pilih Wilayah AWS yang Anda inginkan.
3. Di panel navigasi, pilih Dasbor.
4. Pilih Tambahkan widget.
5. Untuk log Job, pilih Add widget.

6. Untuk Job id, masukkan ID pekerjaan untuk pekerjaan yang Anda inginkan.
7. Pilih Tambahkan.

Mengatur dengan AWS Batch

Jika Anda sudah mendaftar ke Amazon Web Services (AWS) dan menggunakan Amazon Elastic Compute Cloud (Amazon EC2) atau Amazon Elastic Container Service (Amazon ECS), Anda dapat segera menggunakannya. AWS Batch Proses pengaturan untuk layanan ini serupa. Ini karena AWS Batch menggunakan instans penampung Amazon ECS di lingkungan komputasinya. Untuk menggunakan AWS CLI bersama AWS Batch, Anda harus menggunakan versi AWS CLI yang mendukung fitur AWS Batch terkini. Jika Anda tidak melihat dukungan untuk AWS Batch fitur di AWS CLI, tingkatkan ke versi terbaru. Untuk informasi selengkapnya, lihat <http://aws.amazon.com/cli/>.

Note

Karena AWS Batch menggunakan komponen Amazon EC2, Anda menggunakan konsol Amazon EC2 untuk sebagian besar langkah berikut.

Selesaikan tugas berikut untuk menyiapkan AWS Batch. Jika Anda sudah menyelesaikan salah satu dari langkah-langkah ini, Anda dapat langsung melompat untuk menginstal AWS CLI.

Topik

- [Mendaftar Akun AWS](#)
- [Membuat pengguna administratif](#)
- [Buat peran IAM untuk lingkungan komputasi dan instance kontainer Anda](#)
- [Membuat pasangan kunci](#)
- [Buat VPC](#)
- [Membuat grup keamanan](#)
- [Instal AWS CLI](#)

Mendaftar Akun AWS

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar Akun AWS

1. Buka <https://portal.aws.amazon.com/billing/signup>.

2. Ikuti petunjuk secara online.

Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

Saat Anda mendaftar Akun AWS, Pengguna root akun AWS akan dibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, [tetapkan akses administratif ke pengguna administratif](#), dan hanya gunakan pengguna root untuk melakukan [tugas yang memerlukan akses pengguna root](#).

AWS akan mengirimkan email konfirmasi kepada Anda setelah proses pendaftaran selesai. Anda dapat melihat aktivitas akun saat ini dan mengelola akun dengan mengunjungi <https://aws.amazon.com/> dan memilih Akun Saya.

Membuat pengguna administratif

Setelah mendaftar Akun AWS, amankan Pengguna root akun AWS, aktifkan AWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Mengamankan Pengguna root akun AWS Anda

1. Masuk ke [AWS Management Console](#) sebagai pemilik akun dengan memilih Pengguna root dan memasukkan alamat email Akun AWS Anda. Di halaman berikutnya, masukkan kata sandi Anda.

Untuk bantuan masuk menggunakan pengguna root, lihat [Masuk sebagai pengguna root](#) dalam Panduan Pengguna AWS Sign-In.

2. Aktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

Untuk petunjuknya, silakan lihat [Mengaktifkan perangkat MFA virtual untuk pengguna root Akun AWS Anda \(konsol\)](#) dalam Panduan Pengguna IAM.

Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

Untuk petunjuk, lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan Pengguna AWS IAM Identity Center.

2. Di Pusat Identitas IAM, berikan izin akses administratif ke pengguna administratif.

Untuk tutorial tentang penggunaan Direktori Pusat Identitas IAM sebagai sumber identitas Anda, lihat [Mengonfigurasi akses pengguna dengan Direktori Pusat Identitas IAM default](#) di Panduan Pengguna AWS IAM Identity Center.

Masuk sebagai pengguna administratif

- Untuk masuk dengan pengguna Pusat Identitas IAM, gunakan URL masuk yang dikirim ke alamat email Anda saat Anda membuat pengguna Pusat Identitas IAM.

Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, lihat [Masuk ke portal akses AWS](#) dalam Panduan Pengguna AWS Sign-In.

Buat peran IAM untuk lingkungan komputasi dan instance kontainer Anda

Lingkungan AWS Batch komputasi dan instans penampung Anda memerlukan Akun AWS kredensial untuk melakukan panggilan ke AWS API lain atas nama Anda. Buat peran IAM yang menyediakan kredensial ini ke lingkungan komputasi dan instance container Anda, lalu kaitkan peran tersebut dengan lingkungan komputasi Anda.

Note

Lingkungan AWS Batch komputasi dan peran instance container dibuat secara otomatis untuk Anda dalam pengalaman konsol yang dijalankan pertama kali. Jadi, jika Anda berniat menggunakan AWS Batch konsol, Anda dapat melanjutkan ke bagian berikutnya. Jika Anda berencana menggunakan AWS CLI, lengkapi prosedur di [Menggunakan peran terkait layanan untuk AWS Batch](#) dan [Peran instans Amazon ECS](#) sebelum membuat lingkungan komputasi pertama Anda.

Membuat pasangan kunci

AWS menggunakan kriptografi kunci publik untuk mengamankan informasi masuk instans Anda. Instance Linux, seperti instance container lingkungan AWS Batch komputasi, tidak memiliki kata sandi untuk digunakan untuk akses SSH. Anda menggunakan pasangan kunci untuk masuk

dengan aman ke instans Anda. Anda menentukan nama pasangan kunci saat membuat lingkungan komputasi, kemudian berikan kunci privat saat masuk menggunakan SSH.

Jika Anda belum membuat key pair, Anda dapat membuatnya menggunakan konsol Amazon EC2. Perhatikan bahwa, jika Anda berencana untuk meluncurkan instance dalam beberapa Wilayah AWS, buat key pair di setiap Region. Untuk informasi lebih lanjut tentang Wilayah, lihat [Wilayah dan Availability Zones](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk membuat pasangan kunci

1. Buka konsol Amazon EC2 di <https://console.aws.amazon.com/ec2/>.
2. Dari bilah navigasi, pilih Wilayah AWS untuk key pair. Anda dapat memilih Wilayah yang tersedia untuk Anda, di mana pun lokasi Anda; namun, pasangan kunci bersifat khusus untuk Wilayah tertentu. Misalnya, jika Anda berencana untuk meluncurkan instance di Wilayah AS Barat (Oregon), buat key pair untuk instance di Wilayah yang sama.
3. Di panel navigasi, pilih Key Pairs (Pasangan Kunci), Create Key Pair (Buat Pasangan Kunci).
4. Di kotak dialog Create Key Pair (Buat Pasangan Kunci), untuk Key pair name (Nama pasangan kunci), masukkan nama untuk pasangan kunci baru, dan pilih Create (Buat). Pilih nama yang dapat Anda ingat, seperti nama pengguna Anda, diikuti oleh `-key-pair`, ditambah nama Wilayah. Misalnya, `me-key-pair-uswest2`.
5. File kunci privat tersebut akan secara otomatis diunduh oleh peramban Anda. Nama file dasar adalah nama yang Anda tentukan sebagai nama key pair Anda, dan ekstensi nama file adalah `.pem`. Simpan file kunci privat di suatu tempat yang aman.

Important

Ini adalah satu-satunya kesempatan Anda untuk menyimpan file kunci privat tersebut. Anda perlu memberikan nama key pair Anda saat meluncurkan instance dan kunci pribadi yang sesuai setiap kali Anda terhubung ke instance.

6. Jika Anda menggunakan klien SSH di komputer Mac atau Linux untuk terhubung ke instance Linux Anda, gunakan perintah berikut untuk mengatur izin file kunci pribadi Anda. Dengan begitu, hanya Anda yang bisa membacanya.

```
$ chmod 400 your_user_name-key-pair-region_name.pem
```

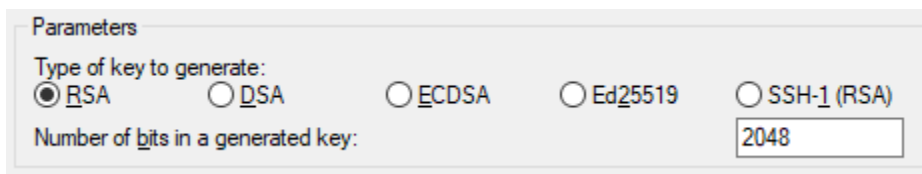
Untuk informasi selengkapnya, silakan lihat [Pasangan Kunci Amazon EC2](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk menyambungkan ke instans Anda menggunakan pasangan kunci baru

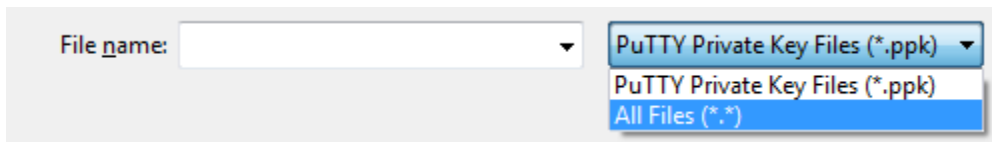
Untuk menyambungkan ke instans Linux Anda dari komputer yang menjalankan Mac atau Linux, tentukan file `.pem` untuk klien SSH Anda dengan opsi `-i` dan jalur ke kunci privat Anda. Untuk terhubung ke instance Linux Anda dari komputer yang menjalankan Windows, gunakan salah satu MindTerm atau Putty. Jika Anda berencana untuk menggunakan PuTTY, instal dan gunakan prosedur berikut untuk mengonversi `.pem` file menjadi `.ppk` file.

(Opsional) Untuk bersiap menyambungkan instans Linux dari Windows menggunakan PuTTY

1. Unduh dan install PuTTY dari <http://www.chiark.greenend.org.uk/~sgtatham/putty/>. Pastikan untuk menginstal seluruh rangkaian.
2. Mulai PuTTYgen (misalnya, dari menu Start (Mulai), pilih All Programs, PuTTY, and PuTTYgen (Semua Program, PuTTY, dan PuTTYgen)).
3. Di bawah tipe kunci yang akan dibuat, pilih RSA. Jika Anda menggunakan versi sebelumnya dari PuTTYgen, pilih SSH-2 RSA.



4. Pilih Muat. Secara default, PuTTYgen hanya menampilkan file dengan ekstensi `.ppk`. Untuk menemukan file `.pem`, pilih opsi untuk menampilkan file dari semua tipe.



5. Pilih file kunci privat yang Anda buat di prosedur sebelumnya dan pilih Open (Buka). Pilih OK untuk menutup kotak dialog konfirmasi.
6. Pilih Save private key (Simpan kunci privat). PuTTYgen menampilkan peringatan tentang menyimpan kunci tanpa frasa sandi. Pilih Ya.
7. Tentukan nama yang sama untuk kunci yang Anda gunakan untuk pasangan kunci. PuTTY secara otomatis menambahkan ekstensi file `.ppk`.

Buat VPC

Dengan Amazon Virtual Private Cloud (Amazon VPC), Anda dapat meluncurkan AWS sumber daya ke jaringan virtual yang telah Anda tentukan. Kami sangat menyarankan agar Anda meluncurkan instance container Anda di VPC.

Jika Anda memiliki VPC default, Anda juga dapat melewati bagian ini dan pindah ke tugas berikutnya. [Membuat grup keamanan](#) Untuk menentukan apakah Anda memiliki VPC default, lihat [Platform yang Didukung di Konsol Amazon EC2 di Panduan Pengguna](#) Amazon EC2 untuk Instans Linux

Untuk informasi tentang cara membuat VPC Amazon, lihat [Membuat VPC hanya di Panduan Pengguna Amazon VPC](#). Lihat tabel berikut untuk menentukan opsi apa yang harus dipilih.

Opsi	Nilai	
Sumber daya untuk dibuat	Hanya VPC	
Nama	Secara opsional berikan nama untuk VPC Anda.	
Blok CIDR IPv4	Input manual IPv4 CIDR Ukuran blok CIDR harus memiliki ukuran antara /16 dan /28.	
Blok CIDR IPv6	Tidak ada blok IPv6 CIDR	
Penghunian	Default	

Untuk informasi lebih lanjut tentang Amazon VPC, lihat [Apa itu Amazon VPC?](#) dalam Panduan Pengguna Amazon VPC.

Membuat grup keamanan

Grup keamanan bertindak sebagai firewall untuk instans kontainer lingkungan komputasi terkait, yang mengontrol lalu lintas masuk dan keluar di tingkat instans kontainer. grup keamanan hanya dapat digunakan dalam VPC yang menjadi tujuan pembuatan grup keamanan tersebut.

Anda harus menambahkan aturan ke grup keamanan yang memungkinkan Anda terhubung ke instans kontainer dari alamat IP Anda menggunakan SSH. Anda juga dapat menambahkan aturan yang mengizinkan akses HTTP dan HTTPS masuk dan keluar dari mana saja. Tambahkan aturan apa pun untuk membuka port yang diperlukan oleh tugas Anda.

Ingatlah bahwa jika Anda berencana meluncurkan instans kontainer di beberapa Wilayah, Anda perlu membuat grup keamanan di setiap Wilayah. Untuk informasi lebih lanjut, lihat [Wilayah dan Availability Zones](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Note

Anda memerlukan alamat IP publik komputer lokal Anda, yang bisa Anda dapatkan menggunakan layanan. Sebagai contoh, kami menyediakan layanan berikut: <http://checkip.amazonaws.com/> atau <https://checkip.amazonaws.com/>. Untuk menemukan layanan lain yang menyediakan alamat IP Anda, gunakan frasa pencarian "apa alamat IP saya." Jika Anda terhubung melalui penyedia layanan Internet (ISP) atau dari belakang firewall tanpa alamat IP statis, cari tahu kisaran alamat IP yang digunakan oleh komputer klien.


Untuk membuat grup keamanan menggunakan konsol

1. Buka konsol Amazon VPC di <https://console.aws.amazon.com/vpc/>.
2. Di panel navigasi, pilih Grup Keamanan.
3. Pilih Buat grup keamanan.
4. Masukkan nama dan deskripsi untuk grup keamanan tersebut. Anda tidak dapat mengubah nama dan deskripsi grup keamanan setelah dibuat.
5. Dari VPC, pilih VPC.
6. (Opsional) Secara default, grup keamanan baru dimulai dengan hanya aturan keluar yang memungkinkan semua lalu lintas meninggalkan sumber daya. Anda harus menambahkan aturan-aturan lain untuk mengizinkan lalu lintas ke dalam atau membatasi lalu lintas ke luar.

AWS Batchinstance kontainer tidak memerlukan port masuk untuk dibuka. Namun, Anda mungkin ingin menambahkan aturan SSH. Dengan begitu, Anda dapat masuk ke instance container dan memeriksa container dalam pekerjaan dengan perintah Docker. Jika Anda ingin instance container Anda meng-host pekerjaan yang menjalankan server web, Anda juga dapat menambahkan aturan untuk HTTP. Selesaikan langkah-langkah berikut untuk menambahkan aturan grup keamanan opsional.

Di tab Inbound (Masuk), buat aturan berikut dan pilih Create (Buat):

- Pilih Add Rule (Tambahkan Aturan). Untuk Type (Tipe), pilih HTTP. Untuk Source (Sumber), pilih Anywhere (Mana Saja) (0.0.0.0/0).
- Pilih Add Rule (Tambahkan Aturan). Untuk Tipe, pilih SSH. Untuk Sumber, pilih IP Kustom, dan tentukan alamat IP publik komputer atau jaringan Anda dalam notasi Classless Inter-Domain Routing (CIDR). Jika perusahaan Anda mengalokasikan alamat-alamat dari rentang alamat, maka masukkan rentang alamat tersebut secara keseluruhan, seperti 203.0.113.0/24. Untuk menentukan alamat IP individual dalam notasi CIDR, pilih IP Saya. Ini menambahkan awalan routing /32 ke alamat IP publik.

 Note

Untuk alasan keamanan, kami tidak menyarankan Anda mengizinkan akses SSH dari semua alamat IP (0.0.0.0/0) ke instans Anda tetapi hanya untuk tujuan pengujian dan hanya untuk waktu yang singkat.

7. Anda dapat menambahkan tanda sekarang, atau Anda dapat menambahkannya nanti. Untuk menambahkan tanda, pilih Tambahkan tanda baru dan masukkan kunci dan nilai tanda.
8. Pilih Create security group (Buat grup keamanan).

Untuk membuat grup keamanan menggunakan baris perintah, lihat [create-security-group](#)(AWS CLI)

Untuk informasi selengkapnya tentang grup keamanan, lihat [Bekerja dengan grup keamanan](#).

Instal AWS CLI

Untuk menggunakan AWS CLI dengan AWS Batch, instal versi AWS CLI terbaru. Untuk informasi tentang menginstal AWS CLI atau meningkatkannya ke versi terbaru, lihat [Menginstal AWS Command Line Interface](#) dalam Panduan Pengguna AWS Command Line Interface.

Memulai dengan AWS Batch

Anda dapat menggunakan wizard yang AWS Batch dijalankan pertama untuk memulai dengan AWS Batch cepat. Setelah menyelesaikan Prasyarat, Anda dapat menggunakan wizard yang dijalankan pertama untuk membuat lingkungan komputasi, definisi pekerjaan, dan antrian pekerjaan.

Anda juga dapat mengirimkan contoh pekerjaan “Hello World” menggunakan wizard yang AWS Batch dijalankan pertama untuk menguji konfigurasi Anda. Jika Anda sudah memiliki gambar Docker yang ingin Anda luncurkan AWS Batch, Anda dapat menggunakan gambar itu untuk membuat definisi pekerjaan.

Prasyarat

Pastikan untuk melakukan hal berikut sebelum Anda memulai wizard yang AWS Batch dijalankan pertama:

- Selesaikan langkah-langkah yang dijelaskan dalam [Mengatur dengan AWS Batch](#).
- Verifikasi bahwa Anda Akun AWS memiliki [izin yang diperlukan](#).

Memulai - Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) menyediakan kapasitas komputasi yang dapat diskalakan di AWS Cloud. Menggunakan Amazon EC2 menghilangkan kebutuhan Anda untuk berinvestasi pada perangkat keras di awal, sehingga Anda dapat mengembangkan dan menerapkan aplikasi lebih cepat.


Anda dapat menggunakan Amazon EC2 untuk meluncurkan server virtual sebanyak atau sesedikit yang Anda butuhkan, mengonfigurasi keamanan dan jaringan, dan mengelola penyimpanan. Amazon EC2 memungkinkan Anda untuk meningkatkan atau menurunkan untuk menangani perubahan dalam persyaratan atau lonjakan popularitas, mengurangi kebutuhan Anda untuk memperkirakan lalu lintas.

Membuat lingkungan komputasi

Untuk membuat lingkungan komputasi untuk orkestrasi Amazon EC2, lakukan hal berikut:


1. Buka [wizard yang dijalankan pertama AWS Batch konsol](#).

2. Untuk Pilih jenis orkestrasi, pilih Amazon Elastic Compute Cloud (Amazon EC2).
3. Pilih Berikutnya.
4. Di bagian konfigurasi lingkungan komputasi untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
5. Untuk peran Instance, pilih profil instans yang sudah ada yang memiliki izin IAM yang diperlukan terlampir. Profil instans ini memungkinkan instans container Amazon ECS di lingkungan komputasi Anda untuk melakukan panggilan ke operasi API yang diperlukan AWS . Untuk informasi selengkapnya, lihat [Peran instans Amazon ECS](#).
6. (Opsional) Tag adalah label yang ditetapkan ke sumber daya. Untuk menambahkan tag atau tag Amazon EC2, perluas Tag, lalu pilih Tambah tag. Masukkan pasangan kunci-nilai, lalu pilih Tambah tag lagi.

 Important


Jika Anda memilih Tambah tag, Anda harus memasukkan pasangan kunci-nilai dan memilih Tambah tag lagi atau pilih Hapus tag.

7. (Opsional) Di bagian Konfigurasi instans untuk Menggunakan instans Spot Amazon EC2, aktifkan Aktifkan menggunakan instans Spot.
8. (Hanya spot) Untuk harga maksimum% sesuai permintaan, masukkan persentase maksimum harga On-Demand yang ingin Anda bayar untuk sumber daya Spot.
9. (Opsional) (Hanya spot) Untuk peran armada Spot, pilih peran IAM Armada Spot Amazon EC2 yang ada untuk diterapkan ke lingkungan komputasi Spot Anda. Jika Anda belum memiliki peran IAM Armada Spot Amazon EC2, Anda harus membuatnya terlebih dahulu. Untuk informasi selengkapnya, lihat [Peran armada spot Amazon EC2](#).


 Important

Untuk menandai Instans Spot saat pembuatan, peran IAM Armada Spot Amazon EC2 Anda harus menggunakan kebijakan terkelola AmazonEC2 yang lebih baru SpotFleetTaggingRole. Kebijakan SpotFleetRole terkelola AmazonEC2 tidak memiliki izin yang diperlukan untuk menandai Instans Spot. Lihat informasi yang lebih lengkap di [Instans Spot tidak ditandai pada pembuatan](#) dan [the section called "Penandaan sumber daya Anda"](#).

10. Untuk vCPU Minimum, pilih jumlah minimum vCPU EC2 yang dipertahankan oleh lingkungan komputasi Anda, terlepas dari permintaan antrian pekerjaan.
11. Untuk vCPU yang Diinginkan, pilih jumlah vCPU EC2 yang diluncurkan oleh lingkungan komputasi Anda. Ketika permintaan antrian pekerjaan meningkat, AWS Batch meningkatkan jumlah vCPU yang diinginkan dan menambahkan instans EC2. Jumlah vCPU dapat meningkat hingga jumlah maksimum vCPU. Ketika permintaan AWS Batch menurun, mengurangi jumlah vCPU yang diinginkan dan menghapus instance. Jumlah penurunan sampai ke jumlah minimum vCPU.
12. Untuk Maximum vCPUs (vCPUs maksimum), pilih jumlah maksimum penskalaan keluar vCPUs EC2 yang dapat dilakukan lingkungan komputasi Anda, terlepas dari permintaan antrian tugas.
13. Untuk Allowed instance types (Tipe instans yang diizinkan), pilih tipe instans Amazon EC2 yang dapat diluncurkan. Anda dapat menentukan keluarga instans untuk meluncurkan tipe instans apa pun dalam keluarga tersebut (misalnya, c5, c5n, atau p3). Selain itu, Anda dapat menentukan ukuran tertentu dalam suatu keluarga (seperti c5.8xlarge). Jenis instans logam tidak ada dalam keluarga instance. Misalnya, c5 tidak mencakup c5.meta1. Anda juga dapat memilih optimal untuk memilih jenis instans (dari C4, M4, dan keluarga R4 instance) yang sesuai dengan permintaan antrian pekerjaan Anda.

 Note

Ketika Anda membuat lingkungan komputasi, tipe instans yang Anda pilih untuk lingkungan komputasi harus memiliki arsitektur yang sama. Misalnya, Anda tidak dapat mencampur instans x86 dan ARM di lingkungan komputasi yang sama.

 Note

AWS Batch skala GPU berdasarkan jumlah yang diperlukan dalam antrian pekerjaan Anda. Untuk menggunakan penjadwalan GPU, lingkungan komputasi harus menyertakan jenis instance dari p2,,p3,p4,, p5 g3 g3sg4, atau keluarga. g5

Note

Saat ini, optimal menggunakan tipe instance dari keluarga C4M4,, dan R4 instance. Dalam Wilayah AWS hal itu tidak memiliki tipe instance dari keluarga instance tersebut, tipe instance dari C5M5,, dan keluarga R5 instance digunakan.

14. Perluas Additional configuration (Konfigurasi tambahan).
15. (Opsional) Untuk grup Penempatan, masukkan nama grup penempatan untuk mengelompokkan sumber daya di lingkungan komputasi.
16. (Opsional) Untuk key pair EC2, pilih public dan private key pair sebagai security credentials saat Anda terhubung ke instans. Untuk informasi selengkapnya tentang pasangan kunci Amazon EC2, lihat pasangan kunci [Amazon EC2 dan instans Linux](#).
17. Untuk Allocation strategy (Strategi alokasi), pilih strategi alokasi yang akan digunakan ketika memilih tipe instans dari daftar tipe instans yang diizinkan. BEST_FIT_PROGRESSIVE biasanya merupakan pilihan yang lebih baik untuk lingkungan komputasi Sesuai Permintaan EC2 dan SPOT_CAPACITY_OPTIMIZED untuk lingkungan komputasi Spot EC2. Untuk informasi selengkapnya, lihat [the section called "Strategi alokasi"](#).
18. (Opsional) Untuk konfigurasi EC2, pilih Tambahkan konfigurasi EC2. Pilih Nilai ganti tipe gambar dan ID Gambar untuk memberikan informasi AWS Batch untuk memilih Amazon Machine Images (AMI) untuk instance di lingkungan komputasi. Jika penggantian ID Gambar tidak ditentukan untuk setiap jenis Gambar, AWS Batch pilih AMI [Amazon ECS terbaru yang dioptimalkan](#). Jika tidak ada jenis Gambar yang ditentukan, defaultnya adalah Amazon Linux 2 untuk instance non-GPU, non AWS Graviton.

Important

Untuk menggunakan AMI kustom, pilih jenis gambar lalu masukkan ID AMI kustom di kotak penggantian ID Gambar.

[Amazon Linux 2](#)

Default untuk semua keluarga instance AWS berbasis Graviton (misalnya,, C6g M6gR6g, danT4g) dan dapat digunakan untuk semua jenis instans non-GPU.

[Amazon Linux 2 \(GPU\)](#)

Default untuk semua keluarga instance GPU (misalnya P4 dan G4) dan dapat digunakan untuk semua jenis instance berbasis non AWS Graviton.

Amazon Linux

Dapat digunakan untuk keluarga instance non-GPU, non AWS Graviton. Dukungan standar untuk Amazon Linux AMI telah berakhir. Untuk informasi lebih lanjut, lihat [Amazon Linux AMI](#).

Note

AMI yang Anda pilih untuk lingkungan komputasi harus cocok dengan arsitektur tipe instance yang ingin Anda gunakan untuk lingkungan komputasi tersebut. Misalnya, jika lingkungan komputasi Anda menggunakan tipe A1 instans, AMI sumber daya komputasi yang Anda pilih harus mendukung Arm instans. Amazon ECS menjual keduanya x86 dan Arm versi Amazon ECS yang dioptimalkan Amazon Linux 2 AMI. Untuk informasi lebih lanjut, lihat [AMI Amazon Linux 2 yang dioptimalkan untuk Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

19. (Opsional) Untuk Launch template (Templat peluncuran), pilih templat peluncuran Amazon EC2 yang sudah ada untuk mengonfigurasi sumber daya komputasi Anda. Versi default templat akan diisi secara otomatis. Untuk informasi selengkapnya, lihat [Luncurkan dukungan templat](#).

Note

Dalam template peluncuran, Anda dapat menentukan AMI kustom yang Anda buat.

20. (Opsional) Untuk Launch template version (Versi templat peluncuran), masukkan `$Default`, `$Latest`, atau nomor versi tertentu yang akan digunakan.

Important

Setelah lingkungan komputasi dibuat, versi templat peluncuran tidak berubah meskipun versi `$Default` atau `$Latest` untuk templat peluncuran diperbarui. Untuk menggunakan versi template peluncuran baru, pertama buat lingkungan komputasi baru, tambahkan lingkungan komputasi baru ke antrian pekerjaan yang ada. Kemudian, hapus

lingkungan komputasi lama dari antrian pekerjaan, dan hapus lingkungan komputasi lama.

21. Di bagian Konfigurasi Jaringan:

- a. Untuk ID Virtual Private Cloud (VPC), pilih VPC Amazon.
- b. Untuk Subnet, subnet untuk Anda Akun AWS terdaftar. Jika Anda ingin membuat kumpulan subnet khusus, pilih Hapus subnet, lalu pilih subnet yang Anda inginkan.

Important

Sumber daya komputasi harus berkomunikasi dengan titik akhir VPC Amazon ECS melalui titik akhir VPC atau beberapa alamat IP publik. Untuk informasi selengkapnya, lihat [Titik akhir AWS PrivateLinkVPC antarmuka Amazon ECS](#) (). Jika instans Anda tidak memiliki titik akhir VPC yang dikonfigurasi atau alamat IP publik, Anda dapat menggunakan terjemahan alamat jaringan (NAT). Untuk informasi lebih lanjut tentang NAT, lihat gateway [NAT](#) dan [Membuat cloud pribadi virtual](#)

- c. Untuk grup Keamanan, pilih grup keamanan Amazon EC2 yang ingin Anda kaitkan dengan instans. Jika Anda ingin membuat kumpulan grup keamanan khusus, pilih Hapus grup keamanan. Kemudian, pilih grup keamanan yang Anda inginkan.

22. Pilih Berikutnya.

Membuat antrean tugas

Antrian pekerjaan menyimpan pekerjaan yang Anda kirimkan hingga AWS Batch Scheduler menjalankan pekerjaan pada sumber daya di lingkungan komputasi Anda. Lihat informasi yang lebih lengkap di [Antrean tugas](#)

Untuk membuat antrean pekerjaan untuk orkestrasi Amazon EC2, lakukan hal berikut:

1. Di bagian konfigurasi antrian Job untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
2. Untuk Prioritas, masukkan bilangan bulat antara 0 dan 100 untuk antrian pekerjaan.

⚠ Important

Nilai integer yang lebih tinggi diberi prioritas yang lebih tinggi oleh AWS Batch Scheduler.

3. Pilih Berikutnya.

Buat definisi pekerjaan

AWS Batch definisi pekerjaan menentukan bagaimana pekerjaan harus dijalankan. Meskipun setiap pekerjaan harus mereferensikan definisi pekerjaan, banyak parameter yang ditentukan dalam definisi pekerjaan dapat diganti saat runtime.

Untuk membuat definisi pekerjaan:

1. Di bagian Konfigurasi umum:

- a. Di bagian Konfigurasi umum untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat berisi huruf besar dan kecil, angka, tanda hubung (-), dan garis bawah (_).
- b. (Opsional) Untuk batas waktu Eksekusi, masukkan jumlah waktu (dalam detik) bahwa pekerjaan yang belum selesai berakhir setelahnya.

⚠ Important

Batas waktu minimum adalah 60 detik.

- c. (Opsional) Tag adalah label yang ditetapkan ke sumber daya. Untuk menambahkan tag, perluas Tag, lalu pilih Tambah tag. Masukkan pasangan kunci-nilai, lalu pilih Tambah tag lagi.

⚠ Important

Jika Anda memilih Tambah tag, Anda harus memasukkan pasangan kunci-nilai dan memilih Tambah tag lagi atau pilih Hapus tag.

- d. (Opsional) Aktifkan Propagate tag untuk menyebarkan tag ke tugas Amazon Elastic Container Service.


2. Di bagian Konfigurasi kontainer:

- a. Untuk Gambar, masukkan nama gambar yang digunakan untuk meluncurkan wadah. Secara default, semua gambar di registri Docker Hub tersedia. Anda juga dapat menentukan repositori lain dalam format `repository-url/image:tag`. Parameter dapat memiliki panjang hingga 255 karakter. Parameter dapat berisi huruf besar dan kecil, angka, tanda hubung (-), garis bawah (_), titik dua (:), titik (.), garis miring (/), dan tanda angka (#). Parameter ini sesuai dengan Image di bagian [Buat kontainer](#) di [Docker Remote API](#) dan IMAGE parameter [docker run](#).

Note


Dockerarsitektur gambar harus sesuai dengan arsitektur prosesor dari sumber daya komputasi yang dijadwalkan. Misalnya, Docker gambar Arm berbasis hanya dapat berjalan pada sumber daya komputasi Arm berbasis.

- Gambar di Amazon ECR Repositori Publik menggunakan konvensi lengkap `registry/repository[:tag]` atau `registry/repository[@digest]` penamaan (misalnya, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - Gambar di repositori Amazon ECR menggunakan konvensi `registry/repository:tag` penamaan lengkap (misalnya, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Gambar di repositori resmi di Docker Hub menggunakan satu nama (misalnya, `ubuntu` atau `mongo`).
 - Gambar di repositori lain di Docker Hub memenuhi syarat dengan nama organisasi (misalnya, `amazon/amazon-ecs-agent`).
 - Gambar di repositori online lainnya memenuhi syarat lebih lanjut dengan nama domain (misalnya, `quay.io/assemblyline/ubuntu`).
- b. Untuk Command (Perintah), tentukan perintah untuk diteruskan ke kontainer. Parameter ini sesuai dengan `Cmd` di bagian [Buat kontainer](#) di [Docker Remote API](#) dan parameter `COMMAND` untuk [docker run](#). Untuk informasi lebih lanjut tentang parameter CMD Docker, lihat <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note


Anda dapat menggunakan nilai dan placeholder default substitusi parameter dalam perintah Anda. Untuk informasi selengkapnya, lihat [Parameter-parameter](#).

- c. (Opsional) Untuk peran Eksekusi, tentukan peran IAM yang memberikan izin kepada agen penampung Amazon ECS untuk melakukan panggilan AWS API atas nama Anda. Fitur ini menggunakan peran Amazon ECS IAM untuk tugas. Untuk informasi selengkapnya, lihat [peran IAM eksekusi tugas Amazon ECS](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.
- d. (Opsional) Untuk konfigurasi Job Role, pilih peran IAM yang memiliki izin ke API. AWS Fitur ini menggunakan peran Amazon ECS IAM untuk tugas. Untuk informasi lebih lanjut, lihat [Peran IAM untuk Tugas](#) dalam Panduan Developer Amazon Elastic Container Service.

 Note

Hanya peran yang memiliki hubungan kepercayaan Amazon Elastic Container Service Task Role (Peran Tugas Amazon Elastic Container Service) yang ditampilkan di sini. Untuk informasi selengkapnya tentang membuat peran IAM untuk AWS Batch pekerjaan Anda, lihat [Membuat Peran dan Kebijakan IAM untuk Tugas Anda](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.

- e. (Opsional) Anda dapat menambahkan parameter ke definisi pekerjaan sebagai pemetaan nilai kunci untuk mengganti default definisi pekerjaan. Untuk menambahkan parameter:
 - Untuk Parameter, pilih Tambah parameter. Masukkan pasangan kunci-nilai dan kemudian pilih Tambah parameter lagi.

 Important

Jika Anda memilih Tambah parameter, Anda harus mengkonfigurasi setidaknya satu parameter atau memilih Hapus parameter.

- f. Di bagian konfigurasi Lingkungan untuk vCPU, tentukan jumlah vCPU yang akan dicadangkan untuk wadah. Parameter ini sesuai dengan CpuShares di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--cpu-shares` untuk [docker run](#). Setiap vCPU setara dengan 1.024 bagian CPU.

- g. Untuk Memori, tentukan batas keras (dalam MiB) memori untuk disajikan ke wadah pekerjaan. Jika kontainer Anda mencoba untuk melebihi memori yang ditentukan di sini, kontainer akan dihentikan. Parameter ini sesuai dengan Memory di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--memory` untuk [docker run](#).
- h. Untuk Jumlah GPU, pilih jumlah GPU yang akan dicadangkan untuk kontainer.
- i. (Opsional) Untuk konfigurasi variabel Lingkungan, pilih Tambahkan variabel lingkungan untuk menambahkan variabel lingkungan untuk diteruskan ke wadah. Parameter ini memetakan ke Env di bagian [Membuat kontainer](#) dari [API Jarak Jauh Docker](#) dan pilihan `--env` untuk [docker run](#).
- j. (Opsional) Untuk Rahasia, pilih Tambahkan rahasia untuk menambahkan rahasia sebagai pasangan nama-nilai. Rahasia-rahasia ini terungkap dalam wadah. Untuk informasi lebih lanjut, lihat [secretOptions](#) di [Parameter definisi Job untuk ContainerProperties](#).
- k. (Opsional) Di bagian konfigurasi Linux:
 - i. Untuk User (Pengguna), masukkan nama pengguna untuk digunakan di dalam kontainer. Parameter ini sesuai dengan User di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--user` untuk [docker run](#).
 - ii. Untuk memberikan izin yang lebih tinggi pada container pekerjaan pada instance host (mirip dengan root pengguna), seret slider Privileged ke kanan. Parameter ini memetakan ke Privileged di bagian [Membuat kontainer](#) dari [API Jarak Jauh Docker](#) dan pilihan `--privileged` untuk [docker run](#).
 - iii. Aktifkan Aktifkan proses init untuk menjalankan `init` proses di dalam wadah. Proses ini meneruskan sinyal dan menuai proses.
- l. (Opsional) Di bagian konfigurasi Sistem File:
 - i. Aktifkan Aktifkan sistem file baca saja untuk menghapus akses tulis ke volume.
 - ii. Untuk ukuran memori bersama, masukkan ukuran (dalam MiB) volume. `/dev/shm`
 - iii. Untuk ukuran swap Max, masukkan jumlah total memori swap (dalam MiB) yang dapat digunakan wadah.
 - iv. Untuk Swappiness masukkan nilai antara 0 dan 100 untuk menunjukkan perilaku swappiness wadah. Jika Anda tidak menentukan nilai dan pertukaran diaktifkan, nilai defaultnya menjadi 60. Untuk informasi lebih lanjut, lihat [swappiness](#) di [Parameter definisi Job untuk ContainerProperties](#).
 - v. (Opsional) Perluas Additional configuration (Konfigurasi tambahan).

- vi. Untuk Tmpfs, pilih Tambahkan tmpfs untuk menambahkan mount. tmpfs
 - vii. Untuk Perangkat, pilih Tambah perangkat untuk menambahkan perangkat:
 - A. Untuk Container path (Jalur kontainer), tentukan jalur dalam instans kontainer untuk mengungkap perangkat yang dipetakan ke instans host. Jika Anda mengosongkan ini, jalur host digunakan dalam wadah.
 - B. Untuk Host path (Jalur host), tentukan jalur perangkat dalam instans host.
 - C. Untuk Izin, pilih satu atau beberapa izin untuk diterapkan ke perangkat. Izin yang tersedia adalah BACA, TULIS, dan MKNOD.
 - viii. (Opsional) Untuk konfigurasi Ulimits, pilih Tambahkan ulimit untuk menambahkan `ulimits` nilai untuk wadah. Masukkan nilai Name, Soft limit, dan Hard limit, lalu pilih Add ulimit.
3. Pilih Berikutnya.

Buat pekerjaan

Untuk membuat pekerjaan, lakukan hal berikut:

1. Di bagian konfigurasi Job untuk Nama, tentukan nama unik untuk pekerjaan tersebut. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
2. Pilih Berikutnya.

Tinjau dan buat

Pada halaman Tinjau dan buat, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat sumber daya.

Memulai - Fargate

AWS Fargate meluncurkan dan menskalakan komputasi agar sesuai dengan persyaratan sumber daya yang Anda tentukan untuk penampung. Dengan Fargate, Anda tidak perlu menyediakan secara berlebihan atau membayar server tambahan. Untuk informasi lebih lanjut, lihat [Fargate](#).

Membuat lingkungan komputasi

Untuk membuat lingkungan komputasi untuk orkestrasi Fargate, lakukan hal berikut:

1. Buka [wizard yang dijalankan pertama AWS Batch konsol](#).
2. Untuk Pilih jenis orkestrasi, pilih Fargate.
3. Pilih Berikutnya.
4. Di bagian konfigurasi lingkungan komputasi untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
5. (Opsional) Tag adalah label yang ditetapkan ke sumber daya. Untuk menambahkan tag, perluas Tag, lalu pilih Tambah tag. Masukkan pasangan kunci-nilai, lalu pilih Tambah tag lagi.

Important

Jika Anda memilih Tambah tag, Anda harus memasukkan pasangan kunci-nilai dan memilih Tambah tag lagi atau pilih Hapus tag.

6. (Opsional) Di bagian konfigurasi Instans untuk Gunakan kapasitas Spot Fargate, aktifkan Aktifkan menggunakan instance Spot.
7. Untuk vCPU Maksimum, masukkan jumlah maksimum vCPU yang dapat digunakan instance.
8. Di bagian Konfigurasi Jaringan:
 - a. Untuk ID Virtual Private Cloud (VPC), pilih VPC Amazon.
 - b. Untuk Subnet, subnet untuk Anda Akun AWS terdaftar. Jika Anda ingin membuat kumpulan subnet khusus, pilih Hapus subnet, lalu pilih subnet yang Anda inginkan.

Important

Sumber daya komputasi harus berkomunikasi dengan titik akhir VPC Amazon ECS melalui titik akhir VPC atau beberapa alamat IP publik. Untuk informasi selengkapnya, lihat [Titik akhir AWS PrivateLinkVPC antarmuka Amazon ECS](#) (). Jika instans Anda tidak memiliki titik akhir VPC yang dikonfigurasi atau alamat IP publik, Anda dapat menggunakan terjemahan alamat jaringan (NAT). Untuk informasi lebih lanjut tentang NAT, lihat gateway [NAT](#) dan [Membuat cloud pribadi virtual](#)


- c. Untuk grup Keamanan, pilih grup keamanan Amazon EC2 yang ingin Anda kaitkan dengan instans. Jika Anda ingin membuat kumpulan grup keamanan khusus, pilih Hapus grup keamanan. Kemudian, pilih grup keamanan yang Anda inginkan.
9. Pilih Berikutnya.

Membuat antrean tugas

Antrian pekerjaan menyimpan pekerjaan yang Anda kirimkan hingga AWS Batch Scheduler menjalankan pekerjaan pada sumber daya di lingkungan komputasi Anda. Untuk membuat antrian pekerjaan:

Untuk membuat antrian pekerjaan untuk orkestrasi Fargate, lakukan hal berikut:

1. Di bagian konfigurasi antrian Job untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
2. Untuk Prioritas, masukkan bilangan bulat antara 0 dan 100 untuk antrian pekerjaan.

 Important

Nilai integer yang lebih tinggi diberi prioritas yang lebih tinggi oleh AWS Batch Scheduler.

3. Pilih Berikutnya.

Buat definisi pekerjaan

Untuk membuat definisi pekerjaan:

1. Di bagian Konfigurasi umum:
 - a. Untuk Nama, masukkan nama definisi pekerjaan khusus.

Di bagian Konfigurasi umum untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).

- b. (Opsional) Untuk batas waktu Eksekusi, masukkan jumlah waktu (dalam detik) bahwa pekerjaan yang belum selesai berakhir setelahnya.

 Important

Batas waktu minimum adalah 60 detik.

- c. (Opsional) Tag adalah label yang ditetapkan ke sumber daya. Untuk menambahkan tag, perluas Tag, lalu pilih Tambah tag. Masukkan pasangan kunci-nilai, lalu pilih Tambah tag lagi.


 Important

Jika Anda memilih Tambah tag, Anda harus memasukkan pasangan kunci-nilai dan memilih Tambah tag lagi atau pilih Hapus tag.

- d. (Opsional) Aktifkan Propagate tag untuk menyebarkan tag ke tugas Amazon Elastic Container Service.

2. Di bagian konfigurasi platform Fargate:

- a. (Opsional) Untuk versi platform Fargate, masukkan lingkungan runtime tertentu yang Anda inginkan.
- b. Untuk platform Runtime, pilih LINUX atau Windows.
- c. (Hanya Windows) Untuk Keluarga Sistem Operasi, pilih sistem operasi.
- d. Untuk arsitektur CPU, pilih arsitektur CPU yang Anda inginkan.
- e. (Opsional) Aktifkan Tetapkan IP publik untuk menetapkan alamat IP publik.
- f. Untuk penyimpanan Ephemeral, masukkan jumlah penyimpanan sementara yang Anda inginkan.


 Note

Secara default, 20 GiB penyimpanan sementara digunakan. Untuk menggunakan penyimpanan sementara tambahan, masukkan nilai antara 21 GiB dan 100 GiB.

- g. Untuk peran Eksekusi, pilih peran eksekusi tugas yang memungkinkan agen Amazon Elastic Container Service (Amazon ECS) AWS melakukan panggilan atas nama Anda. Misalnya, Anda dapat memilih ecsTaskExecutionPeran.


3. Di bagian Konfigurasi kontainer:

- a. Untuk Gambar, masukkan nama gambar yang digunakan untuk meluncurkan wadah. Secara default, semua gambar di registri Docker Hub tersedia. Anda juga dapat menentukan repositori lain dalam format `repository-url/image:tag`. Parameter dapat memiliki panjang hingga 255 karakter. Parameter dapat berisi huruf besar dan huruf kecil, angka, tanda hubung (-), garis bawah (_), titik dua (:), titik (.), garis miring (/), dan tanda pagar (#). Parameter ini sesuai dengan Image di bagian [Buat kontainer](#) di [Docker Remote API](#) dan IMAGE parameter [docker run](#).

 Note

Docker arsitektur gambar harus sesuai dengan arsitektur prosesor dari sumber daya komputasi yang dijadwalkan. Misalnya, Docker gambar Arm berbasis hanya dapat berjalan pada sumber daya komputasi Arm berbasis.

- Gambar di Amazon ECR Repositori Publik menggunakan konvensi lengkap `registry/repository[:tag]` atau `registry/repository[@digest]` penamaan (misalnya, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - Gambar di repositori Amazon ECR menggunakan konvensi `registry/repository:tag` penamaan lengkap (misalnya, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Gambar di repositori resmi di Docker Hub menggunakan satu nama (misalnya, `ubuntu` atau `mongo`).
 - Gambar di repositori lain di Docker Hub memenuhi syarat dengan nama organisasi (misalnya, `amazon/amazon-ecs-agent`).
 - Gambar di repositori online lainnya memenuhi syarat lebih lanjut dengan nama domain (misalnya, `quay.io/assemblyline/ubuntu`).
- b. Untuk Command (Perintah), tentukan perintah untuk diteruskan ke kontainer. Parameter ini sesuai dengan `Cmd` di bagian [Buat kontainer](#) di [Docker Remote API](#) dan parameter `COMMAND` untuk [docker run](#). Untuk informasi lebih lanjut tentang parameter CMD Docker, lihat <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note


Anda dapat menggunakan nilai dan placeholder default substitusi parameter dalam perintah Anda. Untuk informasi selengkapnya, lihat [Parameter-parameter](#).

 Tip

Pilih Info untuk meninjau contoh kode Bash dan JSON.

c. (Opsional) Anda dapat menambahkan parameter ke definisi pekerjaan sebagai pemetaan nilai kunci untuk mengganti default definisi pekerjaan. Untuk menambahkan parameter:

- Untuk Parameter, pilih Tambah parameter. Masukkan pasangan kunci-nilai dan kemudian pilih Tambah parameter lagi.

 Important

Jika Anda memilih Tambah parameter, Anda harus mengkonfigurasi setidaknya satu parameter atau memilih Hapus parameter.

- d. (Opsional) Di bagian konfigurasi Lingkungan untuk konfigurasi peran Job, pilih peran IAM yang memberikan izin untuk menggunakan AWS API.
- e. Di bagian konfigurasi Lingkungan untuk vCPU, tentukan jumlah vCPU yang akan dicadangkan untuk wadah. Parameter ini sesuai dengan CpuShares di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--cpu-shares` untuk [docker run](#). Setiap vCPU setara dengan 1.024 bagian CPU.
- f. Untuk Memori, tentukan batas keras (dalam MiB) memori untuk disajikan ke wadah pekerjaan. Jika kontainer Anda mencoba untuk melebihi memori yang ditentukan di sini, kontainer akan dihentikan. Parameter ini sesuai dengan Memory di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--memory` untuk [docker run](#).
- g. (Opsional) Untuk variabel Lingkungan, pilih Tambahkan variabel lingkungan untuk menambahkan variabel lingkungan untuk diteruskan ke wadah. Parameter ini memetakan ke Env di bagian [Membuat kontainer](#) dari [API Jarak Jauh Docker](#) dan pilihan `--env` untuk [docker run](#).

4. Pilih Berikutnya.

Buat pekerjaan

Untuk membuat pekerjaan Fargate, lakukan hal berikut:

1. Di bagian konfigurasi Job untuk Nama, tentukan nama unik untuk pekerjaan tersebut. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
2. Pilih Berikutnya.

Tinjau dan buat

Pada halaman Tinjau dan buat, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat sumber daya.

Memulai dengan AWS Batch di Amazon EKS

AWS Batch di Amazon EKS adalah layanan terkelola untuk menjadwalkan dan menskalakan beban kerja batch ke dalam kluster Amazon EKS yang ada. AWS Batch tidak membuat, mengelola, atau menjalankan operasi siklus hidup kluster Amazon EKS Anda atas nama Anda. AWS Batch skala orkestrasi naik dan turun node yang dikelola oleh AWS Batch dan menjalankan pod pada node tersebut.

AWS Batch tidak menyentuh node, grup node penskalaan otomatis, atau siklus hidup pod yang tidak terkait dengan AWS Batch lingkungan komputasi dalam kluster Amazon EKS Anda. AWS Batch Agar dapat beroperasi secara efektif, [peran terkait layanannya memerlukan izin kontrol akses Kubernetes berbasis peran](#) (RBAC) di kluster Amazon EKS Anda yang ada. Untuk informasi selengkapnya, lihat [Menggunakan Otorisasi RBAC dalam dokumentasi](#). Kubernetes

AWS Batch membutuhkan Kubernetes namespace di mana ia dapat mencakup pod sebagai AWS Batch pekerjaan ke dalam. Kami merekomendasikan namespace khusus untuk mengisolasi AWS Batch pod dari beban kerja kluster Anda yang lain.

Setelah AWS Batch diberi akses RBAC dan namespace telah dibuat, Anda dapat mengaitkan kluster Amazon EKS tersebut ke lingkungan AWS Batch komputasi menggunakan operasi API. [CreateComputeEnvironment](#) Antrian pekerjaan dapat dikaitkan dengan lingkungan komputasi Amazon EKS yang baru ini. AWS Batch pekerjaan dikirimkan ke antrian pekerjaan berdasarkan definisi pekerjaan Amazon EKS menggunakan operasi [SubmitJob](#) API. AWS Batch kemudian

meluncurkan node AWS Batch terkelola dan menempatkan pekerjaan dari antrian pekerjaan sebagai Kubernetes pod ke dalam cluster EKS yang terkait dengan lingkungan AWS Batch komputasi.

Bagian berikut mencakup cara menyiapkan AWS Batch di Amazon EKS.

Daftar Isi

- [Prasyarat](#)
- [Langkah 1: Mempersiapkan cluster Amazon EKS Anda AWS Batch](#)
- [Langkah 2: Membuat lingkungan komputasi Amazon EKS](#)
- [Langkah 3: Buat antrian pekerjaan dan lampirkan lingkungan komputasi](#)
- [Langkah 4: Buat definisi pekerjaan](#)
- [Langkah 5: Kirim pekerjaan](#)
- [\(Opsional\) Kirim pekerjaan dengan penggantian](#)
- [Memulai dengan AWS Batch di Amazon EKS Private Clusters](#)
 - [Prasyarat](#)
 - [Langkah 1: Mempersiapkan cluster EKS Anda AWS Batch](#)
 - [Langkah 2: Membuat lingkungan komputasi Amazon EKS](#)
 - [Langkah 3: Buat antrian pekerjaan dan lampirkan lingkungan komputasi](#)
 - [Langkah 4: Buat definisi pekerjaan](#)
 - [Langkah 5: Kirim pekerjaan](#)
 - [\(Opsional\) Kirim pekerjaan dengan penggantian](#)
 - [Pemecahan Masalah](#)

Prasyarat

Sebelum memulai tutorial ini, Anda harus menginstal dan mengkonfigurasi alat dan sumber daya berikut yang Anda butuhkan untuk membuat AWS Batch dan mengelola sumber daya Amazon EKS.

- AWS CLI — Alat baris perintah untuk bekerja dengan layanan AWS , termasuk Amazon EKS. Panduan ini mengharuskan Anda menggunakan versi 2.8.6 atau yang lebih baru atau 1.26.0 atau yang lebih baru. Untuk informasi selengkapnya, lihat [Menginstal, memperbarui, dan mencopot instalasi AWS CLI](#) di Panduan Pengguna AWS Command Line Interface . Setelah menginstal AWS CLI, kami sarankan Anda juga mengkonfigurasinya. Untuk informasi selengkapnya, lihat [Konfigurasi cepat dengan aws configure](#) di Panduan AWS Command Line Interface Pengguna.

- **kubectl**— Alat baris perintah untuk bekerja dengan Kubernetes cluster. Panduan ini mengharuskan Anda menggunakan versi 1.23 atau yang lebih baru. Untuk informasi selengkapnya, silakan lihat [Menginstal atau memperbarui kubectl](#) di Panduan Pengguna Amazon EKS.
- **eksctl**— Alat baris perintah untuk bekerja dengan kluster Amazon EKS yang mengotomatiskan banyak tugas individu. Panduan ini mengharuskan Anda menggunakan versi 0.115.0 atau yang lebih baru. Untuk informasi selengkapnya, silakan lihat [Menginstal atau memperbarui eksctl](#) di Panduan Pengguna Amazon EKS.
- Izin IAM yang diperlukan — Prinsip keamanan IAM yang Anda gunakan harus memiliki izin untuk bekerja dengan peran Amazon EKS IAM dan peran terkait layanan, serta VPC AWS CloudFormation dan sumber daya terkait. Untuk informasi selengkapnya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk Amazon Elastic Kubernetes Service dan Menggunakan peran terkait](#) layanan di Panduan Pengguna IAM. Anda harus menyelesaikan semua langkah dalam panduan ini sebagai pengguna yang sama.
- Membuat kluster Amazon EKS — Untuk informasi selengkapnya, lihat [Memulai Amazon EKS — eksctl](#) di Panduan Pengguna Amazon EKS.

Note

AWS Batch hanya mendukung kluster Amazon EKS dengan titik akhir server API yang memiliki akses publik, dapat diakses oleh internet publik. Secara default, titik akhir server API kluster Amazon EKS memiliki akses publik. Untuk informasi selengkapnya, lihat [kontrol akses titik akhir kluster Amazon EKS](#) di Panduan Pengguna Amazon EKS.

Note

AWS Batch tidak menyediakan orkestrasi node terkelola untuk CoreDNS atau pod penerapan lainnya. Jika Anda membutuhkan CoreDNS, lihat [Menambahkan add-on CoreDNS Amazon EKS di Panduan Pengguna Amazon EKS](#). Atau, gunakan `eksctl create cluster create` untuk membuat cluster, itu termasuk CoreDNS secara default.

- Izin — Pengguna yang memanggil operasi [CreateComputeEnvironment](#) API untuk membuat lingkungan komputasi yang menggunakan sumber daya Amazon EKS memerlukan izin untuk operasi API. `eks:DescribeCluster` Menggunakan AWS Management Console untuk membuat

sumber daya komputasi menggunakan sumber daya Amazon EKS memerlukan izin untuk keduanya `dan eks:DescribeCluster`. `eks:ListClusters`

Langkah 1: Mempersiapkan cluster Amazon EKS Anda AWS Batch

Semua langkah diperlukan.

1. Buat namespace khusus untuk pekerjaan AWS Batch

Gunakan `kubectl` untuk membuat namespace baru.

```
$ namespace=my-aws-batch-namespace
$ cat - <<EOF | kubectl create -f -
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
}
EOF
```

Output:

```
namespace/my-aws-batch-namespace created
```

2. Aktifkan akses melalui kontrol akses berbasis peran (RBAC)

Gunakan `kubectl` untuk membuat Kubernetes peran bagi klaster agar memungkinkan AWS Batch untuk menonton node dan pod, dan untuk mengikat peran. Anda harus melakukan ini sekali untuk setiap cluster EKS.

Note

Untuk informasi selengkapnya tentang penggunaan otorisasi RBAC, lihat [Menggunakan Otorisasi RBAC](#) di Panduan Pengguna. Kubernetes

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: ["" ]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: ["" ]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["" ]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["" ]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Output:

```
clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created
```

Buat Kubernetes peran cakupan namespace untuk mengelola dan mendaur hidup pod AWS Batch dan mengikatnya. Anda harus melakukan ini sekali untuk setiap namespace unik.

```
$ namespace=my-aws-batch-namespace
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["create", "get", "list", "watch", "delete", "patch"]
  - apiGroups: [""]
    resources: ["serviceaccounts"]
    verbs: ["get", "list"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["roles", "rolebindings"]
    verbs: ["get", "list"]
  ---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
  - kind: User
    name: aws-batch
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Output:

```
role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
created
```

Kubernetesaws-authPerbarui peta konfigurasi untuk memetakan izin RBAC sebelumnya ke peran terkait layanan. AWS Batch

```
$ eksctl create iamidentitymapping \
  --cluster my-cluster-name \
  --arn "arn:aws:iam::<your-account>:role/AWSServiceRoleForBatch" \
  --username aws-batch
```

Output:

```
2022-10-25 20:19:57 [#] adding identity "arn:aws:iam::<your-account>:role/
AWSServiceRoleForBatch" to auth ConfigMap
```

Note

Jalur `aws-service-role/batch.amazonaws.com/` telah dihapus dari ARN dari peran terkait layanan. Ini karena masalah dengan peta `aws-auth` konfigurasi. Untuk informasi selengkapnya, lihat [Peran dengan jalur tidak berfungsi saat jalur disertakan dalam ARN mereka di. aws-auth configmap](#)

Langkah 2: Membuat lingkungan komputasi Amazon EKS

AWS Batch lingkungan komputasi menentukan parameter sumber daya komputasi untuk memenuhi kebutuhan beban kerja batch Anda. Di lingkungan komputasi terkelola, AWS Batch membantu Anda mengelola kapasitas dan jenis instans sumber daya komputasi (Kubernetesnode) dalam kluster Amazon EKS Anda. Ini didasarkan pada spesifikasi sumber daya komputasi yang Anda tentukan ketika membuat lingkungan komputasi. Anda dapat menggunakan Instans Sesuai Permintaan EC2 atau Instans Spot EC2.

Sekarang peran `AWSServiceRoleForBatch` terkait layanan memiliki akses ke kluster Amazon EKS Anda, Anda dapat membuat AWS Batch sumber daya. Pertama, buat lingkungan komputasi yang mengarah ke cluster Amazon EKS Anda.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:123456789012:cluster/<cluster-name>",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 128,
    "instanceTypes": [
      "m5"
    ],
    "subnets": [
      "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
    ],
    "securityGroupIds": [
      "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
EOF
$ aws batch create-compute-environment --cli-input-json file:///./batch-eks-compute-environment.json
```

Catatan

- `serviceRoleParameter` tidak boleh ditentukan, maka peran AWS Batch terkait layanan akan digunakan. AWS Batch di Amazon EKS hanya mendukung peran AWS Batch terkait layanan.
- Hanya `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED`, dan strategi `SPOT_PRICE_CAPACITY_OPTIMIZED` alokasi yang didukung untuk lingkungan komputasi Amazon EKS.

Note

Kami menyarankan Anda menggunakan `SPOT_PRICE_CAPACITY_OPTIMIZED` daripada `SPOT_CAPACITY_OPTIMIZED` dalam kebanyakan kasus.

- Untuk mengetahui `instanceRole`, lihat [Membuat peran IAM node Amazon EKS](#) dan [Mengaktifkan akses utama IAM ke klaster Anda di Panduan Pengguna Amazon EKS](#). Jika Anda menggunakan jaringan pod, lihat [Mengonfigurasi plugin Amazon VPC CNI untuk menggunakan peran IAM Kubernetes untuk akun](#) layanan di Panduan Pengguna Amazon EKS.
- Cara untuk mendapatkan subnet yang berfungsi untuk subnet parameter tersebut adalah dengan menggunakan subnet publik grup node terkelola Amazon EKS yang dibuat oleh `eksctl` saat membuat cluster Amazon EKS. Jika tidak, gunakan subnet yang memiliki jalur jaringan yang mendukung penarikan gambar.
- `securityGroupIdsParameter` dapat menggunakan grup keamanan yang sama dengan cluster Amazon EKS. Perintah ini mengambil ID grup keamanan untuk cluster.

```
$ eks describe-cluster \  
  --name <cluster-name> \  
  --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- Pemeliharaan lingkungan komputasi Amazon EKS adalah tanggung jawab bersama. Untuk informasi selengkapnya, lihat [Tanggung jawab bersama dari Kubernetes node](#).

Important

Penting untuk memastikan bahwa lingkungan komputasi sehat sebelum melanjutkan. Operasi [DescribeComputeEnvironments](#) API dapat digunakan untuk melakukan ini.

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Konfirmasikan bahwa status parameternya tidak `INVALID`. Jika ya, lihat `statusReason` parameter penyebabnya. Untuk informasi selengkapnya, lihat [Pemecahan masalah AWS Batch](#).

Langkah 3: Buat antrian pekerjaan dan lampirkan lingkungan komputasi

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Pekerjaan yang dikirimkan ke antrian pekerjaan baru ini dijalankan sebagai pod pada node AWS Batch terkelola yang bergabung dengan kluster Amazon EKS yang terkait dengan lingkungan komputasi Anda.

```
$ cat <<EOF > ./batch-eks-job-queue.json
{
  "jobQueueName": "My-Eks-JQ1",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "My-Eks-CE1"
    }
  ]
}
EOF
$ aws batch create-job-queue --cli-input-json file://./batch-eks-job-queue.json
```

Langkah 4: Buat definisi pekerjaan

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "60"
          ],
          "resources": {
            "limits": {
              "cpu": "1",

```

```
        "memory": "1024Mi"
      }
    }
  ],
  "metadata": {
    "labels": {
      "environment": "test"
    }
  }
}
EOF
$ aws batch register-job-definition --cli-input-json file://./batch-eks-job-
definition.json
```

Catatan

- Hanya pekerjaan kontainer tunggal yang didukung.
- Ada pertimbangan untuk memory parameter cpu dan. Untuk informasi selengkapnya, lihat [Pertimbangan memori dan vCPU untuk Amazon AWS Batch EKS](#).

Langkah 5: Kirim pekerjaan

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \  
  --job-definition MyJobOnEks_Sleep --job-name My-Eks-Job1  
$ aws batch describe-jobs --job <jobId-from-submit-response>
```

Catatan

- Hanya pekerjaan kontainer tunggal yang didukung.
- Pastikan Anda terbiasa dengan semua pertimbangan yang relevan untuk memory parameter cpu dan. Untuk informasi selengkapnya, lihat [Pertimbangan memori dan vCPU untuk Amazon AWS Batch EKS](#).
- Untuk informasi selengkapnya tentang menjalankan pekerjaan di sumber daya Amazon EKS, lihat [Lowongan kerja Amazon EKS](#).

(Opsional) Kirim pekerjaan dengan penggantian

Pekerjaan ini mengesampingkan perintah yang diteruskan ke wadah.

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
            "echo hello world"
          ]
        }
      ]
    }
  }
}
EOF
$ aws batch submit-job --cli-input-json file:///./submit-job-override.json
```

Catatan

- AWS Batch secara agresif membersihkan pod setelah pekerjaan selesai untuk mengurangi beban. Kubernetes Untuk memeriksa detail pekerjaan, logging harus dikonfigurasi. Untuk informasi selengkapnya, lihat [Gunakan CloudWatch Log untuk memantau pekerjaan AWS Batch Amazon EKS](#).
- Untuk meningkatkan visibilitas ke detail operasi, aktifkan pencatatan pesawat kontrol Amazon EKS. Untuk informasi selengkapnya, lihat [pencatatan pesawat kontrol Amazon EKS](#) di Panduan Pengguna Amazon EKS.
- Daemonsets dan kubelets overhead mempengaruhi vCPU dan sumber daya memori yang tersedia, khususnya penskalaan dan penempatan kerja. Untuk informasi selengkapnya, lihat [Pertimbangan memori dan vCPU untuk Amazon AWS Batch EKS](#).

Memulai dengan AWS Batch di Amazon EKS Private Clusters

AWS Batch adalah layanan terkelola yang mengatur beban kerja batch di kluster Amazon Elastic Kubernetes Service (Amazon EKS) Anda. Ini termasuk antrian, pelacakan ketergantungan, percobaan ulang dan prioritas pekerjaan terkelola, manajemen pod, dan penskalaan node. Fitur ini menghubungkan kluster Amazon EKS pribadi Anda yang ada AWS Batch untuk menjalankan pekerjaan Anda dalam skala besar. Anda dapat menggunakan [eksctl](#) (antarmuka baris perintah untuk Amazon EKS), AWS konsol, atau [AWS Command Line Interface](#) untuk membuat cluster Amazon EKS pribadi dengan semua sumber daya lain yang diperlukan. Support untuk kluster Amazon EKS pribadi pada AWS Batch umumnya tersedia dalam [komersial Wilayah AWS jika AWS Batch](#) tersedia.

[Amazon EKS private only cluster](#) tidak memiliki akses internet inbound/outbound, dan hanya memiliki subnet pribadi. Titik akhir Amazon VPC digunakan untuk mengaktifkan akses pribadi ke layanan lain. AWS `eksctl` mendukung pembuatan cluster yang sepenuhnya pribadi menggunakan VPC dan subnet Amazon yang sudah ada sebelumnya. `eksctl` juga membuat titik akhir VPC Amazon di VPC Amazon yang disediakan dan memodifikasi tabel rute untuk subnet yang disediakan.

Setiap subnet harus memiliki tabel rute eksplisit yang terkait dengannya karena `eksctl` tidak mengubah tabel rute utama. [Cluster](#) Anda harus menarik gambar dari registri kontainer yang ada di VPC Amazon Anda. Selain itu, Anda dapat membuat Amazon Elastic Container Registry di Amazon VPC Anda dan menyalin gambar kontainer ke sana untuk diambil node Anda. Untuk informasi selengkapnya, lihat [Menyalin gambar kontainer dari satu repositori ke repositori lain](#). Untuk memulai repositori pribadi Amazon ECR, lihat repositori pribadi [Amazon ECR](#).

Anda dapat secara opsional membuat [aturan cache tarik melalui](#) dengan Amazon ECR. Setelah aturan cache pull through dibuat untuk registri publik eksternal, Anda dapat menarik gambar dari registri publik eksternal tersebut menggunakan Amazon ECR private registry uniform resource identifier (URI). Kemudian Amazon ECR membuat repositori dan menyimpan gambar. Ketika gambar yang di-cache ditarik menggunakan URI registri pribadi Amazon ECR, Amazon ECR memeriksa registri jarak jauh untuk melihat apakah ada versi baru gambar dan memperbarui registri pribadi Anda hingga sekali setiap 24 jam.

Daftar Isi

- [Prasyarat](#)
- [Langkah 1: Mempersiapkan cluster EKS Anda AWS Batch](#)
- [Langkah 2: Membuat lingkungan komputasi Amazon EKS](#)
- [Langkah 3: Buat antrian pekerjaan dan lampirkan lingkungan komputasi](#)

- [Langkah 4: Buat definisi pekerjaan](#)
- [Langkah 5: Kirim pekerjaan](#)
- [\(Opsional\) Kirim pekerjaan dengan penggantian](#)
- [Pemecahan Masalah](#)

Prasyarat

Sebelum memulai tutorial ini, Anda harus menginstal dan mengkonfigurasi alat dan sumber daya berikut yang Anda butuhkan untuk membuat AWS Batch dan mengelola sumber daya Amazon EKS. Anda juga perlu membuat semua sumber daya yang diperlukan termasuk VPC, subnet, tabel rute, titik akhir VPC, dan cluster Amazon EKS. Anda perlu menggunakan AWS CLI.

- **AWS CLI**— Alat baris perintah untuk bekerja dengan AWS layanan, termasuk Amazon EKS. Panduan ini mengharuskan Anda menggunakan versi 2.8.6 atau yang lebih baru atau 1.26.0 atau yang lebih baru. Untuk informasi selengkapnya, lihat [Menginstal, memperbarui, dan mencopot instalasi AWS CLI](#) di Panduan Pengguna AWS Command Line Interface .

Setelah menginstal AWS CLI, kami sarankan Anda mengkonfigurasinya. Untuk informasi selengkapnya, lihat [Konfigurasi cepat dengan aws configure](#) di Panduan AWS Command Line Interface Pengguna.

- **kubectl**— Alat baris perintah untuk bekerja dengan Kubernetes cluster. Panduan ini mengharuskan Anda menggunakan versi 1.23 atau yang lebih baru. Untuk informasi selengkapnya, silakan lihat [Menginstal atau memperbarui kubectl](#) di Panduan Pengguna Amazon EKS.
- **eksctl**— Alat baris perintah untuk bekerja dengan kluster Amazon EKS yang mengotomatiskan banyak tugas individu. Panduan ini mengharuskan Anda menggunakan versi 0.115.0 atau yang lebih baru. Untuk informasi selengkapnya, silakan lihat [Menginstal atau memperbarui eksctl](#) di Panduan Pengguna Amazon EKS.
- Izin Wajib AWS Identity and Access Management (IAM) — Prinsip keamanan IAM yang Anda gunakan harus memiliki izin untuk bekerja dengan peran Amazon EKS IAM dan peran terkait layanan, serta VPC AWS CloudFormation dan sumber daya terkait. Untuk informasi selengkapnya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk Amazon Elastic Kubernetes Service dan Menggunakan peran terkait](#) layanan di Panduan Pengguna IAM. Anda harus menyelesaikan semua langkah dalam panduan ini sebagai pengguna yang sama.
- Membuat kluster Amazon EKS — Untuk informasi selengkapnya, lihat [Memulai Amazon EKS — eksctl](#) di Panduan Pengguna Amazon EKS.

Note

AWS Batch tidak menyediakan orkestrasi node terkelola untuk CoreDNS atau pod penerapan lainnya. Jika Anda membutuhkan CoreDNS, lihat [Menambahkan add-on CoreDNS Amazon EKS di Panduan Pengguna Amazon EKS](#). Atau, gunakan `eksctl create cluster create` untuk membuat cluster, itu termasuk CoreDNS secara default.

- Izin — Pengguna yang memanggil operasi [CreateComputeEnvironment](#) API untuk membuat lingkungan komputasi yang menggunakan sumber daya Amazon EKS memerlukan izin untuk operasi API. `eks:DescribeCluster` Menggunakan AWS Management Console untuk membuat sumber daya komputasi menggunakan sumber daya Amazon EKS memerlukan izin untuk keduanya dan `eks:DescribeCluster. eks:ListClusters`
- Buat cluster [EKS pribadi](#) di wilayah us-east-1 menggunakan contoh `eksctl` file konfigurasi.

```
kind: ClusterConfig
apiVersion: eksctl.io/v1alpha5
availabilityZones:
  - us-east-1a
  - us-east-1b
  - us-east-1d
managedNodeGroups:
  privateNetworking: true
privateCluster:
  enabled: true
  skipEndpointCreation: false
```

Buat sumber daya Anda menggunakan perintah: `eksctl create cluster -f clusterConfig.yaml`

- Node terkelola batch harus disebar ke subnet yang memiliki titik akhir antarmuka VPC yang Anda butuhkan. Untuk informasi selengkapnya, lihat [Persyaratan klaster pribadi](#).

Langkah 1: Mempersiapkan cluster EKS Anda AWS Batch

Semua langkah diperlukan.

1. Buat namespace khusus untuk pekerjaan AWS Batch

Gunakan `kubectl` untuk membuat namespace baru.

```
$ namespace=my-aws-batch-namespace
$ cat - <<EOF | kubectl create -f -
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
}
EOF
```

Output:

```
namespace/my-aws-batch-namespace created
```

2. Aktifkan akses melalui kontrol akses berbasis peran (RBAC)

Gunakan `kubectl` untuk membuat Kubernetes peran bagi kluster agar memungkinkan AWS Batch untuk menonton node dan pod, dan untuk mengikat peran. Anda harus melakukan ini sekali untuk setiap cluster Amazon EKS.

Note

Untuk informasi selengkapnya tentang penggunaan otorisasi RBAC, lihat [Menggunakan Otorisasi RBAC](#) dalam dokumentasi. Kubernetes

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
- apiGroups: [""]
  resources: ["namespaces"]
  verbs: ["get"]
- apiGroups: [""]
```



```

resources: ["nodes"]
verbs: ["get", "list", "watch"]
- apiGroups: [""]
resources: ["pods"]
verbs: ["get", "list", "watch"]
- apiGroups: [""]
resources: ["configmaps"]
verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
verbs: ["get", "list", "watch"]
- apiGroups: ["rbac.authorization.k8s.io"]
resources: ["clusterroles", "clusterrolebindings"]
verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF

```

Output:

```

clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created

```

Buat Kubernetes peran cakupan namespace untuk mengelola dan mendaur hidup pod AWS Batch dan mengikatnya. Anda harus melakukan ini sekali untuk setiap namespace unik.

```

$ namespace=my-aws-batch-namespace
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:

```

```

name: aws-batch-compute-environment-role
namespace: ${namespace}
rules:
- apiGroups: ["" ]
  resources: ["pods"]
  verbs: ["create", "get", "list", "watch", "delete", "patch"]
- apiGroups: ["" ]
  resources: ["serviceaccounts"]
  verbs: ["get", "list"]
- apiGroups: ["rbac.authorization.k8s.io"]
  resources: ["roles", "rolebindings"]
  verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF

```

Output:

```

role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
created

```

Kubernetesaws-authPerbarui peta konfigurasi untuk memetakan izin RBAC sebelumnya ke peran terkait layanan. AWS Batch

```

$ eksctl create iamidentitymapping \
  --cluster my-cluster-name \
  --arn "arn:aws:iam::<your-account>:role/AWSServiceRoleForBatch" \
  --username aws-batch

```

Output:

```
2022-10-25 20:19:57 [#] adding identity "arn:aws:iam::<your-account>:role/AWSServiceRoleForBatch" to auth ConfigMap
```

Note

Jalur `aws-service-role/batch.amazonaws.com/` telah dihapus dari ARN dari peran terkait layanan. Ini karena masalah dengan peta `aws-auth` konfigurasi. Untuk informasi selengkapnya, lihat [Peran dengan jalur tidak berfungsi saat jalur disertakan dalam ARN mereka di. aws-auth configmap](#)

Langkah 2: Membuat lingkungan komputasi Amazon EKS

AWS Batch lingkungan komputasi menentukan parameter sumber daya komputasi untuk memenuhi kebutuhan beban kerja batch Anda. Di lingkungan komputasi terkelola, AWS Batch membantu Anda mengelola kapasitas dan jenis instans sumber daya komputasi (Kubernetesnode) dalam kluster Amazon EKS Anda. Ini didasarkan pada spesifikasi sumber daya komputasi yang Anda tentukan ketika membuat lingkungan komputasi. Anda dapat menggunakan Instans Sesuai Permintaan EC2 atau Instans Spot EC2.

Sekarang peran `AWSServiceRoleForBatch` terkait layanan memiliki akses ke kluster Amazon EKS Anda, Anda dapat membuat AWS Batch sumber daya. Pertama, buat lingkungan komputasi yang mengarah ke cluster Amazon EKS Anda.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:123456789012:cluster/<cluster-name>",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
```

```

    "maxvCpus": 128,
    "instanceTypes": [
        "m5"
    ],
    "subnets": [
        "<eks-cluster-subnets-with-access-to-the-image-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
}
}
EOF
$ aws batch create-compute-environment --cli-input-json file://./batch-eks-compute-environment.json

```

Catatan

- `serviceRoleParameter` tidak boleh ditentukan, maka peran AWS Batch terkait layanan akan digunakan. AWS Batch di Amazon EKS hanya mendukung peran AWS Batch terkait layanan.
- Hanya `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED`, dan strategi `SPOT_PRICE_CAPACITY_OPTIMIZED` alokasi yang didukung untuk lingkungan komputasi Amazon EKS.

Note

Kami menyarankan Anda menggunakan `SPOT_PRICE_CAPACITY_OPTIMIZED` bukan `SPOT_CAPACITY_OPTIMIZED` dalam banyak kasus.

- Untuk mengetahui `instanceRole`, lihat [Membuat peran IAM node Amazon EKS](#) dan [Mengaktifkan akses utama IAM ke kluster Anda di Panduan Pengguna Amazon EKS](#). Jika Anda menggunakan jaringan pod, lihat [Mengonfigurasi plugin Amazon VPC CNI untuk menggunakan peran IAM Kubernetes untuk akun](#) layanan di Panduan Pengguna Amazon EKS.
- Cara untuk mendapatkan subnet yang berfungsi untuk `subnets` parameter tersebut adalah dengan menggunakan subnet publik grup node terkelola Amazon EKS yang dibuat oleh `eksctl` saat membuat cluster Amazon EKS. Jika tidak, gunakan subnet yang memiliki jalur jaringan yang mendukung penarikan gambar.

- `securityGroupIdsParameter` dapat menggunakan grup keamanan yang sama dengan cluster Amazon EKS. Perintah ini mengambil ID grup keamanan untuk cluster.

```
$ eks describe-cluster \
  --name <cluster-name> \
  --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- Pemeliharaan lingkungan komputasi Amazon EKS adalah tanggung jawab bersama. Untuk informasi selengkapnya, lihat [Keamanan di Amazon EKS](#).

Important

Penting untuk memastikan bahwa lingkungan komputasi sehat sebelum melanjutkan. Operasi [DescribeComputeEnvironments](#) API dapat digunakan untuk melakukan ini.

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Konfirmasikan bahwa status parameternya tidak `INVALID`. Jika ya, lihat `statusReason` parameter penyebabnya. Untuk informasi selengkapnya, lihat [Pemecahan masalah AWS Batch](#).

Langkah 3: Buat antrian pekerjaan dan lampirkan lingkungan komputasi

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Pekerjaan yang dikirimkan ke antrian pekerjaan baru ini dijalankan sebagai pod pada node AWS Batch terkelola yang bergabung dengan kluster Amazon EKS yang terkait dengan lingkungan komputasi Anda.

```
$ cat <<EOF > ./batch-eks-job-queue.json
{
  "jobQueueName": "My-Eks-JQ1",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "My-Eks-CE1"
    }
  ]
}
```

```
    ]
  }
EOF
$ aws batch create-job-queue --cli-input-json file:///./batch-eks-job-queue.json
```

Langkah 4: Buat definisi pekerjaan

Di bidang gambar definisi pekerjaan, alih-alih menyediakan tautan ke gambar di repositori ECR publik, berikan tautan ke gambar yang disimpan di repositori ECR pribadi kami. Lihat contoh definisi pekerjaan berikut:

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "account-id.dkr.ecr.region.amazonaws.com/amazonlinux:2",
          "command": [
            "sleep",
            "60"
          ],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ],
      "metadata": {
        "labels": {
          "environment": "test"
        }
      }
    }
  }
}
EOF
```

```
$ aws batch register-job-definition --cli-input-json file:///./batch-eks-job-
definition.json
```

Untuk menjalankan perintah `kubectl`, Anda memerlukan akses pribadi ke kluster Amazon EKS Anda. Ini berarti semua lalu lintas ke server API cluster Anda harus berasal dari dalam VPC kluster Anda atau [jaringan yang terhubung](#).

Langkah 5: Kirim pekerjaan

```
$ aws batch submit-job - -job-queue My-Eks-JQ1 \
  - -job-definition MyJobOnEks_Sleep - -job-name My-Eks-Job1
$ aws batch describe-jobs - -job <jobId-from-submit-response>
```

Catatan

- Hanya pekerjaan kontainer tunggal yang didukung.
- Pastikan Anda terbiasa dengan semua pertimbangan yang relevan untuk memory parameter cpu dan. Untuk informasi selengkapnya, lihat [Pertimbangan memori dan vCPU untuk Amazon AWS Batch EKS](#).
- Untuk informasi selengkapnya tentang menjalankan pekerjaan di sumber daya Amazon EKS, lihat [Lowongan kerja Amazon EKS](#).

(Opsional) Kirim pekerjaan dengan penggantian

Pekerjaan ini mengesampingkan perintah yang diteruskan ke wadah.

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
```

```
        "-c",
        "echo hello world"
    ]
}
]
}
}
EOF
$ aws batch submit-job - -cli-input-json file:///./submit-job-override.json
```

Catatan

- AWS Batch secara agresif membersihkan pod setelah pekerjaan selesai untuk mengurangi beban. Kubernetes Untuk memeriksa detail pekerjaan, logging harus dikonfigurasi. Untuk informasi selengkapnya, lihat [Gunakan CloudWatch Log untuk memantau pekerjaan AWS Batch Amazon EKS](#).
- Untuk meningkatkan visibilitas ke detail operasi, aktifkan pencatatan pesawat kontrol Amazon EKS. Untuk informasi selengkapnya, lihat [pencatatan pesawat kontrol Amazon EKS](#) di Panduan Pengguna Amazon EKS.
- Daemonsets dan kubelets overhead mempengaruhi vCPU dan sumber daya memori yang tersedia, khususnya penskalaan dan penempatan kerja. Untuk informasi selengkapnya, lihat [Pertimbangan memori dan vCPU untuk Amazon AWS Batch EKS](#).

Pemecahan Masalah

Jika node yang diluncurkan oleh AWS Batch tidak memiliki akses ke repositori Amazon ECR (atau repositori lain) yang menyimpan gambar Anda, maka pekerjaan Anda dapat tetap dalam status MULAI. Ini karena pod tidak akan dapat mengunduh gambar dan menjalankan AWS Batch pekerjaan Anda. Jika Anda mengklik nama pod yang diluncurkan oleh AWS Batch Anda akan dapat melihat pesan kesalahan dan mengonfirmasi masalah. Pesan kesalahan akan terlihat mirip dengan yang berikut:

```
Failed to pull image "public.ecr.aws/amazonlinux/amazonlinux:2": rpc error: code =
Unknown desc = failed to pull and unpack image
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to resolve reference
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to do request: Head
"https://public.ecr.aws/v2/amazonlinux/amazonlinux/manifests/2": dial tcp: i/o timeout
```

Untuk skenario pemecahan masalah umum lainnya, lihat [Pemecahan masalah. AWS Batch](#) Untuk mengatasi masalah berdasarkan status pod, lihat [Bagaimana cara memecahkan masalah status pod di Amazon EKS?](#) .

Tugas

Pekerjaan adalah unit kerja yang dimulai oleh AWS Batch. Pekerjaan dapat dipanggil sebagai aplikasi kontainer yang berjalan di instans penampung Amazon ECS di cluster ECS.

Tugas dalam kontainer dapat mereferensikan citra kontainer, perintah, dan parameter. Untuk informasi selengkapnya, lihat [Parameter definisi Job untuk ContainerProperties](#).

Anda dapat mengirimkan sejumlah besar tugas independen dan sederhana.

Topik

- [Mengirimkan pekerjaan](#)
- [Status Job](#)
- [AWS Batch variabel lingkungan kerja](#)
- [Percobaan kembali tugas secara otomatis](#)
- [Dependensi Job](#)
- [Batas waktu Job](#)
- [Lowongan kerja Amazon EKS](#)
- [Tugas array](#)
- [Tugas paralel multisimpul](#)
- [Lowongan kerja GPU](#)
- [Untuk membuat pekerjaan berbasis GPU di sumber daya Amazon EKS](#)
- [Cari dan filter AWS Batch pekerjaan](#)
- [Log Job](#)
- [Informasi Job](#)


Mengirimkan pekerjaan

Setelah Anda mendaftarkan definisi pekerjaan, Anda dapat mengirimkannya sebagai pekerjaan ke antrian AWS Batch pekerjaan. Anda dapat mengganti banyak parameter yang ditentukan dalam definisi pekerjaan saat runtime.

Untuk mengirim tugas


1. Buka AWS Batch konsol di <https://console.aws.amazon.com/batch/>.

2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Tugas.
4. Pilih Kirim pekerjaan baru.
5. Untuk Name (Name), masukkan nama yang unik untuk ketentuan tugas Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
6. Untuk definisi Job, pilih definisi pekerjaan yang ada untuk pekerjaan Anda. Untuk informasi selengkapnya, lihat [Membuat definisi pekerjaan simpul tunggal](#).
7. Untuk antrean Job, pilih antrean pekerjaan yang ada. Untuk informasi selengkapnya, lihat [Membuat antrean tugas](#).
8. Untuk dependensi Job, pilih Add Job dependencies.
 - Untuk id Job, masukkan ID pekerjaan untuk dependensi apa pun. Kemudian pilih Tambahkan dependensi pekerjaan. Sebuah pekerjaan dapat memiliki hingga 20 dependensi. Untuk informasi selengkapnya, lihat [Dependensi Job](#).
9. (Khusus tugas array) Untuk Array size (Ukuran array), tentukan ukuran array antara 2 hingga 10.000.
10. (Opsional) Perluas Tag dan kemudian pilih Tambahkan tag untuk menambahkan tag ke sumber daya. Masukkan kunci dan nilai opsional, lalu pilih Tambah tag.
11. Pilih Halaman berikutnya.
12. Di bagian Job overrides:
 - a. (Opsional) Untuk prioritas Penjadwalan, masukkan nilai prioritas penjadwalan antara 0 dan 100. Nilai yang lebih tinggi diberikan prioritas yang lebih tinggi.
 - b. (Opsional) Untuk upaya Job, masukkan jumlah maksimum kali yang AWS Batch mencoba memindahkan pekerjaan ke RUNNABLE status. Anda dapat memasukkan angka antara 1 dan 10. Untuk informasi selengkapnya, lihat [Percobaan kembali tugas secara otomatis](#).
 - c. (Opsional) Untuk batas waktu Eksekusi, masukkan nilai batas waktu (dalam detik). Batas waktu eksekusi adalah lamanya waktu sebelum pekerjaan yang belum selesai dihentikan. Jika upaya melebihi durasi batas waktu, itu dihentikan dan dipindahkan ke FAILED status. Untuk informasi selengkapnya, lihat [Batas waktu Job](#). Nilai minimum adalah 60 detik.

 Important


Jangan mengandalkan pekerjaan yang berjalan pada sumber daya Fargate untuk berjalan selama lebih dari 14 hari. Setelah 14 hari, sumber daya Fargate mungkin tidak lagi tersedia dengan pekerjaan yang kemungkinan diberhentikan.

- d. (Opsional) Aktifkan Propagate tag untuk menyebarkan tag dari definisi pekerjaan dan pekerjaan ke tugas Amazon ECS.
13. Perluas Additional configuration (Konfigurasi tambahan).
 14. (Opsional) Untuk kondisi strategi Coba lagi, pilih Tambahkan evaluasi saat keluar. Masukkan setidaknya satu nilai parameter dan kemudian pilih Action. Untuk setiap rangkaian kondisi, Tindakan harus diatur ke Coba Ulang atau Keluar. Tindakan ini berarti sebagai berikut:
 - Coba lagi — AWS Batch coba lagi sampai jumlah upaya pekerjaan yang Anda tentukan tercapai.
 - Keluar — AWS Batch berhenti mencoba kembali pekerjaan.

 Important

Jika Anda memilih Tambahkan evaluasi saat keluar, konfigurasi setidaknya satu parameter dan pilih salah satu Tindakan atau pilih Hapus evaluasi saat keluar.

15. Untuk Parameter, pilih Tambahkan parameter untuk menambahkan placeholder substitusi parameter. Kemudian, masukkan Kunci dan Nilai opsional.
16. Di bagian Container overrides:
 - a. Untuk Command (Perintah), tentukan perintah untuk diteruskan ke kontainer. Untuk perintah sederhana, masukkan perintah seperti yang Anda lakukan untuk command prompt. Untuk perintah yang lebih rumit, misalnya dengan karakter khusus), gunakan sintaks JSON.


 Note

Parameter ini tidak dapat berisi string kosong.

- b. Untuk vCPU, masukkan jumlah vCPU yang akan dicadangkan untuk wadah. Parameter ini sesuai dengan CpuShares di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi --

cpu-shares untuk [docker run](#). Setiap vCPU setara dengan 1.024 bagian CPU. Anda harus menentukan setidaknya satu vCPU.

- c. Untuk Memori, masukkan batas memori yang tersedia untuk wadah. Jika kontainer Anda mencoba untuk melebihi memori yang ditentukan di sini, kontainer akan dihentikan. Parameter ini sesuai dengan Memory di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--memory` untuk [docker run](#). Anda harus menentukan sedikitnya 4 MiB memori untuk sebuah tugas.

 Note

Untuk memaksimalkan pemanfaatan sumber daya Anda, prioritaskan memori untuk pekerjaan dari jenis instans tertentu. Untuk informasi selengkapnya, lihat [Manajemen Memori Sumber Daya Komputasi](#).

- d. (Opsional) Untuk Jumlah GPU, pilih jumlah GPU yang akan dicadangkan untuk kontainer.
- e. (Opsional) Untuk variabel Lingkungan, pilih Tambahkan variabel lingkungan untuk menambahkan variabel lingkungan sebagai pasangan nama-nilai. Variabel-variabel ini diteruskan ke wadah.
- f. Pilih Halaman berikutnya.
- g. Untuk tinjauan Job, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat definisi pekerjaan.

Status Job

Ketika Anda mengirimkan pekerjaan ke antrian AWS Batch pekerjaan, pekerjaan memasuki SUBMITTED negara bagian. Tugas kemudian melewati status-status berikut sampai berhasil (keluar dengan kode 0) atau gagal (keluar dengan kode bukan nol). Tugas AWS Batch dapat memiliki status sebagai berikut:

SUBMITTED

Pekerjaan yang dikirimkan ke antrian, dan belum dievaluasi oleh penjadwal. Penjadwal mengevaluasi tugas untuk menentukan apakah tugas memiliki dependensi yang belum diselesaikan dengan keberhasilan penyelesaian tugas lain. Jika ada dependensi, tugas dipindahkan ke PENDING. Jika tidak ada dependensi, tugas dipindahkan ke RUNNABLE.

PENDING

Tugas yang berada dalam antrean dan belum dapat berjalan karena dependensi dengan tugas atau sumber daya lain. Setelah dependensi diselesaikan, tugas dipindahkan ke `RUNNABLE`.

RUNNABLE

Tugas yang berada dalam antrean, tidak memiliki dependensi yang belum diselesaikan, dan karena itu siap dijadwalkan untuk host. Tugas dalam status ini dimulai segera setelah ada cukup sumber daya yang tersedia di salah satu lingkungan komputasi yang dipetakan ke antrean tugas. Namun, tugas dapat tetap berada dalam status ini tanpa batas waktu ketika sumber daya yang cukup tidak tersedia.

Note

Jika pekerjaan Anda tidak berkembang `STARTING`, lihat [Pekerjaan terjebak dalam `RUNNABLE` status](#) di bagian pemecahan masalah.

STARTING

Tugas ini telah dijadwalkan untuk host dan operasi inisiasi kontainer yang relevan sedang berlangsung. Setelah citra kontainer ditarik dan kontainer sudah berjalan, tugas bertransisi ke `RUNNING`.

RUNNING

Tugas berjalan sebagai tugas kontainer di instans kontainer Amazon ECS dalam lingkungan komputasi. Ketika tugas kontainer keluar, kode keluar proses menentukan apakah tugas berhasil atau gagal. Kode keluar `0` menunjukkan keberhasilan, dan kode keluar bukan nol menunjukkan kegagalan. Jika tugas yang terkait dengan upaya yang gagal memiliki sisa upaya yang tersisa dalam konfigurasi strategi percobaan kembali opsional, tugas dipindahkan ke `RUNNABLE` lagi. Untuk informasi selengkapnya, lihat [Percobaan kembali tugas secara otomatis](#).

Note

Log untuk `RUNNING` pekerjaan tersedia di CloudWatch Log. Grup log adalah `/aws/batch/job`, dan format nama aliran log adalah sebagai berikut: `first200CharsOfJobDefinitionName/default/ecs_task_id`. Format ini mungkin berubah di masa depan.

Setelah pekerjaan mencapai RUNNING status, Anda dapat mengambil nama aliran lognya secara terprogram dengan operasi API. [DescribeJobs](#) Untuk informasi selengkapnya, lihat [Melihat Data Log yang Dikirim ke CloudWatch Log](#) di Panduan Pengguna CloudWatch Log Amazon. Secara default, log ini tidak pernah kedaluwarsa. Namun, Anda dapat mengubah periode retensi. Untuk informasi selengkapnya, lihat [Mengubah Penyimpanan Data CloudWatch Log di Log](#) di Panduan Pengguna CloudWatch Log Amazon.

SUCCEEDED

Tugas telah berhasil diselesaikan dengan kode keluar 0. Status pekerjaan untuk SUCCEEDED pekerjaan dipertahankan setidaknya AWS Batch selama 7 hari.

Note

Log untuk SUCCEEDED pekerjaan tersedia di CloudWatch Log. Grup log adalah `/aws/batch/job`, dan format nama aliran log adalah sebagai berikut: `first200CharsOfJobDefinitionName/default/ecs_task_id`. Format ini dapat berubah di masa depan.

Setelah pekerjaan mencapai RUNNING status, Anda dapat mengambil nama aliran lognya secara terprogram dengan operasi API. [DescribeJobs](#) Untuk informasi selengkapnya, lihat [Melihat Data Log yang Dikirim ke CloudWatch Log](#) di Panduan Pengguna CloudWatch Log Amazon. Secara default, log ini tidak pernah kedaluwarsa. Namun, Anda dapat mengubah periode retensi. Untuk informasi selengkapnya, lihat [Mengubah Penyimpanan Data CloudWatch Log di Log](#) di Panduan Pengguna CloudWatch Log Amazon.

FAILED

Tugas telah gagal dalam semua upaya yang tersedia. Status pekerjaan untuk FAILED pekerjaan dipertahankan setidaknya AWS Batch selama 7 hari.

Note

Log untuk FAILED pekerjaan tersedia di CloudWatch Log. Grup log adalah `/aws/batch/job`, dan format nama aliran log adalah sebagai berikut: `first200CharsOfJobDefinitionName/default/ecs_task_id`. Format ini dapat berubah di masa depan.

Setelah pekerjaan mencapai RUNNING status, Anda dapat mengambil aliran lognya secara terprogram dengan operasi API. [DescribeJobs](#) Untuk informasi selengkapnya, lihat [Melihat Data Log yang Dikirim ke CloudWatch Log](#) di Panduan Pengguna CloudWatch Log Amazon. Secara default, log ini tidak pernah kedaluwarsa. Namun, Anda dapat mengubah periode retensi. Untuk informasi selengkapnya, lihat [Mengubah Penyimpanan Data CloudWatch Log di Log](#) di Panduan Pengguna CloudWatch Log Amazon.

AWS Batch variabel lingkungan kerja

AWS Batch menetapkan variabel lingkungan tertentu dalam pekerjaan kontainer. Variabel lingkungan ini memberikan introspeksi untuk wadah di dalam pekerjaan. Anda dapat menggunakan nilai-nilai variabel-variabel ini dalam logika aplikasi Anda. Semua variabel yang AWS Batch ditetapkan dimulai dengan `AWS_BATCH_` awalan. Ini adalah awalan variabel lingkungan yang dilindungi. Anda tidak dapat menggunakan awalan ini untuk variabel Anda sendiri dalam definisi pekerjaan atau penggantian.

Variabel lingkungan berikut tersedia dalam kontainer tugas:

`AWS_BATCH_CE_NAME`

Variabel ini diatur ke nama lingkungan komputasi tempat pekerjaan Anda ditempatkan.

`AWS_BATCH_JOB_ARRAY_INDEX`

Variabel ini hanya ditetapkan dalam tugas array turunan. Indeks tugas array dimulai dari 0, dan setiap tugas turunan menerima nomor indeks yang unik. Misalnya, tugas array dengan 10 turunan memiliki nilai indeks 0-9. Anda dapat menggunakan nilai indeks ini untuk mengontrol cara turunan tugas array Anda dibedakan. Untuk informasi selengkapnya, lihat [Tutorial: Menggunakan indeks pekerjaan array untuk mengontrol diferensiasi pekerjaan](#).

`AWS_BATCH_JOB_ARRAY_SIZE`

Variabel ini diatur ke ukuran pekerjaan array induk. Ukuran pekerjaan array induk diteruskan ke pekerjaan array anak dalam variabel ini.

`AWS_BATCH_JOB_ATTEMPT`

Variabel ini ditetapkan ke nomor upaya tugas. Upaya pertama diberi nomor 1. Untuk informasi selengkapnya, lihat [Percobaan kembali tugas secara otomatis](#).

AWS_BATCH_JOB_ID

Variabel ini diatur ke ID AWS Batch pekerjaan.

AWS_BATCH_JOB_KUBERNETES_NODE_UID

Variabel ini ditetapkan sebagai Kubernetes UID dari objek node yang ada di cluster Kubernetes tempat pod berjalan. Variabel ini hanya disetel untuk pekerjaan yang berjalan di sumber daya Amazon EKS. Untuk informasi selengkapnya, lihat [UID](#) dalam Kubernetes dokumentasi.

AWS_BATCH_JOB_MAIN_NODE_INDEX

Variabel ini hanya ditetapkan dalam tugas paralel multisimpul. Variabel ini ditetapkan ke nomor indeks simpul utama tugas. Kode aplikasi Anda dapat AWS_BATCH_JOB_MAIN_NODE_INDEX membandingkan dengan AWS_BATCH_JOB_NODE_INDEX pada node individual untuk menentukan apakah itu node utama.

AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS

Variabel ini hanya diatur dalam node anak pekerjaan paralel multi-node. Variabel ini tidak ada pada node utama, tetapi diatur ke alamat IPv4 pribadi dari node utama pekerjaan. Kode aplikasi simpul turunan Anda dapat menggunakan alamat ini untuk berkomunikasi dengan simpul utama.

AWS_BATCH_JOB_NODE_INDEX

Variabel ini hanya ditetapkan dalam tugas paralel multisimpul. Variabel ini ditetapkan ke nomor indeks simpul pada simpul. Indeks simpul dimulai dari 0, dan setiap simpul menerima nomor indeks yang unik. Misalnya, tugas paralel multisimpul dengan 10 turunan memiliki nilai indeks 0-9.

AWS_BATCH_JOB_NUM_NODES

Variabel ini hanya ditetapkan dalam tugas paralel multisimpul. Variabel ini diatur ke jumlah node yang Anda minta untuk pekerjaan paralel multi-node Anda.

AWS_BATCH_JQ_NAME

Variabel ini diatur ke nama antrian pekerjaan tempat pekerjaan Anda dikirimkan.

Percobaan kembali tugas secara otomatis

Anda dapat menerapkan strategi percobaan kembali untuk tugas dan ketentuan tugas Anda yang memungkinkan tugas gagal untuk dicoba ulang secara otomatis. Skenario kegagalan yang mungkin termasuk yang berikut:

- Kode keluar yang bukan nol dari tugas kontainer
- Kegagalan atau pengakhiran instans Amazon EC2
- Kesalahan atau pemadaman AWS layanan internal

Ketika pekerjaan diserahkan ke antrian pekerjaan dan ditempatkan ke RUNNING negara bagian yang dianggap sebagai upaya. Secara default, setiap tugas diberikan satu upaya untuk berpindah, baik ke status tugas SUCCEEDED maupun FAILED. Namun, definisi pekerjaan dan alur kerja pengiriman pekerjaan dapat digunakan untuk menentukan strategi coba lagi dengan antara 1 dan 10 upaya. Jika [evaluateOnExit](#) ditentukan, dapat berisi hingga 5 strategi coba lagi. Jika [evaluateOnExit](#) ditentukan, tetapi tidak ada strategi coba lagi yang cocok, maka pekerjaan itu dicoba lagi. Untuk pekerjaan yang tidak cocok untuk keluar, tambahkan entri akhir yang keluar karena alasan apa pun. Misalnya, `evaluateOnExit` objek ini memiliki dua entri yang dengan tindakan RETRY dan entri akhir dengan tindakan. EXIT

```
"evaluateOnExit": [  
  {  
    "action": "RETRY",  
    "onReason": "AGENT"  
  },  
  {  
    "action": "RETRY",  
    "onStatusReason": "Task failed to start"  
  },  
  {  
    "action": "EXIT",  
    "onReason": "*"   
  }  
]
```

Pada saat waktu aktif, variabel lingkungan `AWS_BATCH_JOB_ATTEMPT` diatur ke jumlah upaya tugas yang sesuai dengan kontainer. Upaya pertama diberi nomor1, dan upaya selanjutnya dalam urutan menaik (misalnya, 2, 3, 4).

Misalnya, anggaplah upaya pekerjaan gagal karena alasan apa pun dan jumlah upaya yang ditentukan dalam konfigurasi coba lagi lebih besar dari `AWS_BATCH_JOB_ATTEMPT` jumlahnya. Kemudian, pekerjaan ditempatkan kembali di RUNNABLE negara bagian. Untuk informasi selengkapnya, lihat [Status Job](#).

Note

Pekerjaan yang dibatalkan atau dihentikan tidak akan dicoba lagi. Juga, pekerjaan yang gagal karena definisi pekerjaan yang tidak valid tidak dicoba lagi.

Untuk informasi selengkapnya [Strategi percobaan kembali](#), lihat [Membuat definisi pekerjaan simpel tunggal](#), [Mengirimkan pekerjaan](#) dan [Kode kesalahan tugas berhenti](#).

Dependensi Job

Saat Anda mengirimkan AWS Batch pekerjaan, Anda dapat menentukan ID pekerjaan yang bergantung pada pekerjaan itu. Ketika Anda melakukannya, penjadwal AWS Batch memastikan bahwa tugas Anda dijalankan hanya setelah dependensi yang ditentukan berhasil diselesaikan. Setelah dependensi berhasil, tugas yang bergantung akan bertransisi dari PENDING ke RUNNABLE dan kemudian ke STARTING dan RUNNING. Jika salah satu dependensi tugas gagal, tugas yang bergantung secara otomatis akan bertransisi dari PENDING ke FAILED.

Misalnya, Tugas A dapat mengekspresikan dependensi pada maksimum 20 tugas lain yang harus berhasil sebelum dapat Tugas A dapat berjalan. Anda kemudian dapat mengirimkan tugas tambahan yang memiliki dependensi pada Tugas A dan maksimum 19 tugas lainnya.

Untuk tugas array, Anda dapat menentukan tipe dependensi SEQUENTIAL tanpa menentukan ID tugas sehingga setiap tugas array turunan selesai secara berurutan, mulai dari indeks 0. Anda juga dapat menentukan tipe dependensi N_TO_N dengan ID tugas. Dengan cara itu, setiap turunan indeks dari tugas ini harus menunggu turunan indeks dari setiap dependensi selesai sebelum turunan tersebut dapat dimulai. Untuk informasi selengkapnya, lihat [Tugas array](#).

Untuk mengirimkan AWS Batch pekerjaan dengan dependensi, lihat [Mengirimkan pekerjaan](#)

Batas waktu Job

Anda dapat mengonfigurasi durasi batas waktu untuk pekerjaan Anda sehingga jika pekerjaan berjalan lebih lama dari itu, AWS Batch mengakhiri pekerjaan. Misalnya, Anda mungkin memiliki tugas yang Anda tahu hanya perlu waktu 15 menit untuk diselesaikan. Kadang-kadang aplikasi terjebak dalam perulangan dan berjalan tanpa henti, jadi Anda dapat mengatur batas waktu timeout 30 menit untuk mengakhiri tugas yang terjebak.

⚠ Important

Secara default, AWS Batch tidak memiliki batas waktu kerja. Jika Anda tidak menentukan batas waktu pekerjaan, pekerjaan akan berjalan hingga kontainer keluar.

Anda menentukan parameter `attemptDurationSeconds`, yang sedikitnya harus berjumlah 60 detik, dalam ketentuan tugas Anda atau ketika Anda mengirimkan tugas. Ketika jumlah detik ini telah berlalu setelah `startedAt` stempel waktu upaya pekerjaan, AWS Batch hentikan pekerjaan. Di sumber daya komputasi, kontainer tugas Anda menerima sinyal `SIGTERM` untuk memberi aplikasi Anda kesempatan untuk berhenti dengan baik. Jika kontainer masih berjalan setelah 30 detik, sinyal `SIGKILL` dikirim untuk menghentikan kontainer secara paksa.

Pengakhiran akibat timeout ditangani dengan dasar usaha terbaik. Anda sebaiknya tidak mengharapkan pengakhiran akibat timeout terjadi persis ketika waktu upaya tugas habis (mungkin diperlukan beberapa detik lebih lama). Jika aplikasi Anda memerlukan eksekusi timeout yang presisi, Anda harus menerapkan logika ini dalam aplikasi. Jika Anda memiliki sejumlah besar tugas yang mengalami timeout secara bersamaan, pengakhiran akibat timeout akan dilakukan dengan antrean `first in first out` (yang masuk terlebih dahulu akan keluar terlebih dahulu), dan tugas akan diakhiri dalam batch.

ℹ Note

Tidak ada nilai batas waktu maksimum untuk suatu AWS Batch pekerjaan.

Jika tugas diakhiri karena melebihi durasi timeout, itu tidak akan dicoba lagi. Jika upaya tugas gagal dengan sendirinya, upaya dapat dicoba lagi jika percobaan kembali diaktifkan, dan hitung mundur timeout dimulai kembali untuk upaya yang baru.

⚠ Important

Pekerjaan yang berjalan pada sumber daya Fargate tidak dapat diharapkan berjalan selama lebih dari 14 hari. Jika durasi timeout melebihi 14 hari, sumber daya Fargate mungkin tidak lagi tersedia dan tugas akan dihentikan.

Untuk tugas array, tugas turunan memiliki konfigurasi timeout yang sama dengan tugas induk.

Untuk informasi tentang mengirimkan AWS Batch pekerjaan dengan konfigurasi batas waktu, lihat.

[Mengirimkan pekerjaan](#)

Lowongan kerja Amazon EKS

Pekerjaan adalah unit kerja terkecil di AWS Batch. AWS Batch Pekerjaan di Amazon EKS memiliki one-to-one pemetaan ke Kubernetes pod. Definisi AWS Batch pekerjaan adalah templat untuk suatu AWS Batch pekerjaan. Saat Anda mengirimkan AWS Batch pekerjaan, Anda mereferensikan definisi pekerjaan, menargetkan antrian pekerjaan, dan memberikan nama untuk suatu pekerjaan. Dalam definisi pekerjaan AWS Batch pekerjaan di Amazon EKS, parameter [eksProperties](#) mendefinisikan kumpulan parameter yang didukung oleh pekerjaan di AWS Batch Amazon EKS. Dalam [SubmitJob](#) permintaan, [eksPropertiesOverride](#) parameter memungkinkan penggantian ke beberapa parameter umum. Dengan cara ini, Anda dapat menggunakan templat definisi pekerjaan untuk banyak pekerjaan. Saat pekerjaan dikirim ke kluster Amazon EKS Anda, AWS Batch ubah pekerjaan menjadi podspec (`Pod podspec`). Menggunakan beberapa AWS Batch parameter tambahan untuk memastikan bahwa pekerjaan diskalakan dan dijadwalkan dengan benar. AWS Batch menggabungkan label dan taints untuk memastikan pekerjaan hanya berjalan pada node AWS Batch terkelola dan bahwa pod lain tidak berjalan pada node tersebut.

Important

- Jika `hostNetwork` parameter tidak disetel secara eksplisit dalam definisi pekerjaan Amazon EKS, mode jaringan pod secara AWS Batch default ke mode host. Lebih khusus lagi, pengaturan berikut diterapkan: `hostNetwork=true` dan `dnsPolicy=ClusterFirstWithHostNet`.
- AWS Batch membersihkan pod pekerjaan segera setelah pod menyelesaikan tugasnya. Untuk melihat log aplikasi pod, konfigurasi layanan logging untuk kluster Anda. Untuk informasi selengkapnya, lihat [Gunakan CloudWatch Log untuk memantau pekerjaan AWS Batch Amazon EKS](#).

Memetakan pekerjaan yang sedang berjalan ke pod dan node

Pekerjaan `podProperties` yang sedang berjalan memiliki `podName` dan `nodeName` parameter yang ditetapkan untuk upaya pekerjaan saat ini. Gunakan operasi [DescribeJobs](#) API untuk melihat parameter ini.

Berikut ini adalah output contoh.

```
$ aws batch describe-jobs --job 2d044787-c663-4ce6-a6fe-f2baf7e51b04
{
  "jobs": [
    {
      "status": "RUNNING",
      "jobArn": "arn:aws:batch:us-east-1:123456789012:job/2d044787-c663-4ce6-a6fe-f2baf7e51b04",
      "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/MyJobOnEks_SleepWithRequestsOnly:1",
      "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/My-Eks-JQ1",
      "jobId": "2d044787-c663-4ce6-a6fe-f2baf7e51b04",
      "eksProperties": {
        "podProperties": {
          "nodeName": "ip-192-168-55-175.ec2.internal",
          "containers": [
            {
              "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
              "resources": {
                "requests": {
                  "cpu": "1",
                  "memory": "1024Mi"
                }
              }
            }
          ]
        },
        "podName": "aws-batch.b0aca953-ba8f-3791-83e2-ed13af39428c"
      }
    }
  ]
}
```

Untuk pekerjaan dengan percobaan ulang diaktifkan, podName dan nodeName dari setiap upaya yang diselesaikan ada dalam parameter eksAttempts daftar operasi [DescribeJobsAPI](#). podNameDan nodeName dari upaya yang sedang berjalan saat ini ada di podProperties objek.

Cara memetakan pod yang sedang berjalan kembali ke pekerjaannya

Sebuah pod memiliki label yang menunjukkan jobId dan uuid dari lingkungan komputasi yang dimilikinya. AWS Batch menyuntikkan variabel lingkungan sehingga runtime pekerjaan dapat

mereferensikan informasi pekerjaan. Untuk informasi selengkapnya, lihat [AWS Batch variabel lingkungan kerja](#). Anda dapat melihat informasi ini dengan menjalankan perintah berikut. Outputnya adalah sebagai berikut.

```
$ kubectl describe pod aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1 -n my-aws-batch-namespace
Name:          aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1
Namespace:     my-aws-batch-namespace
Priority:       0
Node:          ip-192-168-45-88.ec2.internal/192.168.45.88
Start Time:    Wed, 26 Oct 2022 00:30:48 +0000
Labels:        batch.amazonaws.com/compute-environment-uuid=5c19160b-d450-31c9-8454-86cf5b30548f
               batch.amazonaws.com/job-id=f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
               batch.amazonaws.com/node-uid=a4be5c1d-9881-4524-b967-587789094647
...
Status:        Running
IP:            192.168.45.88
IPs:
  IP: 192.168.45.88
Containers:
  default:
    Image:      public.ecr.aws/amazonlinux/amazonlinux:2
    ...
  Environment:
    AWS_BATCH_JOB_KUBERNETES_NODE_UID:  a4be5c1d-9881-4524-b967-587789094647
    AWS_BATCH_JOB_ID:                   f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
    AWS_BATCH_JQ_NAME:                  My-Eks-JQ1
    AWS_BATCH_JOB_ATTEMPT:              1
    AWS_BATCH_CE_NAME:                  My-Eks-CE1
...

```

Fitur yang didukung pekerjaan AWS Batch Amazon EKS

Ini adalah fitur AWS Batch spesifik yang juga umum untuk Kubernetes pekerjaan yang berjalan di Amazon EKS:

- [Dependensi Job](#)
- [Tugas array](#)
- [Batas waktu Job](#)

- [Percobaan kembali tugas secara otomatis](#)
- [Penjadwalan saham yang adil](#)

Kubernetes **Secrets** dan **ServiceAccounts**

AWS Batch mendukung referensi Kubernetes **Secrets** dan **ServiceAccounts**. Anda dapat mengonfigurasi pod untuk menggunakan peran Amazon EKS IAM untuk akun layanan. Untuk informasi selengkapnya, lihat [Mengonfigurasi pod untuk menggunakan akun Kubernetes layanan](#) di [Panduan Pengguna Amazon EKS](#).

Dokumen terkait

- [Pertimbangan memori dan vCPU untuk Amazon AWS Batch EKS](#)
- [Untuk membuat pekerjaan berbasis GPU di sumber daya Amazon EKS](#)
- [Pekerjaan terjebak dalam **RUNNABLE** status](#)

Tugas array

Tugas array adalah tugas yang berbagi parameter umum, seperti ketentuan tugas, vCPUs, dan memori. Ini berjalan sebagai kumpulan pekerjaan dasar terkait namun terpisah yang mungkin didistribusikan di beberapa host dan mungkin berjalan secara bersamaan. Tugas array adalah cara yang paling efisien untuk menjalankan tugas yang sangat paralel, seperti simulasi Monte Carlo, sapuan parametrik, atau tugas render yang besar.

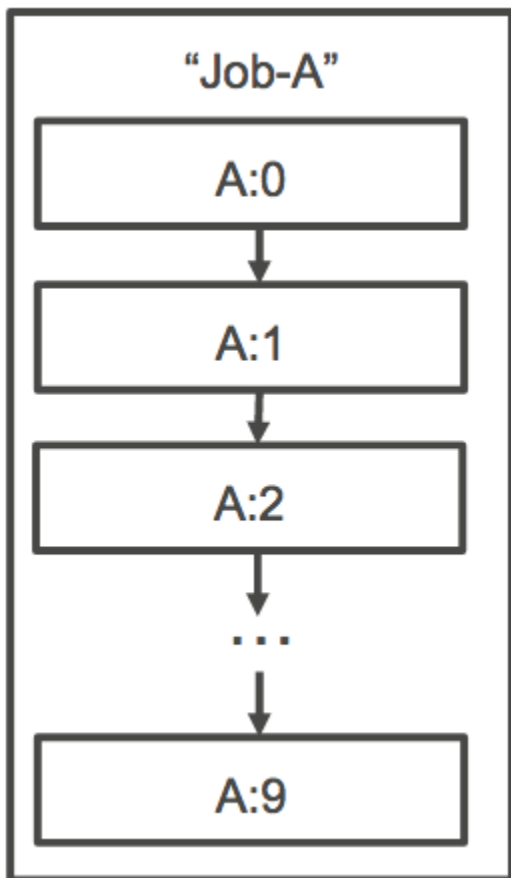
AWS Batch pekerjaan array dikirimkan seperti pekerjaan biasa. Namun, Anda menentukan ukuran array (antara 2 dan 10.000) untuk menentukan berapa banyak tugas turunan yang harus berjalan dalam array. Jika Anda mengirim tugas dengan ukuran array sebesar 1.000, satu tugas akan berjalan dan menghasilkan 1.000 tugas turunan. Tugas array adalah referensi atau penunjuk untuk mengelola semua tugas turunan. Dengan cara ini, Anda dapat mengirimkan beban kerja besar dengan satu kueri. Batas waktu yang ditentukan dalam `attemptDurationSeconds` parameter berlaku untuk setiap pekerjaan anak. Pekerjaan array induk tidak memiliki batas waktu.

Saat Anda mengirimkan pekerjaan array, pekerjaan array induk mendapatkan ID AWS Batch pekerjaan normal. Setiap pekerjaan anak memiliki ID dasar yang sama. Namun, indeks array untuk pekerjaan anak ditambahkan ke akhir ID induk, seperti `example_job_ID:0` untuk pekerjaan anak pertama dari array.

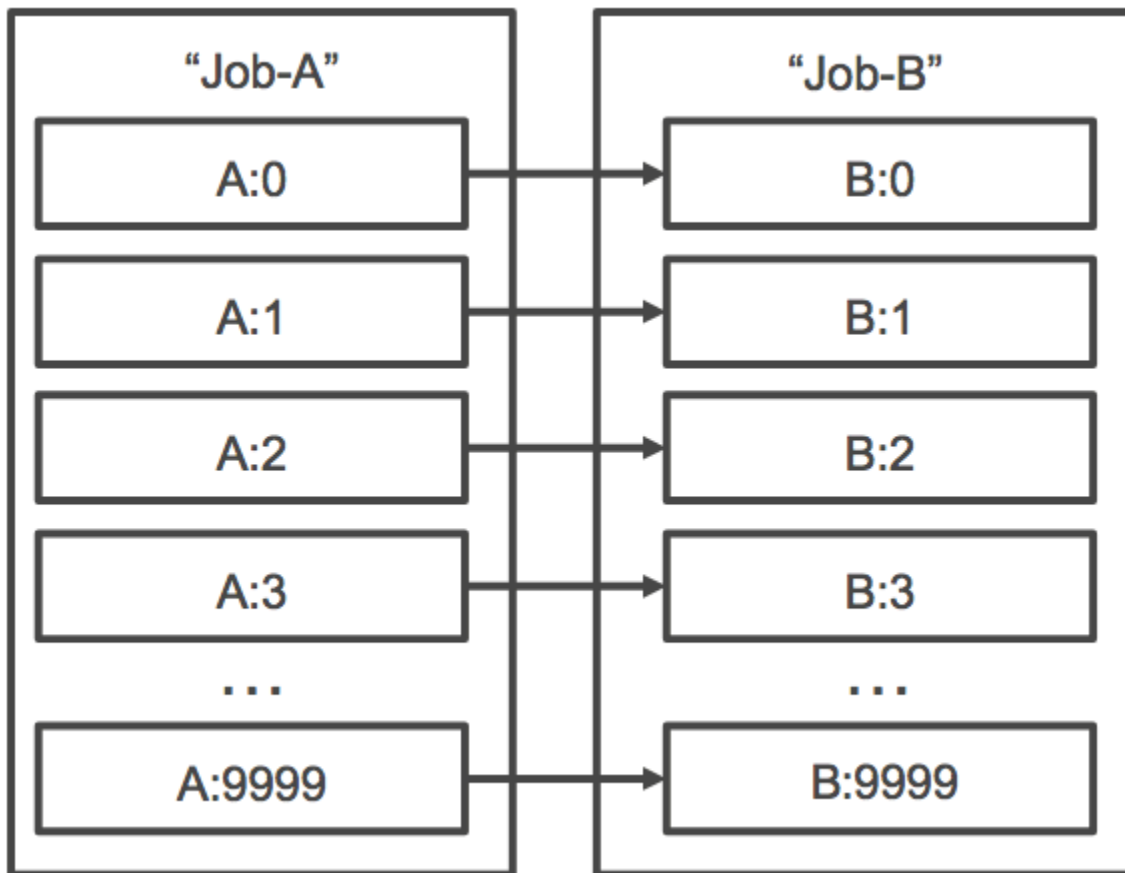
Pekerjaan array induk dapat memasukkan SUBMITTED,, PENDINGFAILED, atau SUCCEEDED status. Pekerjaan induk array diperbarui PENDING ketika setiap pekerjaan anak diperbarui keRUNNABLE. Untuk informasi selengkapnya tentang dependensi pekerjaan, lihat. [Dependensi Job](#)

Pada saat waktu aktif, variabel lingkungan AWS_BATCH_JOB_ARRAY_INDEX ditetapkan ke nomor indeks tugas array yang sesuai dengan kontainer. Indeks pekerjaan array pertama diberi nomor 0, dan upaya selanjutnya dalam urutan menaik (misalnya, 1, 2, dan 3). Anda dapat menggunakan nilai indeks ini untuk mengontrol cara turunan tugas array Anda dibedakan. Untuk informasi selengkapnya, lihat [Tutorial: Menggunakan indeks pekerjaan array untuk mengontrol diferensiasi pekerjaan](#).

Untuk dependensi tugas array, Anda dapat menentukan tipe untuk dependensi, misalnya SEQUENTIAL atau N_TO_N. Anda dapat menentukan tipe dependensi SEQUENTIAL (tanpa menentukan ID tugas) sehingga setiap tugas array turunan selesai secara berurutan, mulai dari indeks 0. Misalnya, jika Anda mengirimkan tugas array dengan ukuran array 100, dan menentukan dependensi dengan tipe SEQUENTIAL, 100 tugas turunan akan dihasilkan secara berurutan, dan tugas turunan pertama harus berhasil sebelum tugas turunan berikutnya dimulai. Gambar di bawah ini menunjukkan Tugas A, sebuah tugas array dengan ukuran array 10. Setiap tugas dalam indeks turunan Tugas A bergantung pada tugas turunan sebelumnya. Tugas A:1 tidak dapat mulai sebelum tugas A:0 selesai.



Anda juga dapat menentukan ketergantungan N_TO_N tipe dengan ID pekerjaan untuk pekerjaan array. Dengan cara itu, setiap turunan indeks dari tugas ini harus menunggu turunan indeks dari setiap dependensi selesai sebelum turunan tersebut dapat dimulai. Gambar berikut menunjukkan Job A dan Job B, dua pekerjaan array dengan ukuran array masing-masing 10.000. Setiap tugas dalam indeks turunan Tugas B bergantung pada indeks yang sesuai di Tugas A. Tugas B:1 tidak dapat mulai sebelum tugas A:1 selesai.



Jika Anda membatalkan atau mengakhiri tugas array induk, semua tugas turunan juga dibatalkan atau diakhiri. Anda dapat membatalkan atau mengakhiri setiap tugas turunan (yang memindahkannya ke status FAILED) tanpa memengaruhi tugas turunan lainnya. Namun, jika pekerjaan array anak gagal (sendiri atau dengan membatalkan atau mengakhiri secara manual), pekerjaan induk juga gagal.

Contoh alur kerja pekerjaan array

Alur kerja umum bagi AWS Batch pelanggan adalah menjalankan pekerjaan persiapan prasyarat, menjalankan serangkaian perintah terhadap sejumlah besar tugas input, dan kemudian menyimpulkan dengan pekerjaan yang mengumpulkan hasil dan menulis data ringkasan ke Amazon S3, DynamoDB, Amazon Redshift, atau Aurora.

Sebagai contoh:

- JobA: Tugas standar non-array yang membuat daftar cepat dan metadata validasi objek dalam bucket Amazon S3, BucketA. Sintaks [SubmitJob](#)JSON adalah sebagai berikut.

```
{
  "jobName": "JobA",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobA-list-and-validate:1"
}
```

- JobB: Pekerjaan array dengan 10.000 salinan yang bergantung pada JobA yang menjalankan perintah intensif CPU terhadap setiap objek di BucketA dan mengunggah hasilnya. BucketB Sintaks [SubmitJob](#)JSON adalah sebagai berikut.

```
{
  "jobName": "JobB",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobB-CPU-Intensive-Processing:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "4096"
      },
      {
        "type": "VCPU",
        "value": "32"
      }
    ]
  }
  "arrayProperties": {
    "size": 10000
  },
  "dependsOn": [
    {
      "jobId": "JobA_job_ID"
    }
  ]
}
```

- JobC: Pekerjaan 10.000 salinan array lainnya yang bergantung JobB pada model N_TO_N ketergantungan, yang menjalankan perintah intensif memori terhadap setiap item di, menulis metadata ke DynamoDBBucketB, dan mengunggah output yang dihasilkan ke. BucketC Sintaks [SubmitJob](#)JSON adalah sebagai berikut.

```
{
  "jobName": "JobC",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobC-Memory-Intensive-Processing:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "32768"
      },
      {
        "type": "VCPU",
        "value": "1"
      }
    ]
  }
  "arrayProperties": {
    "size": 10000
  },
  "dependsOn": [
    {
      "jobId": "JobB_job_ID",
      "type": "N_TO_N"
    }
  ]
}
```

- JobD: Pekerjaan array yang melakukan 10 langkah validasi yang masing-masing perlu kueri DynamoDB dan mungkin berinteraksi dengan salah satu bucket Amazon S3 di atas. Setiap langkah dalam JobD menjalankan perintah yang sama. Namun, perilakunya berbeda berdasarkan nilai variabel `AWS_BATCH_JOB_ARRAY_INDEX` lingkungan dalam wadah pekerjaan. Langkah-langkah validasi ini berjalan secara berurutan (misalnya, JobD:0 dan kemudian). JobD:1 Sintaks [SubmitJob](#) JSON adalah sebagai berikut.

```
{
  "jobName": "JobD",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobD-Sequential-Validation:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
```

```

        "type": "MEMORY",
        "value": "32768"
      },
      {
        "type": "VCPU",
        "value": "1"
      }
    ]
  }
  "arrayProperties": {
    "size": 10
  },
  "dependsOn": [
    {
      "jobId": "JobC_job_ID"
    },
    {
      "type": "SEQUENTIAL"
    }
  ],
]
}

```

- JobE: Tugas akhir non-array yang melakukan beberapa operasi pembersihan sederhana dan mengirimkan notifikasi Amazon SNS dengan pesan bahwa alur telah selesai dan tautan ke URL output. Sintaks [SubmitJobJSON](#) adalah sebagai berikut.

```

{
  "jobName": "JobE",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobE-Cleanup-and-Notification:1",
  "parameters": {
    "SourceBucket": "s3://JobD-Output-Bucket",
    "Recipient": "pipeline-notifications@mycompany.com"
  },
  "dependsOn": [
    {
      "jobId": "JobD_job_ID"
    }
  ]
}

```

Tutorial: Menggunakan indeks pekerjaan array untuk mengontrol diferensiasi pekerjaan

Tutorial ini menjelaskan cara menggunakan variabel `AWS_BATCH_JOB_ARRAY_INDEX` lingkungan untuk membedakan pekerjaan anak. Setiap pekerjaan anak ditugaskan ke variabel ini. Contoh menggunakan nomor indeks pekerjaan anak untuk membaca baris tertentu dalam file. Kemudian, ia mengganti parameter yang terkait dengan nomor baris itu dengan perintah di dalam wadah pekerjaan. Hasilnya adalah Anda dapat memiliki beberapa AWS Batch pekerjaan yang menjalankan gambar Docker dan argumen perintah yang sama. Namun, hasilnya berbeda karena indeks pekerjaan array digunakan sebagai pengubah.

Dalam tutorial ini, Anda membuat file teks yang memiliki semua warna pelangi, dengan setiap warna memiliki barisnya masing-masing. Kemudian, Anda membuat skrip titik masuk untuk wadah Docker yang mengubah indeks menjadi nilai yang dapat digunakan untuk nomor baris dalam file warna. Indeks dimulai dari nol, tetapi nomor baris dimulai dari satu. Buat Dockerfile yang menyalin file warna dan indeks ke gambar kontainer dan set `ENTRYPOINT` untuk gambar ke skrip `entrypoint`. Dockerfile dan sumber daya dibangun ke image Docker yang didorong ke Amazon ECR. Anda kemudian mendaftarkan definisi pekerjaan yang menggunakan gambar kontainer baru Anda, mengirimkan pekerjaan AWS Batch array dengan definisi pekerjaan itu, dan melihat hasilnya.

Prasyarat

Tutorial ini memiliki prasyarat berikut ini:

- Lingkungan AWS Batch komputasi. Untuk informasi selengkapnya, lihat [Membuat lingkungan komputasi](#).
- Antrian AWS Batch pekerjaan dan lingkungan komputasi terkait. Untuk informasi selengkapnya, lihat [Membuat antrean tugas](#).
- Yang AWS CLI diinstal pada sistem lokal Anda. Untuk informasi selengkapnya, lihat [Menginstal AWS Command Line Interface](#) dalam Panduan Pengguna AWS Command Line Interface .
- Docker yang diinstal di sistem lokal Anda. Untuk informasi selengkapnya, lihat [Tentang Docker CE](#) dalam dokumentasi Docker.

Langkah 1: Bangun gambar kontainer

Anda dapat menggunakan definisi `AWS_BATCH_JOB_ARRAY_INDEX` in a job di parameter perintah. Namun, kami menyarankan Anda membuat gambar kontainer yang menggunakan variabel dalam skrip entrypoint sebagai gantinya. Bagian ini menjelaskan cara membuat citra kontainer.

Untuk membangun gambar kontainer Docker Anda

1. Buat direktori baru untuk digunakan sebagai ruang kerja gambar Docker Anda dan buka direktori tersebut.
2. Buat file bernama `colors.txt` di direktori ruang kerja Anda dan tempelkan yang berikut ini ke dalamnya.

```
red
orange
yellow
green
blue
indigo
violet
```

3. Buat file bernama `print-color.sh` di direktori ruang kerja Anda dan tempelkan yang berikut ini ke dalamnya.

Note

Variabel `LINE` diatur ke `AWS_BATCH_JOB_ARRAY_INDEX + 1` karena indeks array dimulai dari 0, tetapi nomor baris dimulai dari 1. `COLOR` variabel diatur ke warna `colors.txt` yang terkait dengan nomor barisnya.

```
#!/bin/sh
LINE=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
COLOR=$(sed -n ${LINE}p /tmp/colors.txt)
echo My favorite color of the rainbow is $COLOR.
```

4. Buat file bernama `Dockerfile` di direktori ruang kerja Anda dan tempelkan konten berikut ke dalamnya. `Dockerfile` ini menyalin file sebelumnya ke kontainer Anda dan menetapkan skrip entrypoint yang akan dijalankan ketika kontainer dimulai.


```
FROM busybox
COPY print-color.sh /tmp/print-color.sh
COPY colors.txt /tmp/colors.txt
RUN chmod +x /tmp/print-color.sh
ENTRYPOINT /tmp/print-color.sh
```

5. Bangun gambar Docker Anda.

```
$ docker build -t print-color .
```

6. Uji kontainer Anda dengan skrip berikut. Skrip ini menetapkan `AWS_BATCH_JOB_ARRAY_INDEX` variabel ke 0 secara lokal dan kemudian meningkatkannya untuk mensimulasikan apa yang dilakukan oleh pekerjaan array dengan tujuh anak.

```
$ AWS_BATCH_JOB_ARRAY_INDEX=0
while [ $AWS_BATCH_JOB_ARRAY_INDEX -le 6 ]
do
    docker run -e AWS_BATCH_JOB_ARRAY_INDEX=$AWS_BATCH_JOB_ARRAY_INDEX print-color
    AWS_BATCH_JOB_ARRAY_INDEX=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
done
```

Berikut ini adalah outputnya.

```
My favorite color of the rainbow is red.
My favorite color of the rainbow is orange.
My favorite color of the rainbow is yellow.
My favorite color of the rainbow is green.
My favorite color of the rainbow is blue.
My favorite color of the rainbow is indigo.
My favorite color of the rainbow is violet.
```

Langkah 2: Dorong gambar Anda ke Amazon ECR

Sekarang setelah Anda membangun dan menguji wadah Docker Anda, dorong ke repositori gambar. Contoh ini menggunakan Amazon ECR, tetapi Anda dapat menggunakan registri lain, seperti DockerHub.

1. Buat repositori citra Amazon ECR untuk menyimpan citra kontainer Anda. Contoh ini hanya menggunakan AWS CLI, tetapi Anda juga dapat menggunakan AWS Management Console.

Untuk informasi selengkapnya, lihat [Membuat Repositori](#) dalam Panduan Pengguna Amazon Elastic Container Registry.

```
$ aws ecr create-repository --repository-name print-color
```

2. Beri tanda citra `print-color` dengan URI repositori Amazon ECR Anda yang dihasilkan dari langkah sebelumnya.

```
$ docker tag print-color aws_account_id.dkr.ecr.region.amazonaws.com/print-color
```

3. Masuk ke registri Amazon ECR Anda. Untuk informasi selengkapnya, lihat [Autentikasi Registri](#) dalam Panduan Pengguna Amazon Elastic Container Registry.

```
$ aws ecr get-login-password \
  --region region | docker login \
  --username AWS \
  --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

4. Dorong gambar Anda ke Amazon ECR.

```
$ docker push aws_account_id.dkr.ecr.region.amazonaws.com/print-color
```

Langkah 3: Buat dan daftarkan definisi pekerjaan

Sekarang gambar Docker Anda ada di registri gambar, Anda dapat menentukannya dalam definisi AWS Batch pekerjaan. Kemudian, Anda dapat menggunakannya nanti untuk menjalankan pekerjaan array. Contoh ini hanya menggunakan AWS CLI. Namun, Anda juga dapat menggunakan AWS Management Console. Untuk informasi selengkapnya, lihat [Membuat definisi pekerjaan tunggal](#).

Untuk membuat ketentuan tugas

1. Buat file bernama `print-color-job-def.json` di direktori ruang kerja Anda dan tempelkan yang berikut ini ke dalamnya. Ganti URI repositori citra dengan URI citra Anda sendiri.

```
{
  "jobDefinitionName": "print-color",
  "type": "container",
  "containerProperties": {
    "image": "aws_account_id.dkr.ecr.region.amazonaws.com/print-color",
```

```
"resourceRequirements": [  
  {  
    "type": "MEMORY",  
    "value": "250"  
  },  
  {  
    "type": "VCPU",  
    "value": "1"  
  }  
]  
}  
}
```

2. Daftarkan definisi pekerjaan dengan AWS Batch.

```
$ aws batch register-job-definition --cli-input-json file://print-color-job-def.json
```

Langkah 4: Kirim pekerjaan AWS Batch array

Setelah Anda mendaftarkan definisi pekerjaan Anda, Anda dapat mengirimkan pekerjaan AWS Batch array yang menggunakan gambar kontainer baru Anda.

Untuk mengirimkan pekerjaan AWS Batch array

1. Buat file bernama `print-color-job.json` di direktori ruang kerja Anda dan tempelkan yang berikut ini ke dalamnya.

Note

Contoh ini menggunakan antrian pekerjaan yang disebutkan di [the section called "Prasyarat"](#) bagian.

```
{  
  "jobName": "print-color",  
  "jobQueue": "existing-job-queue",  
  "arrayProperties": {  
    "size": 7  
  },  
}
```

```
"jobDefinition": "print-color"
}
```

2. Kirim pekerjaan ke antrian AWS Batch pekerjaan Anda. Perhatikan ID pekerjaan yang dikembalikan dalam output.

```
$ aws batch submit-job --cli-input-json file://print-color-job.json
```

3. Deskripsikan status tugas dan tunggu tugas untuk berpindah ke SUCCEEDED.

Langkah 5: Lihat log pekerjaan array Anda

Setelah pekerjaan Anda mencapai SUCCEEDED status, Anda dapat melihat CloudWatch Log dari wadah pekerjaan.

Untuk melihat log pekerjaan Anda di CloudWatch Log

1. Buka AWS Batch konsol di <https://console.aws.amazon.com/batch/>.
2. Di panel navigasi kiri, pilih Jobs (Tugas).
3. Untuk Job queue (Antrean tugas), pilih antrean.
4. Di bagian Status, pilih succeeded (berhasil).
5. Untuk menampilkan semua tugas turunan untuk tugas array Anda, pilih ID tugas yang dikembalikan di bagian sebelumnya.
6. Untuk melihat log dari kontainer tugas, pilih salah satu tugas turunan dan pilih View logs (Lihat log).

Filter events	
Time (UTC +00:00)	Message
2018-07-13	
	<i>No older events found at the moment. Retry.</i>
▶ 20:16:20	My favorite color of the rainbow is red.
	<i>No newer events found at the moment. Retry.</i>

7. Lihat log tugas turunan lainnya. Setiap tugas mengembalikan warna pelangi yang berbeda.

Tugas paralel multisimpul

Anda dapat menggunakan pekerjaan paralel multi-node untuk menjalankan pekerjaan tunggal yang mencakup beberapa instans Amazon EC2. Dengan AWS Batch multi-node parallel jobs, Anda dapat menjalankan aplikasi komputasi berkinerja tinggi skala besar dan pelatihan model GPU terdistribusi tanpa perlu meluncurkan, mengonfigurasi, dan mengelola sumber daya Amazon EC2 secara langsung. Pekerjaan paralel AWS Batch multi-node kompatibel dengan kerangka kerja apa pun yang mendukung komunikasi internode berbasis IP. Contohnya termasuk Apache MXNet,, Caffe2 TensorFlow, atau Message Passing Interface (MPI).

Tugas paralel multisimpul dikirimkan sebagai tugas tunggal. Namun, ketentuan tugas Anda (atau timpaan simpul pengiriman tugas) menentukan jumlah simpul yang akan dibuat untuk tugas dan grup simpul apa yang akan dibuat. Setiap tugas paralel multisimpul berisi simpul utama, yang diluncurkan terlebih dahulu. Setelah simpul utama berjalan, simpul turunan diluncurkan dan dimulai. Pekerjaan selesai hanya jika node utama keluar. Semua node anak kemudian dihentikan. Untuk informasi selengkapnya, lihat [Grup simpul](#).

Node pekerjaan paralel multi-node adalah penyewa tunggal. Ini berarti bahwa hanya satu wadah pekerjaan yang dijalankan di setiap instans Amazon EC2.

Status tugas akhir (SUCCEEDED atau FAILED) ditentukan oleh status tugas akhir dari simpul utama. Untuk mendapatkan status pekerjaan paralel multi-node, jelaskan pekerjaan dengan menggunakan ID pekerjaan yang dikembalikan saat Anda mengirimkan pekerjaan. Jika Anda memerlukan detail untuk node anak, jelaskan setiap node anak satu per satu. Anda dapat mengatasi node menggunakan `#N` notasi (dimulai dengan 0). Misalnya, untuk mengakses detail node kedua pekerjaan, jelaskan `aws_batch_job_id #1` menggunakan operasi API. AWS Batch [DescribeJobs](#) Informasi `started`, `stoppedAt`, `statusReason`, dan `exit` untuk tugas paralel multisimpul diisi dari simpul utama.

Jika Anda menentukan percobaan ulang pekerjaan, kegagalan node utama menyebabkan upaya lain terjadi. Kegagalan simpul anak tidak menyebabkan lebih banyak upaya terjadi. Setiap upaya baru dari tugas paralel multisimpul akan memperbarui upaya yang sesuai dari simpul turunan yang terkait.

Untuk menjalankan tugas paralel multi-node AWS Batch, kode aplikasi Anda harus berisi kerangka kerja dan pustaka yang diperlukan untuk komunikasi terdistribusi.

Variabel-variabel lingkungan

Saat runtime, setiap node dikonfigurasi variabel lingkungan standar yang diterima semua AWS Batch pekerjaan. Selain itu, node dikonfigurasi dengan variabel lingkungan berikut yang khusus untuk pekerjaan paralel multi-node:

`AWS_BATCH_JOB_MAIN_NODE_INDEX`

Variabel ini ditetapkan ke nomor indeks simpul utama tugas. Kode aplikasi Anda dapat `AWS_BATCH_JOB_MAIN_NODE_INDEX` membandingkan dengan `AWS_BATCH_JOB_NODE_INDEX` pada node individual untuk menentukan apakah itu node utama.

`AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS`

Variabel ini hanya diatur dalam node anak pekerjaan paralel multi-node. Variabel ini tidak ada pada node utama. Variabel ini ditetapkan ke alamat IPv4 privat simpul utama tugas. Kode aplikasi simpul turunan Anda dapat menggunakan alamat ini untuk berkomunikasi dengan simpul utama.

`AWS_BATCH_JOB_NODE_INDEX`

Variabel ini ditetapkan ke nomor indeks simpul pada simpul. Indeks simpul dimulai dari 0, dan setiap simpul menerima nomor indeks yang unik. Misalnya, tugas paralel multisimpul dengan 10 turunan memiliki nilai indeks 0-9.

`AWS_BATCH_JOB_NUM_NODES`

Variabel ini ditetapkan ke jumlah simpul yang telah Anda minta untuk tugas paralel multisimpul Anda.

Grup simpul

Grup node adalah grup node pekerjaan yang identik yang semuanya berbagi properti kontainer yang sama. Anda dapat menggunakan AWS Batch untuk menentukan hingga lima grup node yang berbeda untuk setiap pekerjaan.

Setiap grup dapat memiliki citra kontainer, perintah, variabel lingkungan, dan sebagainya sendiri. Misalnya, Anda dapat mengirimkan pekerjaan yang memerlukan satu `c5.xlarge` instance untuk node utama dan lima node `c5.xlarge` turunan instance. Masing-masing grup node yang berbeda ini dapat menentukan gambar atau perintah kontainer yang berbeda untuk dijalankan untuk setiap pekerjaan.

Atau, semua node dalam pekerjaan Anda dapat menggunakan satu grup node. Selain itu, kode aplikasi Anda dapat membedakan peran node seperti node utama dan node anak. Hal ini dilakukan dengan membandingkan variabel `AWS_BATCH_JOB_MAIN_NODE_INDEX` lingkungan terhadap nilainya sendiri untuk `AWS_BATCH_JOB_NODE_INDEX`. Anda dapat memiliki hingga 1.000 node dalam satu pekerjaan. Ini adalah batas default untuk instance di cluster Amazon ECS. Anda dapat [meminta untuk meningkatkan batas ini](#).

Note

Saat ini semua grup simpul dalam tugas paralel multisimpul harus menggunakan tipe instans yang sama.

Siklus hidup Job

Saat Anda mengirimkan pekerjaan paralel multi-node, pekerjaan tersebut memasuki SUBMITTED status. Kemudian, pekerjaan menunggu dependensi pekerjaan selesai. Pekerjaan juga pindah ke RUNNABLE status. Terakhir, AWS Batch berikan kapasitas instans yang diperlukan untuk menjalankan pekerjaan Anda dan meluncurkan instans ini.

Setiap tugas paralel multisimpul berisi simpul utama. Node utama adalah subtugas tunggal yang AWS Batch memantau untuk menentukan hasil dari pekerjaan multi node yang dikirimkan. Simpul utama diluncurkan pertama dan berpindah ke status STARTING. Nilai batas waktu yang ditentukan dalam `attemptDurationSeconds` parameter berlaku untuk seluruh pekerjaan dan bukan ke node.

Ketika node utama mencapai RUNNING status setelah wadah node berjalan, node anak diluncurkan dan mereka juga pindah ke STARTING status. Simpul turunan muncul dalam urutan acak. Tidak ada jaminan terkait waktu atau urutan peluncuran simpul turunan. Untuk memastikan bahwa semua node pekerjaan berada dalam RUNNING status setelah penampung node berjalan, kode aplikasi Anda dapat melakukan kueri AWS Batch API untuk mendapatkan informasi node utama dan node anak. Atau, kode aplikasi dapat menunggu sampai semua node online sebelum memulai tugas pemrosesan terdistribusi. Alamat IP privat dari simpul utama tersedia sebagai variabel lingkungan `AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS` di setiap simpul turunan. Kode aplikasi Anda dapat menggunakan informasi ini untuk mengoordinasikan dan mengomunikasikan data di antara setiap tugas.

Saat simpul individual keluar, simpul pindah ke SUCCEEDED atau FAILED, tergantung pada kode keluarnya. Jika simpul utama keluar, tugas dianggap selesai, dan semua simpul turunan dihentikan.

Jika node anak mati, AWS Batch tidak mengambil tindakan apa pun pada node lain dalam pekerjaan. Jika Anda tidak ingin pekerjaan Anda berlanjut dengan jumlah node yang berkurang, Anda harus memasukkan ini ke dalam kode aplikasi Anda. Melakukan hal ini mengakhiri atau membatalkan pekerjaan.

Hitung pertimbangan lingkungan

Ada beberapa hal yang perlu dipertimbangkan ketika mengonfigurasi lingkungan komputasi untuk menjalankan tugas paralel multisimpul dengan AWS Batch.

- Pekerjaan paralel multi-node tidak didukung pada lingkungan UNMANAGED komputasi.
- Jika Anda ingin mengirimkan pekerjaan paralel multi-node ke lingkungan komputasi, buat grup penempatan klaster dalam satu Availability Zone dan kaitkan dengan sumber daya komputasi Anda. Ini membuat pekerjaan paralel multi-node Anda pada pengelompokan instans yang logis dekat dengan potensi aliran jaringan yang tinggi. Untuk informasi lebih lanjut, lihat [Grup Penempatan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Pekerjaan paralel multi-node tidak didukung pada lingkungan komputasi yang menggunakan Instans Spot.
- AWS Batch multi-node parallel jobs menggunakan mode `awsvpc` jaringan Amazon ECS, yang memberikan container kerja paralel multi-node Anda properti jaringan yang sama dengan instans Amazon EC2. Setiap kontainer tugas paralel multisimpul mendapat antarmuka jaringan elastis sendiri, alamat IP privat utama, dan nama host DNS internal. Antarmuka jaringan dibuat di subnet VPC yang sama dengan sumber daya komputasi host-nya. Grup keamanan yang diterapkan pada sumber daya komputasi Anda juga diterapkan pada antarmuka jaringan. Untuk informasi selengkapnya, lihat, [Jaringan Tugas dengan Mode Jaringan `awsvpc`](#) dalam Panduan Developer Amazon Elastic Container.
- Lingkungan komputasi Anda mungkin memiliki tidak lebih dari lima grup keamanan yang terkait dengannya.
- Mode `awsvpc` jaringan tidak menyediakan antarmuka jaringan elastis untuk pekerjaan paralel multi-node dengan alamat IP publik. Untuk mengakses internet, sumber daya komputasi Anda harus diluncurkan di subnet privat yang dikonfigurasi untuk menggunakan gateway NAT. Untuk informasi lebih lanjut, lihat [Gateway NAT](#) dalam Panduan Pengguna Amazon VPC. Komunikasi antarsimpul harus menggunakan alamat IP privat atau nama host DNS untuk simpul tersebut. Pekerjaan paralel multi-node yang berjalan pada sumber daya komputasi dalam subnet publik tidak memiliki akses jaringan keluar. Untuk membuat VPC dengan subnet privat dan gateway NAT, lihat [Membuat cloud pribadi virtual](#) .

- Antarmuka jaringan elastis yang dibuat dan dilampirkan ke sumber daya komputasi Anda tidak dapat dipisahkan secara manual atau dimodifikasi oleh akun Anda. Ini untuk mencegah penghapusan secara tidak sengaja dari elastic network interface yang terkait dengan pekerjaan yang sedang berjalan. Untuk melepaskan antarmuka jaringan elastis untuk tugas, akhiri tugas.
- Lingkungan komputasi Anda harus memiliki vCPUs maksimum yang cukup untuk mendukung tugas paralel multisimpul Anda.
- Kuota instans Amazon EC2 Anda mencakup jumlah instans yang diperlukan untuk menjalankan pekerjaan Anda. Misalnya, misalkan pekerjaan Anda memerlukan 30 instans, tetapi akun Anda hanya dapat menjalankan 20 instans di Wilayah. Kemudian, pekerjaan Anda akan terjebak dalam RUNNABLE status.
- Jika Anda menentukan tipe instance untuk grup node dalam pekerjaan paralel multi-node, lingkungan komputasi Anda harus meluncurkan tipe instance tersebut.

Lowongan kerja GPU

Tugas GPU membantu Anda menjalankan tugas yang menggunakan instans GPU.

Berikut tipe instans Amazon EC2 berbasis GPU yang didukung. [Untuk informasi selengkapnya, lihat Instans Amazon EC2 G3, Instans Amazon EC2 G4, Instans Amazon EC2 G5, Instans Amazon EC2 P2, Instans Amazon EC2 P3, Instans Amazon EC2 P4d, dan Instans Amazon EC2 P5.](#)

Jenis instans	GPU	Memori GPU	vCPUs	Memori	Bandwith jaringan
g3s.xlarge	1	8 GiB	4	30,5 GiB	10 Gbps
g3.4xlarge	1	8 GiB	16	122 GiB	Hingga 10 Gbps
g3.8xlarge	2	16 GiB	32	244 GiB	10 Gbps
g3.16xlarge	4	32 GiB	64	488 GiB	25 Gbps
g4dn.xlarge	1	16 GiB	4	16 GiB	Hingga 25 Gbps
g4dn.2xlarge	1	16 GiB	8	32 GiB	Hingga 25 Gbps
g4dn.4xlarge	1	16 GiB	16	64 GiB	Hingga 25 Gbps
g4dn.8xlarge	1	16 GiB	32	128 GiB	50 Gbps

Jenis instans	GPU	Memori GPU	vCPUs	Memori	Bandwith jaringan
g4dn.12xlarge	4	64 GiB	48	192 GiB	50 Gbps
g4dn.16xlarge	1	16 GiB	64	256 GiB	50 Gbps
g5.xlarge	1	24 GiB	4	16 GiB	Hingga 10 Gbps
g5.2xlarge	1	24 GiB	8	32 GiB	Hingga 10 Gbps
g5.4xlarge	1	24 GiB	16	64 GiB	Hingga 25 Gbps
g5.8xlarge	1	24 GiB	32	128 GiB	25 Gbps
g5.16xlarge	1	24 GiB	64	256 GiB	25 Gbps
g5.12xlarge	4	96 GiB	48	192 GiB	40 Gbps
g5.24xlarge	4	96 GiB	96	384 GiB	50 Gbps
g5.48xlarge	8	192 GiB	192	768 GiB	100 Gbps
p2.xlarge	1	12 GiB	4	61 GiB	Tinggi
p2.8xlarge	8	96 GiB	32	488 GiB	10 Gbps
p2.16xlarge	16	192 GiB	64	732 GiB	20 Gbps
p3.2xlarge	1	16 GiB	8	61 GiB	Hingga 10 Gbps
p3.8xlarge	4	64 GiB	32	244 GiB	10 Gbps
p3.16xlarge	8	128 GiB	64	488 GiB	25 Gbps
p3dn.24xlarge	8	256 GiB	96	768 GiB	100 Gbps
p4d.24xlarge	8	320 GiB	96	1152 GiB	4x100 Gbps
p5.48xlarge	8	640 GiB	192	2 TiB	32x100 Gbps

Note

Hanya tipe instans yang mendukung GPU NVIDIA dan menggunakan arsitektur x86_64 yang didukung untuk pekerjaan GPU di AWS Batch. Misalnya, keluarga [G4ad](#) dan [G5g](#) instance tidak didukung.

Parameter [resourceRequirements](#) untuk ketentuan tugas menentukan jumlah GPU yang akan disematkan ke kontainer. Jumlah GPU ini tidak tersedia untuk pekerjaan lain yang berjalan pada instance itu selama durasi pekerjaan itu. Semua tipe instans dalam lingkungan komputasi yang menjalankan pekerjaan GPU harus berasal dari keluarga p2., p3, p4, p5, g3, g3sg4, atau g5 instance. Jika hal ini tidak dilakukan, tugas GPU mungkin akan terjebak di status RUNNABLE.

Tugas yang tidak menggunakan GPU dapat dijalankan di instans GPU. Namun, mungkin diperlukan biaya lebih untuk menjalankannya di instans GPU daripada di instans non-GPU yang serupa. Tergantung pada vCPU, memori, dan waktu tertentu yang diperlukan, tugas non-GPU ini mungkin memblokir berjalannya tugas GPU.

Untuk membuat pekerjaan berbasis GPU di sumber daya Amazon EKS

Bagian ini mencakup cara menjalankan beban kerja GPU Amazon EKS. AWS Batch

Daftar Isi

- [Untuk membuat Kubernetes cluster berbasis GPU di Amazon EKS](#)
- [Untuk membuat definisi pekerjaan GPU Amazon EKS](#)
- [Untuk menjalankan pekerjaan GPU di cluster Amazon EKS Anda](#)

Untuk membuat Kubernetes cluster berbasis GPU di Amazon EKS

Sebelum Anda membuat Kubernetes cluster berbasis GPU di Amazon EKS, Anda harus telah menyelesaikan langkah-langkahnya. [Memulai dengan AWS Batch di Amazon EKS](#) Selain itu, pertimbangkan juga hal berikut:

- AWS Batch mendukung jenis instance dengan GPU NVIDIA.

- Secara default, AWS Batch pilih AMI yang dipercepat Amazon EKS dengan Kubernetes versi yang cocok dengan versi bidang kontrol cluster Amazon EKS Anda.

```
$ cat <<EOF > ./batch-eks-gpu-ce.json
{
  "computeEnvironmentName": "My-Eks-GPU-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:<account>:cluster/<cluster-name>",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 1024,
    "instanceTypes": [
      "p3dn.24xlarge",
      "p4d.24xlarge"
    ],
    "subnets": [
      "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
    ],
    "securityGroupIds": [
      "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
EOF

$ aws batch create-compute-environment --cli-input-json file:///./batch-eks-gpu-ce.json
```

AWS Batch tidak mengelola plugin perangkat NVIDIA GPU atas nama Anda. Anda harus menginstal plugin ini ke cluster Amazon EKS Anda dan membiarkannya menargetkan AWS Batch node. Untuk informasi selengkapnya, lihat [Mengaktifkan Dukungan GPU di Kubernetes](#) aktif. GitHub

Untuk mengkonfigurasi plugin NVIDIA perangkat (DaemonSet) untuk menargetkan AWS Batch node, jalankan perintah berikut.

```
# pull nvidia daemonset spec
$ curl -O https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/v0.12.2/nvidia-device-plugin.yml
# using your favorite editor, add Batch node toleration
# this will allow the DaemonSet to run on Batch nodes
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"

$ kubectl apply -f nvidia-device-plugin.yml
```

Kami tidak menyarankan Anda mencampur beban kerja berbasis komputasi (CPU dan memori) dengan beban kerja berbasis GPU dalam pasangan lingkungan komputasi dan antrian pekerjaan yang sama. Ini karena pekerjaan komputasi dapat menggunakan kapasitas GPU.

Untuk melampirkan antrian pekerjaan, jalankan perintah berikut.

```
$ cat <<EOF > ./batch-eks-gpu-jq.json
{
  "jobQueueName": "My-Eks-GPU-JQ1",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "My-Eks-GPU-CE1"
    }
  ]
}
EOF

$ aws batch create-job-queue --cli-input-json file://./batch-eks-gpu-jq.json
```

Untuk membuat definisi pekerjaan GPU Amazon EKS

Hanya `nvidia.com/gpu` didukung saat ini dan nilai sumber daya yang Anda tetapkan harus berupa bilangan bulat. Anda tidak dapat menggunakan pecahan GPU. Untuk informasi selengkapnya, lihat [Menjadwalkan GPU](#) dalam Kubernetes dokumentasi.

Untuk mendaftarkan definisi pekerjaan GPU untuk Amazon EKS, jalankan perintah berikut.

```
$ cat <<EOF > ./batch-eks-gpu-jd.json
```

```

{
  "jobDefinitionName": "MyGPUJobOnEks_Smi",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "nvcr.io/nvidia/cuda:10.2-runtime-centos7",
          "command": ["nvidia-smi"],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi",
              "nvidia.com/gpu": "1"
            }
          }
        }
      ]
    }
  }
}
EOF

```

```
$ aws batch register-job-definition --cli-input-json file://./batch-eks-gpu-jd.json
```

Untuk menjalankan pekerjaan GPU di cluster Amazon EKS Anda

Sumber daya GPU tidak dapat dikompresi. AWS Batch membuat spesifikasi pod untuk pekerjaan GPU di mana nilai permintaan sama dengan nilai limit. Ini adalah Kubernetes persyaratan.

Untuk mengirimkan pekerjaan GPU, jalankan perintah berikut.

```

$ aws batch submit-job --job-queue My-Eks-GPU-JQ1 --job-definition MyGPUJobOnEks_Smi --
job-name My-Eks-GPU-Job

# locate information that can help debug or find logs (if using Amazon CloudWatch Logs
with Fluent Bit)
$ aws batch describe-jobs --job <job-id> | jq '.jobs[].eksProperties.podProperties |
{podName, nodeName}'
{
  "podName": "aws-batch.f3d697c4-3bb5-3955-aa6c-977fcf1cb0ca",
  "nodeName": "ip-192-168-59-101.ec2.internal"
}

```

```
}
```

Cari dan filter AWS Batch pekerjaan

Anda dapat membuat daftar pekerjaan dalam antrian pekerjaan menggunakan AWS Batch konsol. Namun, jika ada banyak pekerjaan dalam antrian pekerjaan, mungkin sulit untuk menemukan pekerjaan tertentu.

Anda dapat menggunakan Pencarian dan pemfilteran untuk mencantumkan pekerjaan yang cocok dengan kriteria pencarian yang Anda tentukan.

1. Buka [konsol AWS Batch](#).
2. Pilih Pekerjaan.
3. Aktifkan Pencarian dan pemfilteran.

Note

Jika Anda memiliki beberapa pekerjaan, proses ini mungkin memakan waktu beberapa menit.

4. Dalam kotak Silakan pilih antrian pekerjaan, pilih antrian pekerjaan yang ingin Anda cari.
5. Di kotak Filter sumber daya menurut properti atau nilai, pilih salah satu properti yang terdaftar.
6. Pilih operator yang ingin Anda gunakan. Misalnya, pilih Status =.

Tip

Untuk menggunakan properti atau operator yang berbeda, tutup kriteria saat ini. Kemudian, pilih properti dan operator yang Anda inginkan.

7. Masukkan atau pilih nilai properti. Misalnya, masukkan semua atau sebagian dari nama pekerjaan atau pilih Status = RUNNABLE.
8. Pilih pekerjaan yang Anda inginkan dalam daftar yang difilter.

Tip

Jika Anda tidak melihat pekerjaan yang Anda inginkan, gulir daftar yang difilter.

Log Job

Anda dapat mengonfigurasi AWS Batch pekerjaan Anda untuk mengirim informasi log ke CloudWatch Log. Dengan cara ini, Anda dapat melihat log yang berbeda dari pekerjaan Anda di satu lokasi yang nyaman. Untuk informasi selengkapnya, lihat [Menggunakan CloudWatch Log dengan AWS Batch](#).

Anda juga dapat menggunakan log Job di AWS Batch konsol untuk memantau atau memecahkan masalah pekerjaan. AWS Batch

1. Buka [konsol AWS Batch](#).
2. Pilih Pekerjaan.
3. Untuk antrean Job, pilih antrean pekerjaan yang Anda inginkan.

Tip

Jika ada beberapa pekerjaan dalam antrian pekerjaan, Anda dapat mengaktifkan Mencari dan memfilter untuk menemukan pekerjaan lebih cepat. Untuk informasi selengkapnya, lihat [Cari dan filter AWS Batch pekerjaan](#).

4. Untuk Status, pilih status pekerjaan yang Anda inginkan.
5. Pilih pekerjaan yang Anda inginkan.
6. Pada halaman Detail, gulir ke bawah ke Job Logs.
7. Pilih Ambil log.
8. Untuk Otorisasi yang diperlukan, masukkan **OK**, lalu pilih Otorisasi untuk menerima tagihan Amazon CloudWatch .

Note

Untuk mencabut otorisasi Anda untuk biaya: CloudWatch

1. Di panel navigasi kiri, pilih Izin.
2. Untuk log Job, pilih Edit.
3. Kosongkan kotak CloudWatch centang Otorisasi Batch untuk menggunakan.
4. Pilih Simpan perubahan.

9. Tinjau data log untuk AWS Batch pekerjaan itu.

i Tip

Anda dapat memfilter log berdasarkan Kata Kunci, Hasil Maks, dan Penyortiran. Anda juga dapat memilih salah satu interval waktu default atau membuat interval khusus untuk menyesuaikan hasil.

Informasi Job

Anda dapat meninjau informasi AWS Batch pekerjaan seperti status, definisi pekerjaan, dan informasi wadah.

1. Buka [konsol AWS Batch](#).
2. Pilih Pekerjaan.
3. Untuk antrean Job, pilih antrean pekerjaan yang Anda inginkan.

i Tip

Jika ada beberapa pekerjaan dalam antrian pekerjaan, Anda dapat mengaktifkan Mencari dan memfilter untuk menemukan pekerjaan lebih cepat. Untuk informasi selengkapnya, lihat [Cari dan filter AWS Batch pekerjaan](#).

4. Pilih pekerjaan yang Anda inginkan.

i Note

Anda juga dapat menggunakan AWS Command Line Interface (AWS CLI) untuk melihat detail tentang suatu AWS Batch pekerjaan. [Untuk informasi selengkapnya, lihat mendeskripsikan pekerjaan di Referensi Perintah.AWS CLI](#)

Ketentuan tugas

AWS Batch definisi pekerjaan menentukan bagaimana pekerjaan harus dijalankan. Meskipun setiap tugas harus mereferensikan ketentuan tugas, banyak parameter yang ditentukan dalam ketentuan tugas dapat diganti pada saat waktu aktif.

Daftar Isi

- [Membuat definisi pekerjaan simpul tunggal](#)
- [Membuat ketentuan tugas paralel multisimpul](#)
- [Membuat definisi pekerjaan menggunakan ContainerProperties](#)
- [Membuat definisi pekerjaan menggunakan EcsProperties](#)
- [Menggunakan driver log awslogs](#)
- [Menentukan data sensitif](#)
- [Otentikasi registri pribadi untuk pekerjaan](#)
- [Volume Amazon EFS](#)
- [Contoh ketentuan tugas](#)

Beberapa atribut yang ditentukan dalam ketentuan tugas meliputi:

- Gambar Docker yang akan digunakan dengan kontainer di tugas Anda
- Jumlah vCPUs dan memori yang akan digunakan dengan kontainer
- Perintah yang harus dijalankan kontainer ketika mulai
- Variabel lingkungan (jika ada) yang harus diteruskan ke kontainer ketika mulai
- Volume data yang harus digunakan dengan kontainer
- Apa (jika ada) peran IAM yang harus digunakan pekerjaan Anda untuk izin AWS

Untuk deskripsi lengkap parameter yang tersedia dalam ketentuan tugas, lihat [Parameter definisi Job untuk ContainerProperties](#).

Membuat definisi pekerjaan simpul tunggal

Sebelum dapat menjalankan tugas di AWS Batch, Anda harus membuat ketentuan tugas. Proses ini sedikit berbeda antara single-node dan multi-node parallel jobs. Topik ini membahas secara khusus

cara membuat definisi pekerjaan untuk AWS Batch pekerjaan yang bukan pekerjaan paralel multi-node.

Anda dapat membuat definisi pekerjaan paralel multi-node di sumber daya Amazon Elastic Container Service. Untuk informasi selengkapnya, lihat [the section called “Membuat ketentuan tugas paralel multisimpul”](#).

Topik

- [Membuat definisi pekerjaan simpul tunggal pada sumber daya Amazon EC2](#)
- [Membuat definisi pekerjaan simpul tunggal pada sumber daya AWS Fargate](#)
- [Membuat definisi pekerjaan simpul tunggal di sumber daya Amazon EKS](#)

Membuat definisi pekerjaan simpul tunggal pada sumber daya Amazon EC2

Untuk membuat definisi pekerjaan baru di sumber daya Amazon EC2:


1. Buka konsol AWS Batch di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi kiri, pilih Definisi Job.
4. Pilih Buat.
5. Untuk jenis Orkestrasi, pilih Amazon Elastic Compute Cloud (Amazon EC2).
6. Untuk konfigurasi platform EC2, matikan Aktifkan pemrosesan paralel multi-node.
7. Untuk Name (Name), masukkan nama yang unik untuk ketentuan tugas Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
8. (Opsional) Untuk batas waktu Eksekusi, masukkan nilai batas waktu (dalam detik). Batas waktu eksekusi adalah lamanya waktu sebelum pekerjaan yang belum selesai dihentikan. Jika upaya melebihi durasi batas waktu, upaya dihentikan dan dipindahkan ke FAILED status. Untuk informasi selengkapnya, lihat [Batas waktu Job](#). Nilai minimum adalah 60 detik.
9. (Opsional) Aktifkan Prioritas penjadwalan. Masukkan nilai prioritas penjadwalan antara 0 dan 100. Nilai yang lebih tinggi diberikan prioritas yang lebih tinggi.
10. (Opsional) Untuk upaya Job, masukkan berapa kali AWS Batch upaya untuk memindahkan pekerjaan ke RUNNABLE status. Masukkan angka antara 1 dan 10.

11. (Opsional) Untuk kondisi strategi Coba lagi, pilih Tambahkan evaluasi saat keluar. Masukkan setidaknya satu nilai parameter dan kemudian pilih Action. Untuk setiap set kondisi, Tindakan harus diatur ke Coba Ulang atau Keluar. Tindakan ini berarti sebagai berikut:
 - Coba lagi — AWS Batch coba lagi sampai jumlah upaya pekerjaan yang Anda tentukan tercapai.
 - Keluar — AWS Batch berhenti mencoba kembali pekerjaan.

 Important

Jika Anda memilih Tambahkan evaluasi saat keluar, Anda harus mengonfigurasi setidaknya satu parameter dan memilih Tindakan atau memilih Hapus evaluasi saat keluar.

12. (Opsional) Perluas Tag dan kemudian pilih Tambahkan tag untuk menambahkan tag ke sumber daya. Masukkan kunci dan nilai opsional, lalu pilih Tambah tag.
13. (Opsional) Aktifkan Propagate tag untuk menyebarkan tag dari definisi pekerjaan dan pekerjaan ke tugas Amazon ECS.
14. Pilih Halaman berikutnya.
15. Di bagian Konfigurasi kontainer:
 - a. Untuk Gambar, pilih Docker gambar yang akan digunakan untuk pekerjaan Anda. Secara default, gambar dalam Docker Hub registri tersedia. Anda juga dapat menentukan repositori lainnya dengan *repository-url/image:tag*. Namanya bisa mencapai 225 karakter panjangnya. Ini dapat berisi huruf besar dan kecil, angka, tanda hubung (-), garis bawah (_), titik dua (:), garis miring maju (/), dan tanda angka (#). Parameter ini sesuai dengan Image di bagian [Buat kontainer](#) di [Docker Remote API](#) dan IMAGE parameter [docker run](#).

 Note


Dockerarsitektur gambar harus sesuai dengan arsitektur prosesor dari sumber daya komputasi yang dijadwalkan. Misalnya, Docker gambar Arm berbasis hanya dapat berjalan pada sumber daya komputasi Arm berbasis.

- Gambar di Amazon ECR Repositori Publik menggunakan konvensi lengkap `registry/repository[:tag]` atau `registry/repository[@digest]` penamaan (misalnya, `public.ecr.aws/registry_alias/my-web-app:latest`)
 - Gambar di repositori Amazon ECR menggunakan konvensi `registry/repository[:tag]` penamaan lengkap (misalnya, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`)
 - Gambar di repositori resmi Docker Hub menggunakan satu nama (misalnya, `ubuntu` atau `mongo`).
 - Gambar di repositori lain di Docker Hub memenuhi syarat dengan nama organisasi (misalnya, `amazon/amazon-ecs-agent`).
 - Gambar di repositori online lainnya memenuhi syarat lebih lanjut berdasarkan nama domain (misalnya, `quay.io/assemblyline/ubuntu`).
- b. Untuk sintaks Command, pilih Bash atau JSON.
- c. Untuk Command (Perintah), tentukan perintah untuk diteruskan ke kontainer. Untuk perintah yang lebih sederhana, masukkan perintah seperti yang Anda lakukan untuk command prompt. Kemudian, verifikasi bahwa JSON hasilnya benar dan diteruskan ke Docker daemon. Untuk perintah yang lebih rumit (misalnya, dengan karakter khusus), gunakan sintaks JSON.

 Tip

Pilih Info untuk melihat Bash dan JSON kode sampel.

Parameter ini sesuai dengan Cmd di bagian [Buat kontainer](#) di [Docker Remote API](#) dan parameter COMMAND untuk [docker run](#). Untuk informasi lebih lanjut tentang Docker CMD parameter, lihat <https://docs.docker.com/engine/reference/builder/#cmd>.


 Note

Anda dapat menggunakan nilai dan placeholder default untuk substitusi parameter dalam perintah Anda. Untuk informasi selengkapnya, lihat [Parameter-parameter](#).

- d. (Opsional) Untuk peran Eksekusi, tentukan peran IAM yang memberikan izin kepada agen penampung Amazon ECS untuk melakukan panggilan AWS API atas nama Anda. Fitur


ini menggunakan peran Amazon ECS IAM untuk tugas. Untuk informasi selengkapnya, lihat [peran IAM eksekusi tugas Amazon ECS](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.

- e. Untuk konfigurasi Job Role, pilih peran IAM yang memiliki izin ke API. AWS Fitur ini menggunakan peran Amazon ECS IAM untuk tugas. Untuk informasi lebih lanjut, lihat [Peran IAM untuk Tugas](#) dalam Panduan Developer Amazon Elastic Container Service.

 Note

Hanya peran yang memiliki hubungan kepercayaan Amazon Elastic Container Service Task Role (Peran Tugas Amazon Elastic Container Service) yang ditampilkan di sini. Untuk informasi lebih lanjut tentang pembuatan IAM role untuk tugas AWS Batch, lihat [Membuat IAM Role dan Kebijakan untuk Tugas Anda](#) dalam Panduan Developer Amazon Elastic Container Service.

16. Untuk Parameter, pilih Tambahkan parameter untuk menambahkan placeholder substitusi parameter sebagai pasangan Kunci dan Nilai opsional.
17. Di bagian konfigurasi Lingkungan:
 - a. Untuk vCPU, masukkan jumlah vCPU yang akan dicadangkan untuk wadah. Parameter ini sesuai dengan CpuShares di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--cpu-shares` untuk [docker run](#). Setiap vCPU setara dengan 1.024 bagian CPU. Anda harus menentukan setidaknya satu vCPU.
 - b. Untuk Memori, masukkan batas memori yang tersedia untuk wadah. Jika penampung Anda mencoba melebihi jumlah memori yang Anda tentukan di sini, penampung dihentikan. Parameter ini sesuai dengan Memory di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--memory` untuk [docker run](#). Anda harus menentukan setidaknya 4 MiB memori untuk sebuah tugas.

 Note


Untuk memaksimalkan pemanfaatan sumber daya Anda, prioritaskan memori untuk pekerjaan dari jenis instans tertentu. Untuk informasi selengkapnya, lihat [Manajemen Memori Sumber Daya Komputasi](#).

- c. Untuk Jumlah GPU, pilih jumlah GPU yang akan dicadangkan untuk kontainer.

- d. (Opsional) Untuk variabel Lingkungan, pilih Tambahkan variabel lingkungan untuk menambahkan variabel lingkungan sebagai pasangan nama-nilai. Variabel-variabel ini diteruskan ke wadah.
 - e. (Opsional) Untuk Rahasia, pilih Tambahkan rahasia untuk menambahkan rahasia sebagai pasangan nama-nilai. Rahasia-rahasia ini terungkap dalam wadah. Untuk informasi lebih lanjut, lihat [secretOptions](#) di [Parameter definisi Job untuk ContainerProperties](#).
18. Pilih Halaman berikutnya.
19. Di bagian konfigurasi Linux:
- a. Untuk User (Pengguna), masukkan nama pengguna untuk digunakan di dalam kontainer. Parameter ini sesuai dengan User di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--user` untuk [docker run](#).
 - b. (Opsional) Untuk memberikan izin tinggi pada container pekerjaan pada instance host (mirip dengan `root` pengguna), seret slider Privileged ke kanan. Parameter ini memetakan ke Privileged di bagian [Membuat kontainer](#) dari [API Jarak Jauh Docker](#) dan pilihan `--privileged` untuk [docker run](#).
 - c. (Opsional) Aktifkan proses init Aktifkan untuk menjalankan `init` proses di dalam wadah. Proses ini meneruskan sinyal dan menuai proses.
20. (Opsional) Di bagian konfigurasi Sistem File:
- a. Aktifkan Aktifkan sistem file baca saja untuk menghapus akses tulis ke volume.
 - b. Untuk ukuran memori bersama, masukkan ukuran (dalam MiB) volume. `/dev/shm`
 - c. Untuk ukuran swap Max, masukkan jumlah total memori swap (dalam MiB) yang dapat digunakan wadah.
 - d. Untuk Swappiness masukkan nilai antara 0 dan 100 untuk menunjukkan perilaku swappiness wadah. Jika Anda tidak menentukan nilai dan pertukaran diaktifkan, nilai defaultnya menjadi 60. Untuk informasi lebih lanjut, lihat [swappiness](#) di [Parameter definisi Job untuk ContainerProperties](#).
 - e. (Opsional) Perluas Additional configuration (Konfigurasi tambahan).
 - f. (Opsional) Untuk Tmpfs, pilih Tambahkan tmpfs untuk menambahkan mount. `tmpfs`
 - g. (Opsional) Untuk Perangkat, pilih Tambah perangkat untuk menambahkan perangkat:
 - i. Untuk Container path (Jalur kontainer), tentukan jalur dalam instans kontainer untuk mengungkap perangkat yang dipetakan ke instans host. Jika Anda mengosongkan ini, jalur host digunakan dalam wadah.

- ii. Untuk Host path (Jalur host), tentukan jalur perangkat dalam instans host.
 - iii. Untuk Izin, pilih satu atau beberapa izin untuk diterapkan ke perangkat. Izin yang tersedia adalah BACA, TULIS, dan MKNOD.
 - h. (Opsional) Untuk konfigurasi Volume, pilih Tambahkan volume untuk membuat daftar volume yang akan diteruskan ke wadah. Masukkan Nama dan jalur Sumber untuk volume dan kemudian pilih Tambahkan volume. Anda juga dapat memilih untuk mengaktifkan Aktifkan EFS.
 - i. (Opsional) Untuk titik Mount, pilih konfigurasi Tambahkan titik pemasangan untuk menambahkan titik pemasangan untuk volume data. Anda harus menentukan volume sumber dan jalur kontainer. Titik pemasangan ini diteruskan ke instance Docker daemon on a container. Anda juga dapat memilih untuk membuat volume Baca saja.
 - j. (Opsional) Untuk konfigurasi Ulimits, pilih Tambahkan ulimit untuk menambahkan ulimits nilai untuk wadah. Masukkan nilai Name, Soft limit, dan Hard limit, lalu pilih Add ulimit.
21. (Opsional) Di bagian konfigurasi Logging:

- a. Untuk driver Log, pilih driver log yang akan digunakan. Untuk informasi selengkapnya tentang driver log yang tersedia, lihat [LogDriver](#) di [Parameter definisi Job untuk ContainerProperties](#).

 Note

Secara default, driver log `awslogs` akan digunakan.

- b. Untuk Opsi, pilih Tambahkan opsi untuk menambahkan opsi. Masukkan pasangan nama-nilai, lalu pilih Tambah opsi.
- c. Untuk Rahasia, pilih Tambahkan rahasia. Masukkan pasangan nama-nilai dan kemudian pilih Tambahkan rahasia untuk menambahkan rahasia.

 Tip

Untuk informasi lebih lanjut, lihat [secretOptions](#) di [Parameter definisi Job untuk ContainerProperties](#).

22. Pilih Halaman berikutnya.


23. Untuk tinjauan definisi Job, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat definisi pekerjaan.

Membuat definisi pekerjaan simpul tunggal pada sumber daya AWS Fargate

Untuk membuat definisi pekerjaan baru tentang AWS Fargate sumber daya:


1. Buka konsol AWS Batch di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi atas, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi kiri, pilih Definisi Job.
4. Pilih Buat.
5. Untuk tipe Orkestrasi, pilih Fargate. Untuk informasi selengkapnya, lihat [AWS Batch di AWS Fargate](#).
6. Untuk Name (Name), masukkan nama yang unik untuk ketentuan tugas Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
7. (Opsional) Untuk batas waktu Eksekusi, masukkan nilai batas waktu (dalam detik). Batas waktu eksekusi adalah lamanya waktu sebelum pekerjaan yang belum selesai dihentikan. Jika upaya melebihi durasi batas waktu, upaya dihentikan dan dipindahkan ke FAILED status. Untuk informasi selengkapnya, lihat [Batas waktu Job](#). Nilai minimum adalah 60 detik.
8. (Opsional) Aktifkan Prioritas penjadwalan. Masukkan nilai prioritas penjadwalan antara 0 dan 100. Nilai yang lebih tinggi diberikan prioritas yang lebih tinggi daripada nilai yang lebih rendah.
9. (Opsional) Perluas Tag, lalu pilih Tambahkan tag untuk menambahkan tag ke sumber daya. Aktifkan Menyebarkan tag untuk menyebarkan tag dari definisi pekerjaan dan pekerjaan.
10. Di bagian konfigurasi platform Fargate:
 - a. Untuk platform Runtime, pilih arsitektur lingkungan komputasi.
 - b. Untuk Keluarga Sistem Operasi, pilih sistem operasi untuk lingkungan komputasi.
 - c. Untuk Arsitektur CPU, pilih arsitektur vCPU.
 - d. Untuk versi platform Fargate, masukkan **LATEST** atau versi lingkungan runtime tertentu.
 - e. (Opsional) Aktifkan Tetapkan IP publik untuk menetapkan alamat IP publik ke antarmuka jaringan pekerjaan Fargate. Untuk tugas yang berjalan di subnet privat untuk mengirim lalu lintas keluar ke internet, subnet privat memerlukan gateway NAT yang dilampirkan untuk merutekan permintaan ke internet. Anda mungkin perlu melakukan hal ini agar dapat menarik gambar kontainer. Untuk informasi lebih lanjut, lihat [Jaringan tugas Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

- f. (Opsional) Untuk penyimpanan Ephemeral, masukkan jumlah penyimpanan fana untuk dialokasikan ke tugas. Jumlah penyimpanan fana harus antara 21 GiB dan 200 GiB. Secara default, 20 GiB penyimpanan sementara dialokasikan jika Anda tidak memasukkan nilai.

 Note

Penyimpanan sementara membutuhkan platform Fargate versi 1.4 atau yang lebih baru.

- g. Untuk peran Eksekusi, tentukan peran IAM yang memberikan izin kepada container Amazon ECS dan agen Fargate untuk melakukan panggilan AWS API atas nama Anda. Fitur ini menggunakan IAM role Amazon ECS untuk fungsionalitas tugas. Untuk informasi selengkapnya termasuk prasyarat konfigurasi, lihat [peran IAM eksekusi tugas Amazon ECS](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.
- h. Untuk upaya Job, masukkan berapa kali AWS Batch upaya untuk memindahkan pekerjaan ke RUNNABLE status. Masukkan angka antara 1 dan 10.
- i. Opsional) Untuk kondisi strategi Coba lagi, pilih Tambahkan evaluasi saat keluar. Masukkan setidaknya satu nilai parameter dan kemudian pilih Action. Untuk setiap set kondisi, Tindakan harus diatur ke Coba Ulang atau Keluar. Tindakan ini berarti sebagai berikut:
- Coba lagi — AWS Batch coba lagi sampai jumlah upaya pekerjaan yang Anda tentukan tercapai.
 - Keluar — AWS Batch berhenti mencoba kembali pekerjaan.

 Important


Jika Anda memilih Tambahkan evaluasi saat keluar, Anda harus mengonfigurasi setidaknya satu parameter dan memilih Tindakan atau memilih Hapus evaluasi saat keluar.

11. Pilih Halaman berikutnya.

12. Di bagian Konfigurasi kontainer:

- a. Untuk Image (Citra), pilih gambar Docker yang akan digunakan untuk tugas Anda. Secara default, gambar di registri Docker Hub akan tersedia. Anda juga dapat menentukan repositori lainnya dengan *repository-url/image:tag*. Namanya bisa mencapai 225

karakter panjangnya. Parameter dapat berisi huruf besar dan huruf kecil, angka, tanda hubung (-), garis bawah (_), titik dua (:), titik (.), garis miring (/), dan tanda pagar (#). Parameter ini sesuai dengan Image di bagian [Buat kontainer](#) di [Docker Remote API](#) dan IMAGE parameter [docker run](#).

 Note


Dockerarsitektur gambar harus sesuai dengan arsitektur prosesor dari sumber daya komputasi yang dijadwalkan. Misalnya, Docker gambar Arm berbasis hanya dapat berjalan pada sumber daya komputasi Arm berbasis.

- Gambar di Amazon ECR Repositori Publik menggunakan konvensi lengkap `registry/repository[:tag]` atau `registry/repository[@digest]` penamaan (misalnya, `public.ecr.aws/registry_alias/my-web-app:latest`)
 - Gambar di repositori Amazon ECR menggunakan konvensi `registry/repository[:tag]` penamaan lengkap (misalnya, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`)
 - Gambar di repositori resmi Docker Hub menggunakan satu nama (misalnya, `ubuntu` atau `mongo`).
 - Gambar di repositori lain di Docker Hub memenuhi syarat dengan nama organisasi (misalnya, `amazon/amazon-ecs-agent`).
 - Gambar di repositori online lainnya memenuhi syarat lebih lanjut berdasarkan nama domain (misalnya, `quay.io/assemblyline/ubuntu`).
- b. Untuk sintaks Command, pilih Bash atau JSON.
- c. Untuk Command (Perintah), tentukan perintah untuk diteruskan ke kontainer. Untuk perintah sederhana, masukkan perintah seperti yang Anda lakukan untuk prompt perintah dan kemudian verifikasi bahwa JSON hasilnya benar. Itu diteruskan ke Docker daemon. Untuk perintah yang lebih rumit (misalnya, dengan karakter khusus), gunakan sintaks JSON.

 Tip


Pilih Info untuk melihat Bash dan JSON kode sampel.

Parameter ini sesuai dengan Cmd di bagian [Buat kontainer](#) di [Docker Remote API](#) dan parameter COMMAND untuk [docker run](#). Untuk informasi lebih lanjut tentang Docker CMD parameter, lihat <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note


Anda dapat menggunakan nilai dan placeholder default untuk substitusi parameter dalam perintah Anda. Untuk informasi selengkapnya, lihat [Parameter-parameter](#).

- d. (Opsional) Tambahkan parameter ke definisi pekerjaan sebagai pemetaan nama-nilai untuk mengganti default definisi pekerjaan. Untuk menambahkan parameter:
- Untuk Parameter, pilih Tambahkan parameter, masukkan pasangan nama-nilai, lalu pilih Tambah parameter.

 Important

Jika Anda memilih Tambah parameter, Anda harus mengkonfigurasi setidaknya satu parameter atau memilih Hapus parameter


- e. Di bagian konfigurasi Lingkungan:
- i. Untuk konfigurasi peran Job, pilih peran IAM yang memiliki izin ke API. AWS Fitur ini menggunakan IAM role Amazon ECS untuk fungsionalitas tugas. Untuk informasi lebih lanjut, lihat [IAM Role untuk Tugas](#) dalam Panduan Developer Amazon Elastic Container Service.

 Note

Hanya peran yang memiliki hubungan kepercayaan Amazon Elastic Container Service Task Role (Peran Tugas Amazon Elastic Container Service) yang ditampilkan di sini. Untuk informasi selengkapnya tentang cara membuat peran IAM untuk AWS Batch pekerjaan Anda, lihat [Membuat Peran dan Kebijakan IAM untuk Tugas Anda](#) di Panduan Pengembang Layanan Amazon Elastic Container.

- ii. Untuk vCPU, masukkan jumlah vCPU yang akan dicadangkan untuk wadah. Parameter ini sesuai dengan CpuShares di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--cpu-shares` untuk [docker run](#). Setiap vCPU setara dengan 1.024 bagian CPU. Anda harus menentukan setidaknya satu vCPU.
- iii. Untuk Memori, masukkan batas memori yang tersedia untuk wadah. Jika kontainer Anda mencoba untuk melebihi memori yang ditentukan di sini, kontainer akan dihentikan. Parameter ini sesuai dengan Memory di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--memory` untuk [docker run](#). Anda harus menentukan sedikitnya 4 MiB memori untuk sebuah tugas.


Jika Anda menggunakan GuardDuty Runtime Monitoring, ada sedikit overhead memori untuk agen GuardDuty keamanan. Oleh karena itu batas memori harus mencakup ukuran agen GuardDuty keamanan. Untuk informasi tentang batas memori agen GuardDuty keamanan, lihat [batas CPU dan memori](#) di Panduan GuardDuty Pengguna. Untuk informasi tentang praktik terbaik, lihat [Bagaimana cara memperbaiki kesalahan memori pada tugas Fargate setelah mengaktifkan Runtime Monitoring](#) di Panduan Pengembang Amazon ECS.

 Note

Untuk memaksimalkan pemanfaatan sumber daya Anda, prioritaskan memori untuk pekerjaan dari jenis instans tertentu. Untuk informasi selengkapnya, lihat [Manajemen Memori Sumber Daya Komputasi](#).

- f. (Opsional) Untuk variabel Lingkungan, pilih Tambahkan variabel lingkungan untuk menambahkan variabel lingkungan sebagai pasangan nama-nilai. Variabel-variabel ini diteruskan ke wadah.
 - g. (Opsional) Untuk Rahasia, pilih Tambahkan rahasia untuk menambahkan rahasia sebagai pasangan nama-nilai. Rahasia-rahasia ini terungkap dalam wadah. Untuk informasi lebih lanjut, lihat [secretOptions](#) di [Parameter definisi Job untuk ContainerProperties](#).
 - h. Pilih Halaman berikutnya.
13. (Opsional) Di bagian konfigurasi Linux:
- a. Untuk Pengguna, masukkan nama pengguna untuk digunakan di dalam wadah.
 - b. Aktifkan proses Aktifkan init untuk menjalankan proses init di dalam wadah. Proses ini meneruskan sinyal dan menuai proses.

- c. Aktifkan Aktifkan sistem file baca saja untuk menghapus akses tulis ke volume.
- d. (Opsional) Perluas Additional configuration (Konfigurasi tambahan).
- e. Untuk konfigurasi titik Mount, pilih konfigurasi Tambahkan titik pemasangan untuk menambahkan titik pemasangan untuk volume data. Anda harus menentukan volume sumber dan jalur kontainer. Titik-titik pemasangan ini diteruskan ke instance Docker daemon on a container.
- f. Untuk konfigurasi Volume, pilih Tambahkan volume untuk membuat daftar volume yang akan diteruskan ke wadah. Masukkan jalur Nama dan Sumber untuk volume, lalu pilih Tambahkan volume.
- g. Di bagian konfigurasi Logging:
 - i. (Opsional) Untuk driver Log, pilih driver log yang akan digunakan. Untuk informasi selengkapnya tentang driver log yang tersedia, lihat [LogDriver](#) di [Parameter definisi Job untuk ContainerProperties](#).

 Note

Secara default, driver log `awslogs` akan digunakan.

- ii. (Opsional) Untuk Opsi, pilih Tambahkan opsi untuk menambahkan opsi. Masukkan pasangan nama-nilai, lalu pilih Tambah opsi.
- iii. (Opsional) Untuk Rahasia, pilih Tambahkan rahasia untuk menambahkan rahasia. Kemudian, masukkan pasangan nama-nilai, dan pilih Tambahkan rahasia.

 Tip

Untuk informasi lebih lanjut, lihat [secretOptions](#) di [Parameter definisi Job untuk ContainerProperties](#).

14. Pilih Halaman berikutnya.
15. Untuk tinjauan definisi Job, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat definisi pekerjaan.

Membuat definisi pekerjaan simpul tunggal di sumber daya Amazon EKS

Untuk membuat definisi pekerjaan baru di resource Amazon Elastic Kubernetes Service:

1. Buka konsol AWS Batch di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi atas, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi kiri, pilih Definisi Job.
4. Pilih Buat.
5. Untuk tipe Orkestrasi, pilih Elastic Kubernetes Service (EKS).
6. Untuk Name (Name), masukkan nama yang unik untuk ketentuan tugas Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
7. (Opsional) Untuk batas waktu Eksekusi, masukkan nilai batas waktu (dalam detik). Batas waktu eksekusi adalah lamanya waktu sebelum pekerjaan yang belum selesai dihentikan. Jika upaya melebihi durasi batas waktu, upaya dihentikan dan dipindahkan ke FAILED status. Untuk informasi selengkapnya, lihat [Batas waktu Job](#). Nilai minimum adalah 60 detik.
8. (Opsional) Aktifkan Prioritas penjadwalan. Masukkan nilai prioritas penjadwalan antara 0 dan 100. Nilai yang lebih tinggi diberikan prioritas yang lebih tinggi daripada nilai yang lebih rendah.
9. (Opsional) Perluas Tag, lalu pilih Tambahkan tag untuk menambahkan tag ke sumber daya.
10. Pilih Halaman berikutnya.
11. Di bagian podproperti EKS:
 - a. Untuk nama akun Layanan, masukkan akun yang memberikan identitas untuk proses yang berjalan di akunpod.
 - b. Hidupkan jaringan Host untuk menggunakan model Kubernetes pod jaringan dan buka port mendengarkan untuk koneksi masuk. Matikan pengaturan ini hanya untuk komunikasi keluar.
 - c. Untuk kebijakan DNS, pilih salah satu dari berikut ini:
 - Tidak ada nilai (null) — pod Mengabaikan pengaturan DNS dari lingkungan. Kubernetes
 - Default — pod Mewarisi konfigurasi resolusi nama dari node yang dijalankan.

Note

Jika kebijakan DNS tidak ditentukan, Default bukan kebijakan DNS default. Sebaliknya, ClusterFirst digunakan.

- ClusterFirst— Setiap kueri DNS yang tidak cocok dengan akhiran domain cluster yang dikonfigurasi diteruskan ke server nama hulu yang diwarisi dari node.
 - ClusterFirstWithHostNet— Gunakan jika jaringan Host dihidupkan.
- d. (Opsional) Untuk label Pod, pilih Tambahkan label pod, lalu masukkan pasangan nama-nilai.

Important


Awalan untuk label pod tidak dapat berisikubernetes.io/,k8s.io/, atau batch.amazonaws.com/.

- e. Pilih Halaman berikutnya.
- f. Di bagian Konfigurasi kontainer:
- i. Untuk Nama, masukkan nama unik untuk wadah. Nama harus dimulai dengan huruf atau angka, dan panjangnya bisa mencapai 63 karakter. Ini dapat berisi huruf besar dan kecil, angka, dan tanda hubung (-).
 - ii. Untuk Gambar, pilih Docker gambar yang akan digunakan untuk pekerjaan Anda. Secara default, gambar di registri Docker Hub akan tersedia. Anda juga dapat menentukan repositori lainnya dengan *repository-url/image:tag*. Nama dapat memuat hingga 255 karakter. Parameter dapat berisi huruf besar dan huruf kecil, angka, tanda hubung (-), garis bawah (_), titik dua (:), titik (.), garis miring (/), dan tanda pagar (#). Parameter ini dipetakan ke Image bagian [Create a container](#) dari [Docker Remote API](#) dan IMAGE parameter [docker run](#)

Note

Docker arsitektur gambar harus sesuai dengan arsitektur prosesor dari sumber daya komputasi yang dijadwalkan. Misalnya, Docker gambar Arm berbasis hanya dapat berjalan pada sumber daya komputasi Arm berbasis.


- Gambar di Amazon ECR Repositori Publik menggunakan konvensi lengkap `registry/repository[:tag]` atau `registry/repository[@digest]` penamaan (misalnya,). `public.ecr.aws/registry_alias/my-web-app:latest`
 - Gambar di repositori Amazon ECR menggunakan konvensi `registry/repository[:tag]` penamaan lengkap (misalnya,). `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`
 - Gambar di repositori resmi Docker Hub menggunakan satu nama (misalnya, ubuntu ataumongo).
 - Gambar di repositori lain di Docker Hub memenuhi syarat dengan nama organisasi (misalnya, amazon/amazon-ecs-agent).
 - Gambar di repositori online lainnya memenuhi syarat lebih lanjut berdasarkan nama domain (misalnya, quay.io/assemblyline/ubuntu).
- iii. (Opsional) Untuk kebijakan tarik gambar, pilih kapan gambar ditarik.
- iv. (Opsional) Untuk Perintah, masukkan JSON perintah Bash atau untuk diteruskan ke wadah.
- v. (Opsional) Untuk Argumen, masukkan argumen untuk diteruskan ke wadah. Jika argumen tidak disediakan, perintah image container digunakan.
- g. (Opsional) Anda dapat menambahkan parameter ke definisi pekerjaan sebagai pemetaan nama-nilai untuk mengganti default definisi pekerjaan. Untuk menambahkan parameter:
- Untuk Parameter, masukkan pasangan nama-nilai, lalu pilih Tambah parameter.

 Important

Jika Anda memilih Tambah parameter, Anda harus mengkonfigurasi setidaknya satu parameter atau memilih Hapus parameter


- h. Di bagian konfigurasi Lingkungan:
- i. Untuk vCPU, masukkan jumlah vCPU yang akan dicadangkan untuk wadah. Parameter ini sesuai dengan CpuShares di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--cpu-shares` untuk [docker run](#). Setiap vCPU setara dengan 1.024 bagian CPU. Anda harus menentukan setidaknya satu vCPU.

- ii. Untuk Memori, masukkan batas memori yang tersedia untuk wadah. Jika kontainer Anda mencoba untuk melebihi memori yang ditentukan di sini, kontainer akan dihentikan. Parameter ini sesuai dengan Memory di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--memory` untuk [docker run](#). Anda harus menentukan sedikitnya 4 MiB memori untuk sebuah tugas.

 Note


Untuk memaksimalkan pemanfaatan sumber daya Anda, prioritaskan memori untuk pekerjaan dari jenis instans tertentu. Untuk informasi selengkapnya, lihat [Manajemen Memori Sumber Daya Komputasi](#).

- i. (Opsional) Untuk variabel Lingkungan, pilih Tambahkan variabel lingkungan untuk menambahkan variabel lingkungan sebagai pasangan nama-nilai. Variabel-variabel ini diteruskan ke wadah.
- j. (Opsional) Untuk pemasangan Volume:
 - i. Pilih Tambahkan volume mount.
 - ii. Masukkan Nama, lalu masukkan jalur Mount di wadah tempat volume dipasang.
 - iii. Pilih Baca saja untuk menghapus izin menulis ke volume.
 - iv. Pilih Tambahkan volume mount.
- k. (Opsional) Untuk Jalankan sebagai pengguna, masukkan ID pengguna untuk menjalankan proses penampung.

 Note

ID pengguna harus ada dalam gambar agar wadah dapat dijalankan.

- l. (Opsional) Untuk Jalankan sebagai grup, masukkan ID grup untuk menjalankan runtime proses kontainer.

 Note

ID grup harus ada dalam gambar agar wadah dapat dijalankan.

- m. (Opsional) Untuk memberikan izin tinggi penampung pekerjaan Anda pada instance host (mirip dengan `root` pengguna), seret slider Privileged ke kanan. Parameter ini memetakan

ke Privileged di bagian [Membuat kontainer](#) dari [API Jarak Jauh Docker](#) dan pilihan `--privileged` untuk [docker run](#).

- n. (Opsional) Aktifkan sistem file root read-only untuk menghapus akses tulis ke sistem file root.
- o. (Opsional) Aktifkan `Run as non-root` untuk menjalankan container di pod sebagai non-root user.

Note

Jika `Run as non-root` diaktifkan, gambar kubelet divalidasi saat runtime untuk memverifikasi bahwa gambar tidak berjalan sebagai UID 0.

- p. Pilih Halaman berikutnya.

12. Untuk tinjauan definisi Job, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat definisi pekerjaan.

Membuat ketentuan tugas paralel multisimpul

Sebelum dapat menjalankan tugas di AWS Batch, Anda harus membuat ketentuan tugas. Proses ini sedikit berbeda antara single-node dan multi-node parallel jobs. Topik ini membahas secara khusus cara membuat definisi pekerjaan untuk pekerjaan paralel AWS Batch multi-node. Untuk informasi selengkapnya, lihat [Tugas paralel multisimpul](#).

Note

AWS Fargate tidak mendukung tugas paralel multisimpul.


Membuat definisi pekerjaan paralel multi-node pada sumber daya Amazon EC2

Untuk membuat ketentuan tugas simpul tunggal, lihat [Membuat definisi pekerjaan simpul tunggal](#).

Untuk membuat definisi pekerjaan paralel multi-node di sumber daya Amazon Elastic Compute Cloud:

1. Buka konsol AWS Batch di <https://console.aws.amazon.com/batch/>.


2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Definisi Job.
4. Pilih Buat.
5. Untuk jenis Orkestrasi, pilih Amazon Elastic Compute Cloud (Amazon EC2).
6. Untuk Aktifkan paralel multi-node, nyalakan paralel multi-node.
7. Untuk Name (Name), masukkan nama yang unik untuk ketentuan tugas Anda. Namanya bisa mencapai 128 karakter, dan dapat berisi huruf besar dan kecil, angka, tanda hubung (-), dan garis bawah (_).
8. (Opsional) Untuk batas waktu Eksekusi, tentukan jumlah maksimum detik yang Anda inginkan untuk menjalankan upaya pekerjaan. Jika upaya melebihi durasi batas waktu, upaya dihentikan dan dipindahkan ke FAILED status. Untuk informasi selengkapnya, lihat [Batas waktu Job](#).
9. (Opsional) Aktifkan Prioritas penjadwalan. Masukkan nilai prioritas penjadwalan antara 0 dan 100. Nilai yang lebih tinggi diberikan prioritas yang lebih tinggi daripada nilai yang lebih rendah.
10. (Opsional) Untuk upaya Job, masukkan berapa kali AWS Batch upaya untuk memindahkan pekerjaan ke RUNNABLE status. Masukkan angka antara 1 dan 10.
11. (Opsional) Untuk kondisi strategi Coba lagi, pilih Tambahkan evaluasi saat keluar. Masukkan setidaknya satu nilai parameter dan kemudian pilih Action. Untuk setiap set kondisi, Tindakan harus diatur ke Coba Ulang atau Keluar. Tindakan ini berarti sebagai berikut:
 - Coba lagi — AWS Batch coba lagi sampai jumlah upaya pekerjaan yang Anda tentukan tercapai.
 - Keluar — AWS Batch berhenti mencoba kembali pekerjaan.

 Important

Jika Anda memilih Tambahkan evaluasi saat keluar, Anda harus mengonfigurasi setidaknya satu parameter dan memilih Tindakan atau memilih Hapus evaluasi saat keluar.

12. (Opsional) Perluas Tag dan kemudian pilih Tambahkan tag untuk menambahkan tag ke sumber daya. Masukkan kunci dan nilai opsional, lalu pilih Tambah tag. Anda juga dapat mengaktifkan Menyebarkan tag untuk menyebarkan tag dari definisi pekerjaan dan pekerjaan ke tugas Amazon ECS.
13. Pilih Halaman berikutnya.

14. Untuk Number of nodes (Jumlah simpul), masukkan jumlah total simpul yang akan digunakan untuk tugas Anda.
15. Untuk Main node (Simpul utama), masukkan indeks simpul yang akan digunakan untuk simpul utama. Indeks simpul utama default adalah 0.
16. Untuk tipe Instance, pilih jenis instance.


 Note

Jenis instance yang Anda pilih berlaku untuk semua node.

17. Untuk Parameter, pilih Tambahkan parameter untuk menambahkan placeholder substitusi parameter sebagai pasangan Kunci dan Nilai opsional.
18. Di bagian rentang Node:
 - a. Pilih Add node range (Tambahkan rentang simpul). Ini akan membuat bagian Node range (Rentang simpul).
 - b. Untuk Target nodes (Simpul target), tentukan rentang untuk grup simpul Anda, menggunakan notasi *range_start:range_end*.

Anda dapat membuat hingga lima rentang node untuk node yang Anda tentukan untuk pekerjaan Anda. Rentang simpul menggunakan nilai indeks untuk simpul, dan indeks simpul dimulai dari 0. Pastikan bahwa nilai indeks akhir rentang grup node akhir Anda adalah satu kurang dari jumlah node yang Anda tentukan. Misalnya, anggaplah Anda menentukan 10 node, dan Anda ingin menggunakan grup node tunggal. Kemudian, rentang akhir Anda adalah 9.


- c. Untuk Gambar, pilih Docker gambar yang akan digunakan untuk pekerjaan Anda. Secara default, gambar dalam Docker Hub registri tersedia. Anda juga dapat menentukan repositori lainnya dengan *repository-url/image:tag*. Namanya bisa mencapai 225 karakter. Ini dapat berisi huruf besar dan kecil, angka, tanda hubung (-), garis bawah (_), titik dua (:), garis miring maju (/), dan tanda angka (#). Parameter ini sesuai dengan Image di bagian [Buat kontainer](#) di [Docker Remote API](#) dan IMAGE parameter [docker run](#).

 Note

Dockerarsitektur gambar harus sesuai dengan arsitektur prosesor dari sumber daya komputasi yang dijadwalkan. Misalnya, Docker gambar Arm berbasis hanya dapat berjalan pada sumber daya komputasi Arm berbasis.


- Gambar di Amazon ECR Repositori Publik menggunakan konvensi lengkap `registry/repository[:tag]` atau `registry/repository[@digest]` penamaan (misalnya), `public.ecr.aws/registry_alias/my-web-app:latest`
 - Gambar di repositori Amazon ECR menggunakan konvensi penamaan `registry/repository[:tag]` lengkap. Misalnya, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`
 - Gambar di repositori resmi Docker Hub menggunakan satu nama (misalnya, ubuntu ataumongo).
 - Gambar di repositori lain di Docker Hub memenuhi syarat dengan nama organisasi (misalnya, amazon/amazon-ecs-agent).
 - Gambar di repositori online lainnya memenuhi syarat lebih lanjut berdasarkan nama domain (misalnya, quay.io/assemblyline/ubuntu).
- d. Untuk sintaks Command, pilih Bash atau JSON.
- e. Untuk Command (Perintah), tentukan perintah untuk diteruskan ke kontainer. Untuk perintah sederhana, Anda dapat memasukkan perintah seperti yang Anda lakukan pada prompt perintah di tab Space delimited. Kemudian, verifikasi bahwa JSON hasilnya benar. Hasil JSON diteruskan ke Docker daemon Untuk perintah yang lebih rumit (misalnya, dengan karakter khusus), Anda dapat beralih ke tab JSON dan masukkan array string yang setara di sana.

Parameter ini sesuai dengan Cmd di bagian [Buat kontainer](#) di [Docker Remote API](#) dan parameter COMMAND untuk [docker run](#). Untuk informasi lebih lanjut tentang Docker CMD parameter, lihat <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note


Anda dapat menggunakan nilai dan placeholder default untuk substitusi parameter dalam perintah Anda. Untuk informasi selengkapnya, lihat [Parameter-parameter](#).

- f. Untuk vCPUs, tentukan jumlah vCPUs yang akan disimpan untuk kontainer. Parameter ini sesuai dengan CpuShares di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--cpu-shares` untuk [docker run](#). Setiap vCPU setara dengan 1.024 bagian CPU. Anda harus menentukan setidaknya satu vCPU.
- g. Untuk Memory (Memori), tentukan batas keras (dalam MiB) memori yang akan diberikan ke kontainer tugas. Jika kontainer Anda mencoba untuk melebihi memori yang ditentukan di sini, kontainer akan dihentikan. Parameter ini sesuai dengan Memory di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--memory` untuk [docker run](#). Anda harus menentukan sedikitnya 4 MiB memori untuk sebuah tugas.


 Note

Untuk memaksimalkan pemanfaatan sumber daya Anda, Anda dapat memberikan pekerjaan Anda sebanyak mungkin memori untuk jenis instance tertentu. Untuk informasi selengkapnya, lihat [Manajemen Memori Sumber Daya Komputasi](#).

- h. (Opsional) Untuk Number of GPUs (Jumlah GPU), tentukan jumlah GPU yang digunakan tugas Anda. Pekerjaan berjalan pada wadah dengan jumlah GPU tertentu yang disematkan ke wadah itu.
- i. (Opsional) Untuk Job role (Peran tugas), Anda dapat menentukan IAM role yang menyediakan kontainer di tugas Anda dengan izin untuk menggunakan API AWS. Fitur ini menggunakan IAM role Amazon ECS untuk fungsionalitas tugas. Untuk informasi selengkapnya termasuk prasyarat konfigurasi, lihat [Peran IAM untuk Tugas di](#) Panduan Pengembang Layanan Kontainer Elastis Amazon.

 Note

Untuk pekerjaan yang berjalan pada sumber daya Fargate, diperlukan peran pekerjaan.

 Note


Hanya peran yang memiliki hubungan kepercayaan Amazon Elastic Container Service Task Role (Peran Tugas Amazon Elastic Container Service) yang ditampilkan di sini. Untuk informasi lebih lanjut tentang pembuatan IAM role untuk

tugas AWS Batch, lihat [Membuat IAM Role dan Kebijakan untuk Tugas Anda](#) dalam Panduan Developer Amazon Elastic Container Service.


- j. (Opsional) Untuk peran Eksekusi, tentukan peran IAM yang memberikan izin kepada agen penampung Amazon ECS untuk melakukan panggilan AWS API atas nama Anda. Fitur ini menggunakan IAM role Amazon ECS untuk fungsionalitas tugas. Untuk informasi selengkapnya, lihat [peran IAM eksekusi tugas Amazon ECS](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.

19. (Opsional) Perluas konfigurasi tambahan:

- a. Untuk variabel Lingkungan, pilih Tambahkan variabel lingkungan untuk menambahkan variabel lingkungan sebagai pasangan nama-nilai. Variabel-variabel ini diteruskan ke wadah.
- b. Untuk konfigurasi peran Job, Anda dapat menentukan peran IAM yang menyediakan kontainer di pekerjaan Anda dengan izin untuk menggunakan API. AWS Fitur ini menggunakan IAM role Amazon ECS untuk fungsionalitas tugas. Untuk informasi selengkapnya termasuk prasyarat konfigurasi, lihat [Peran IAM untuk Tugas di](#) Panduan Pengembang Layanan Kontainer Elastis Amazon.

 Note

Untuk pekerjaan yang berjalan pada sumber daya Fargate, diperlukan peran pekerjaan.

 Note

Hanya peran yang memiliki hubungan kepercayaan Amazon Elastic Container Service Task Role (Peran Tugas Amazon Elastic Container Service) yang ditampilkan di sini. Untuk informasi selengkapnya tentang cara membuat peran IAM untuk AWS Batch pekerjaan Anda, lihat [Membuat Peran dan Kebijakan IAM untuk Tugas Anda](#) di Panduan Pengembang Layanan Amazon Elastic Container.

- c. Untuk peran Eksekusi, tentukan peran IAM yang memberikan izin kepada agen penampung Amazon ECS untuk melakukan panggilan AWS API atas nama Anda. Fitur ini menggunakan IAM role Amazon ECS untuk fungsionalitas tugas. Untuk informasi selengkapnya, lihat [peran](#)

[IAM eksekusi tugas Amazon ECS](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.

20. Di bagian Konfigurasi Keamanan:


- a. (Opsional) Untuk memberikan hak istimewa yang lebih tinggi pada container pekerjaan Anda pada instance host (mirip dengan `root` pengguna), aktifkan `Privileged`. Parameter ini memetakan ke `Privileged` di bagian [Membuat kontainer](#) dari [API Jarak Jauh Docker](#) dan pilihan `--privileged` untuk [docker run](#).
- b. (Opsional) Untuk Pengguna, masukkan nama pengguna untuk digunakan di dalam wadah. Parameter ini memetakan ke `User` di bagian [Membuat kontainer](#) dari [API Jarak Jauh Docker](#) dan pilihan `--user` untuk [docker run](#).
- c. (Opsional) Untuk Rahasia, pilih `Tambahkan` rahasia untuk menambahkan rahasia sebagai pasangan nama-nilai. Rahasia-rahasia ini terungkap dalam wadah. Untuk informasi lebih lanjut, lihat [secretOptions](#) di [Parameter definisi Job untuk ContainerProperties](#).

21. Di bagian konfigurasi Linux:

- a. Aktifkan Aktifkan sistem file baca saja untuk menghapus akses tulis ke volume.
- b. (Opsional) Aktifkan `init` proses Aktifkan untuk menjalankan `init` proses di dalam wadah. Proses ini meneruskan sinyal dan menuai proses.
- c. Untuk ukuran memori bersama, masukkan ukuran (dalam MiB) volume. `/dev/shm`
- d. Untuk ukuran swap Max, masukkan jumlah total memori swap (dalam MiB) yang dapat digunakan wadah.
- e. Untuk Swappiness masukkan nilai antara 0 dan 100 untuk menunjukkan perilaku swappiness wadah. Jika Anda tidak menentukan nilai dan pertukaran diaktifkan, nilai default ke 60. Untuk informasi lebih lanjut, lihat [swappiness](#) di [Parameter definisi Job untuk ContainerProperties](#).
- f. (Opsional) Untuk Perangkat, pilih `Tambah` perangkat untuk menambahkan perangkat:
 - i. Untuk Container path (Jalur kontainer), tentukan jalur dalam instans kontainer untuk mengungkap perangkat yang dipetakan ke instans host. Jika Anda mengosongkan ini, jalur host digunakan dalam wadah.
 - ii. Untuk Host path (Jalur host), tentukan jalur perangkat dalam instans host.
 - iii. Untuk Izin, pilih satu atau beberapa izin untuk diterapkan ke perangkat. Izin yang tersedia adalah `BACA`, `TULIS`, dan `MKNOD`.

22. (Opsional) Untuk titik Mount, pilih konfigurasi Tambahkan titik pemasangan untuk menambahkan titik pemasangan untuk volume data. Anda harus menentukan volume sumber dan jalur kontainer. Titik pemasangan ini diteruskan ke Docker daemon pada instance kontainer. Anda juga dapat memilih untuk membuat volume Baca saja.
23. (Opsional) Untuk konfigurasi Ulimits, pilih Tambahkan ulimit untuk menambahkan ulimits nilai untuk wadah. Masukkan nilai Name, Soft limit, dan Hard limit, lalu pilih Add ulimit.
24. (Opsional) Untuk konfigurasi Volume, pilih Tambahkan volume untuk membuat daftar volume yang akan diteruskan ke wadah. Masukkan Nama dan jalur Sumber untuk volume dan kemudian pilih Tambahkan volume. Anda juga dapat memilih untuk mengaktifkan Aktifkan EFS.
25. (Opsional) Untuk Tmpfs, pilih Tambahkan tmpfs untuk menambahkan mount. tmpfs
26. (Opsional) Di bagian konfigurasi Logging:

- a. Untuk driver Log, pilih driver log yang akan digunakan. Untuk informasi selengkapnya tentang driver log yang tersedia, lihat [LogDriver](#) di [Parameter definisi Job untuk ContainerProperties](#).

 Note

Secara default, driver log `awslogs` akan digunakan.

- b. Untuk Opsi, pilih Tambahkan opsi untuk menambahkan opsi. Masukkan pasangan nama-nilai, lalu pilih Tambah opsi.
- c. Untuk Rahasia, pilih Tambahkan rahasia. Masukkan pasangan nama-nilai dan kemudian pilih Tambahkan rahasia untuk menambahkan rahasia.

 Tip

Untuk informasi lebih lanjut, lihat [secretOptions](#) di [Parameter definisi Job untuk ContainerProperties](#).

27. Pilih Halaman berikutnya.
28. Untuk tinjauan definisi Job, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat definisi pekerjaan.

Membuat definisi pekerjaan menggunakan ContainerProperties

Berikut ini adalah template definisi pekerjaan kosong yang mencakup satu wadah. Anda dapat menggunakan template ini untuk membuat definisi pekerjaan Anda, yang kemudian dapat disimpan ke file dan digunakan dengan AWS CLI `--cli-input-json` opsi. Untuk informasi selengkapnya tentang parameter ini, lihat [Parameter definisi Job untuk ContainerProperties](#).

```
{
  "jobDefinitionName": "",
  "type": "container",
  "parameters": {
    "KeyName": ""
  },
  "schedulingPriority": 0,
  "containerProperties": {
    "image": "",
    "vcpus": 0,
    "memory": 0,
    "command": [
      ""
    ],
    "jobRoleArn": "",
    "executionRoleArn": "",
    "volumes": [
      {
        "host": {
          "sourcePath": ""
        },
        "name": "",
        "efsVolumeConfiguration": {
          "fileSystemId": "",
          "rootDirectory": "",
          "transitEncryption": "ENABLED",
          "transitEncryptionPort": 0,
          "authorizationConfig": {
            "accessPointId": "",
            "iam": "DISABLED"
          }
        }
      }
    ],
    "environment": [
      {
```

```
        "name": "",
        "value": ""
    }
],
"mountPoints": [
    {
        "containerPath": "",
        "readOnly": true,
        "sourceVolume": ""
    }
],
"readonlyRootFilesystem": true,
"privileged": true,
"ulimits": [
    {
        "hardLimit": 0,
        "name": "",
        "softLimit": 0
    }
],
"user": "",
"instanceType": "",
"resourceRequirements": [
    {
        "value": "",
        "type": "MEMORY"
    }
],
"linuxParameters": {
    "devices": [
        {
            "hostPath": "",
            "containerPath": "",
            "permissions": [
                "WRITE"
            ]
        }
    ],
    "initProcessEnabled": true,
    "sharedMemorySize": 0,
    "tmpfs": [
        {
            "containerPath": "",
            "size": 0,
```

```
        "mountOptions": [
            ""
        ]
    },
    "maxSwap": 0,
    "swappiness": 0
},
"logConfiguration": {
    "logDriver": "syslog",
    "options": {
        "KeyName": ""
    },
    "secretOptions": [
        {
            "name": "",
            "valueFrom": ""
        }
    ]
},
"secrets": [
    {
        "name": "",
        "valueFrom": ""
    }
],
"networkConfiguration": {
    "assignPublicIp": "DISABLED"
},
"fargatePlatformConfiguration": {
    "platformVersion": ""
}
},
"nodeProperties": {
    "numNodes": 0,
    "mainNode": 0,
    "nodeRangeProperties": [
        {
            "targetNodes": "",
            "container": {
                "image": "",
                "vcpus": 0,
                "memory": 0,
                "command": [
```

```
    ""
  ],
  "jobRoleArn": "",
  "executionRoleArn": "",
  "volumes": [
    {
      "host": {
        "sourcePath": ""
      },
      "name": "",
      "efsVolumeConfiguration": {
        "fileSystemId": "",
        "rootDirectory": "",
        "transitEncryption": "DISABLED",
        "transitEncryptionPort": 0,
        "authorizationConfig": {
          "accessPointId": "",
          "iam": "ENABLED"
        }
      }
    }
  ],
  "environment": [
    {
      "name": "",
      "value": ""
    }
  ],
  "mountPoints": [
    {
      "containerPath": "",
      "readOnly": true,
      "sourceVolume": ""
    }
  ],
  "readonlyRootFilesystem": true,
  "privileged": true,
  "ulimits": [
    {
      "hardLimit": 0,
      "name": "",
      "softLimit": 0
    }
  ],

```

```
"user": "",
"instanceType": "",
"resourceRequirements": [
  {
    "value": "",
    "type": "MEMORY"
  }
],
"linuxParameters": {
  "devices": [
    {
      "hostPath": "",
      "containerPath": "",
      "permissions": [
        "WRITE"
      ]
    }
  ],
  "initProcessEnabled": true,
  "sharedMemorySize": 0,
  "tmpfs": [
    {
      "containerPath": "",
      "size": 0,
      "mountOptions": [
        ""
      ]
    }
  ],
  "maxSwap": 0,
  "swappiness": 0
},
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {
    "KeyName": ""
  },
  "secretOptions": [
    {
      "name": "",
      "valueFrom": ""
    }
  ]
},
```

```

        "secrets": [
            {
                "name": "",
                "valueFrom": ""
            }
        ],
        "networkConfiguration": {
            "assignPublicIp": "DISABLED"
        },
        "fargatePlatformConfiguration": {
            "platformVersion": ""
        }
    }
}
],
},
"retryStrategy": {
    "attempts": 0,
    "evaluateOnExit": [
        {
            "onStatusReason": "",
            "onReason": "",
            "onExitCode": "",
            "action": "RETRY"
        }
    ]
},
"propagateTags": true,
"timeout": {
    "attemptDurationSeconds": 0
},
"tags": {
    "KeyName": ""
},
"platformCapabilities": [
    "EC2"
],
"eksProperties": {
    "podProperties": {
        "serviceAccountName": "",
        "hostNetwork": true,
        "dnsPolicy": "",
        "containers": [
            {

```



```
    "name": "",
    "image": "",
    "imagePullPolicy": "",
    "command": [
        ""
    ],
    "args": [
        ""
    ],
    "env": [
        {
            "name": "",
            "value": ""
        }
    ],
    "resources": {
        "limits": {
            "KeyName": ""
        },
        "requests": {
            "KeyName": ""
        }
    },
    "volumeMounts": [
        {
            "name": "",
            "mountPath": "",
            "readOnly": true
        }
    ],
    "securityContext": {
        "runAsUser": 0,
        "runAsGroup": 0,
        "privileged": true,
        "readOnlyRootFilesystem": true,
        "runAsNonRoot": true
    }
},
"volumes": [
    {
        "name": "",
        "hostPath": {
            "path": ""
```

```
    },
    "emptyDir": {
      "medium": "",
      "sizeLimit": ""
    },
    "secret": {
      "secretName": "",
      "optional": true
    }
  }
]
}
```

Note

Anda dapat membuat template definisi pekerjaan kontainer tunggal dengan perintah berikut:
AWS CLI

```
$ aws batch register-job-definition --generate-cli-skeleton
```

Parameter definisi Job untuk ContainerProperties

Definisi Job yang [ContainerProperties](#) digunakan dibagi menjadi beberapa bagian:

- nama definisi pekerjaan
- jenis definisi pekerjaan
- default placeholder substitusi parameter
- properti kontainer untuk pekerjaan
- properti Amazon EKS untuk definisi pekerjaan yang diperlukan untuk pekerjaan yang dijalankan di sumber daya Amazon EKS
- properti node yang diperlukan untuk pekerjaan paralel multi-node
- kemampuan platform yang diperlukan untuk pekerjaan yang dijalankan pada sumber daya Fargate
- rincian propagasi tag default dari definisi pekerjaan
- strategi coba lagi default untuk definisi pekerjaan

- prioritas penjadwalan default untuk definisi pekerjaan
- tag default untuk definisi pekerjaan
- batas waktu default untuk definisi pekerjaan

Daftar Isi

- [Nama ketentuan tugas](#)
- [Tipe](#)
- [Parameter-parameter](#)
- [Properti kontainer](#)
- [Properti Amazon EKS](#)
- [Kemampuan platform](#)
- [Propagasi tanda](#)
- [Properti simpul](#)
- [Strategi percobaan kembali](#)
- [Prioritas penjadwalan](#)
- [Tag](#)
- [Waktu habis](#)

Nama ketentuan tugas

jobDefinitionName

Ketika mendaftarkan ketentuan tugas, Anda menentukan namanya. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_). Ketentuan tugas pertama yang terdaftar dengan nama tersebut diberi revisi 1. Ketentuan tugas berikutnya yang terdaftar dengan nama tersebut akan diberi nomor revisi yang makin besar.

Tipe: String

Diperlukan: Ya

Tipe

type

Ketika mendaftarkan ketentuan tugas, Anda menentukan tipe tugas. Jika tugas berjalan di sumber daya Fargate, `multinode` tidak didukung. Untuk informasi lebih lanjut tentang tugas paralel multisimpul, lihat [the section called “Membuat ketentuan tugas paralel multisimpul”](#).

Tipe: String

Nilai yang valid: `container` | `multinode`

Diperlukan: Ya

Parameter-parameter

parameters

Saat mengirimkan pekerjaan, Anda dapat menentukan parameter yang menggantikan placeholder atau mengganti parameter definisi pekerjaan default. Parameter dalam permintaan pengiriman tugas lebih diutamakan daripada default dalam ketentuan tugas. Ini berarti Anda dapat menggunakan definisi pekerjaan yang sama untuk beberapa pekerjaan yang menggunakan format yang sama. Anda juga dapat mengubah nilai secara terprogram dalam perintah pada waktu pengiriman.

Tipe: Peta antar string

Diperlukan: Tidak

Ketika mendaftarkan ketentuan tugas, Anda dapat menggunakan placeholder substitusi parameter di bidang `command` properti kontainer tugas. Sintaksnya adalah sebagai berikut.

```
"command": [  
  "ffmpeg",  
  "-i",  
  "Ref::inputfile",  
  "-c",  
  "Ref::codec",  
  "-o",  
  "Ref::outputfile"
```

```
]
```

Dalam contoh di atas, terdapat placeholder substitusi parameter *Ref::inputfile*, *Ref::codec*, dan *Ref::outputfile* dalam perintah. Anda dapat menggunakan objek `parameters` dalam ketentuan tugas untuk menetapkan nilai default untuk placeholder ini. Misalnya, untuk mengatur default untuk placeholder *Ref::codec*, Anda menentukan hal berikut dalam ketentuan tugas:

```
"parameters" : {"codec" : "mp4"}
```

Ketika ketentuan tugas ini dikirim untuk dijalankan, argumen *Ref::codec* dalam perintah untuk kontainer diganti dengan nilai default, `mp4`.

Properti kontainer

Saat Anda mendaftarkan definisi pekerjaan, tentukan daftar properti kontainer yang diteruskan ke daemon Docker pada instance container saat pekerjaan ditempatkan. Properti kontainer berikut diperbolehkan dalam ketentuan tugas. Untuk tugas simpul tunggal, properti kontainer ini ditetapkan pada tingkat ketentuan tugas. Untuk tugas paralel multisimpul, properti kontainer ditetapkan pada tingkat [Properti simpul](#), untuk setiap grup simpul.

command

Perintah yang disampaikan ke kontainer. Parameter ini sesuai dengan `Cmd` di bagian [Buat kontainer](#) di [Docker Remote API](#) dan parameter `COMMAND` untuk [docker run](#). Untuk informasi lebih lanjut tentang parameter CMD Docker, lihat <https://docs.docker.com/engine/reference/builder/#cmd>.

```
"command": ["string", ...]
```

Tipe: Array string

Diperlukan: Tidak

environment

Variabel lingkungan untuk disampaikan ke kontainer. Parameter ini sesuai dengan `Env` di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--env` untuk [docker run](#).

⚠ Important

Kami tidak merekomendasikan Anda menggunakan variabel lingkungan plaintext untuk informasi sensitif, seperti data kredensial.

ℹ Note

Variabel lingkungan tidak boleh dimulai dengan `AWS_BATCH`. Konvensi penamaan ini dicadangkan untuk variabel yang ditetapkan oleh AWS Batch layanan.

Jenis: Array pasangan nilai-kunci

Diperlukan: Tidak

`name`

Nama variabel lingkungan.

Jenis: String

Wajib: Ya, ketika `environment` digunakan.

`value`

Nilai variabel lingkungan.

Jenis: String

Wajib: Ya, ketika `environment` digunakan.

```
"environment" : [  
  { "name" : "envName1", "value" : "envValue1" },  
  { "name" : "envName2", "value" : "envValue2" }  
]
```

`executionRoleArn`

Ketika mendaftarkan ketentuan tugas, Anda dapat menentukan IAM role. Peran tersebut memberikan izin kepada agen kontainer Amazon ECS untuk memanggil tindakan API yang

ditentukan dalam kebijakannya yang terkait atas nama Anda. Pekerjaan yang berjalan pada sumber daya Fargate harus memberikan peran eksekusi. Untuk informasi selengkapnya, lihat [AWS Batch eksekusi peran IAM](#).

Tipe: String

Wajib: Tidak

`fargatePlatformConfiguration`

Konfigurasi platform untuk pekerjaan yang berjalan pada sumber daya Fargate. Tugas yang berjalan pada sumber daya EC2 tidak harus menentukan parameter ini.

Tipe: Objek [FargatePlatformConfiguration](#)

Diperlukan: Tidak

`platformVersion`

Versi platform AWS Fargate digunakan untuk pekerjaan, atau LATEST untuk menggunakan versi terbaru yang disetujui dari platform Fargate AWS .

Jenis: String

Default: LATEST

Diperlukan: Tidak

`image`

Citra yang digunakan untuk memulai tugas. String ini disampaikan langsung ke daemon Docker. Gambar di registri Docker Hub tersedia secara default. Anda juga dapat menentukan repositori lainnya dengan *repository-url/image:tag*. Maksimum 255 huruf (huruf besar dan huruf kecil), angka, tanda hubung, garis bawah, titik dua, titik miring ke depan, dan tanda pagar diperbolehkan. Parameter ini sesuai dengan Image di bagian [Buat kontainer](#) di [Docker Remote API](#) dan IMAGE parameter [docker run](#).

 Note

Dockerarsitektur gambar harus sesuai dengan arsitektur prosesor dari sumber daya komputasi yang dijadwalkan. Misalnya, Docker gambar Arm berbasis hanya dapat berjalan pada sumber daya komputasi Arm berbasis.

- Gambar di Amazon ECR Repositori Publik menggunakan konvensi lengkap `registry/repository[:tag]` atau `registry/repository[@digest]` penamaan (misalnya, `public.ecr.aws/registry_alias/my-web-app:latest`)
- Gambar di repositori Amazon ECR menggunakan konvensi penamaan `registry/repository:[tag]` lengkap. Misalnya, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`.
- Gambar di repositori resmi di Docker Hub menggunakan satu nama (misalnya, `ubuntu` atau `mongo`).
- Gambar di repositori lain di Docker Hub memenuhi syarat dengan nama organisasi (misalnya, `amazon/amazon-ecs-agent`).
- Gambar di repositori online lainnya memenuhi syarat lebih lanjut dengan nama domain (misalnya, `quay.io/assemblyline/ubuntu`).

Tipe: String

Diperlukan: Ya

instanceType

Tipe instans yang digunakan untuk tugas paralel multisimpul. Semua kelompok simpul dalam pekerjaan paralel multi-simpul harus menggunakan tipe instans yang sama. Parameter ini tidak valid untuk pekerjaan kontainer simpul tunggal atau untuk pekerjaan yang berjalan pada sumber daya Fargate.

Tipe: String

Wajib: Tidak

jobRoleArn

Ketika mendaftarkan ketentuan tugas, Anda dapat menentukan IAM role. Peran tersebut memberikan izin kepada kontainer tugas untuk memanggil tindakan API yang ditentukan dalam kebijakannya yang terkait atas nama Anda. Untuk informasi lebih lanjut, lihat [IAM Role untuk Tugas](#) dalam Panduan Developer Amazon Elastic Container Service.

Tipe: String

Wajib: Tidak

linuxParameters

Modifikasi khusus Linux yang diterapkan ke kontainer, seperti detail untuk pemetaan perangkat.


```

"linuxParameters": {
  "devices": [
    {
      "hostPath": "string",
      "containerPath": "string",
      "permissions": [
        "READ", "WRITE", "MKNOD"
      ]
    }
  ],
  "initProcessEnabled": true/false,
  "sharedMemorySize": 0,
  "tmpfs": [
    {
      "containerPath": "string",
      "size": integer,
      "mountOptions": [
        "string"
      ]
    }
  ],
  "maxSwap": integer,
  "swappiness": integer
}


```

Tipe: Objek [LinuxParameters](#)

Diperlukan: Tidak

devices

Daftar perangkat yang dipetakan ke kontainer. Parameter ini sesuai dengan Devices di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--device` untuk [docker run](#).

 Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Tipe: Array objek [Perangkat](#)

Diperlukan: Tidak

hostPath

Jalur tempat perangkat tersedia dalam instans kontainer host.

Tipe: String

Diperlukan: Ya

containerPath

Jalur tempat perangkat diungkap dalam kontainer. Jika ini tidak ditentukan, perangkat akan diungkap di jalur yang sama dengan jalur host.

Tipe: String

Wajib: Tidak

permissions

Izin untuk perangkat dalam kontainer. Jika hal ini tidak ditentukan, izin ditetapkan ke READ, WRITE, dan MKNOD.

Tipe: Array string

Diperlukan: Tidak

Nilai yang valid: READ | WRITE | MKNOD

initProcessEnabled

Jika betul, jalankan proses `init` di dalam kontainer yang meneruskan sinyal dan melakukan reaping pada proses. Parameter ini sesuai dengan opsi `--init` untuk [docker run](#). Parameter ini memerlukan Docker Remote API versi 1.25 atau lebih besar di instans kontainer Anda. Untuk memeriksa versi Docker Remote API di instans kontainer Anda, masuk ke instans kontainer Anda dan jalankan perintah berikut: `sudo docker version | grep "Server API version"`

Tipe: Boolean


Wajib: Tidak

maxSwap

Jumlah total memori swap (dalam MiB) yang dapat digunakan oleh tugas. Parameter ini diterjemahkan ke opsi `--memory-swap` untuk [docker run](#) di mana nilai adalah jumlah memori

kontainer ditambah nilai `maxSwap`. Untuk informasi lebih lanjut, lihat [detail --memory-swap](#) dalam dokumentasi Docker.

Jika nilai `maxSwap` sebesar `0` ditentukan, kontainer tidak menggunakan swap. Nilai yang diterima adalah `0` atau bilangan bulat positif. Jika `maxSwap` parameter dihilangkan, kontainer menggunakan konfigurasi swap untuk instance container yang dijelankannya. Nilai `maxSwap` harus ditetapkan untuk parameter `swappiness` yang akan digunakan.

 Note


Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Tipe: Integer

Wajib: Tidak

`sharedMemorySize`

Nilai untuk ukuran (dalam MiB) volume `/dev/shm`. Parameter ini sesuai dengan opsi `--shm-size` untuk [docker run](#).

 Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Tipe: Integer


Wajib: Tidak

`swappiness`

Anda dapat menggunakan ini untuk menyetel perilaku `swappiness` memori kontainer. Nilai `swappiness` sebesar `0` menyebabkan swapping tidak terjadi kecuali benar-benar diperlukan. Nilai `swappiness` sebesar `100` menyebabkan halaman ditukar secara agresif. Nilai yang diterima adalah bilangan bulat antara `0` dan `100`. Jika parameter `swappiness` tidak ditentukan, nilai default sebesar `60` akan digunakan. Jika nilai tidak ditentukan untuk `maxSwap`, parameter ini akan diabaikan. Jika `maxSwap` diatur ke `0`, kontainer tidak menggunakan swap. Parameter ini sesuai dengan opsi `--memory-swappiness` untuk [docker run](#).


Pertimbangkan hal berikut ketika Anda menggunakan konfigurasi swap per kontainer.

- Ruang tukar harus diaktifkan dan dialokasikan di instans kontainer untuk kontainer yang akan digunakan.

 Note

AMI yang dioptimalkan untuk Amazon ECS tidak memiliki swap yang diaktifkan secara default. Anda harus mengaktifkan swap di instans untuk menggunakan fitur ini. Untuk informasi lebih lanjut, lihat [Volume Swap Penyimpanan Instans](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux atau [Bagaimana cara mengalokasikan memori agar berfungsi sebagai ruang tukar di instans Amazon EC2 dengan menggunakan file swap?](#).

- Parameter ruang tukar hanya didukung untuk ketentuan tugas yang menggunakan sumber daya EC2.
- Jika swappiness parameter maxSwap dan dihilangkan dari definisi pekerjaan, setiap kontainer memiliki swappiness nilai default 60. Total penggunaan swap dibatasi hingga dua kali reservasi memori kontainer.

 Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.


Tipe: Integer

Wajib: Tidak

tmpfs

Jalur kontainer, opsi pemasangan, dan ukuran pemasangan tmpfs.

Tipe: Array objek [Tmpfs](#)

 Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Diperlukan: Tidak

containerPath

Jalur file absolut dalam kontainer tempat volume tmpfs terpasang.

Tipe: String

Diperlukan: Ya

mountOptions

Daftar opsi pemasangan volume tmpfs.

Nilai yang valid: "defaults" | "ro" | "rw" | "suid" | "nosuid" | "dev" | "nodev" | "exec" | "noexec" | "sync" | "async" | "dirsync" | "remount" | "mand" | "nomand" | "atime" | "noatime" | "diratime" | "nodiratime" | "bind" | "rbind" | "unbindable" | "runbindable" | "private" | "rprivate" | "shared" | "rshared" | "slave" | "rslave" | "relatime" | "norelatime" | "strictatime" | "nostrictatime" | "mode" | "uid" | "gid" | "nr_inodes" | "nr_blocks" | "mpol"

Tipe: Array string

Diperlukan: Tidak

size

Ukuran (dalam MiB) volume tmpfs.

Jenis: Integer

Diperlukan: Ya

logConfiguration

Spesifikasi konfigurasi log untuk tugas.

Parameter ini sesuai dengan LogConfig di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--log-driver` untuk [docker run](#). Secara default, kontainer menggunakan driver pencatatan yang sama dengan yang digunakan daemon Docker. Namun, penampung dapat menggunakan driver logging yang berbeda dari daemon Docker dengan menentukan driver log dengan parameter ini dalam definisi container. Untuk menggunakan driver pencatatan yang berbeda untuk suatu kontainer, sistem log harus dikonfigurasi di instans kontainer atau di server log lain untuk menyediakan opsi pencatatan jarak jauh. Untuk informasi lebih lanjut tentang opsi untuk berbagai driver log yang didukung, lihat [Mengonfigurasi driver pencatatan](#) dalam dokumentasi Docker.

Note

AWS Batch saat ini mendukung subset dari driver logging yang tersedia untuk daemon Docker (ditampilkan dalam tipe data). [LogConfiguration](#)

Parameter ini memerlukan Docker Remote API versi 1.18 atau lebih besar pada instans kontainer Anda. Untuk memeriksa versi Docker Remote API pada instans kontainer Anda, masuk ke instans kontainer Anda dan jalankan perintah berikut: `sudo docker version | grep "Server API version"`

```
"logConfiguration": {
  "devices": [
    {
      "logDriver": "string",
      "options": {
        "optionName1" : "optionValue1",
        "optionName2" : "optionValue2"
      }
      "secretOptions": [
        {
          "name" : "secretOptionName1",
          "valueFrom" : "secretOptionArn1"
        },
        {
          "name" : "secretOptionName2",
          "valueFrom" : "secretOptionArn2"
        }
      ]
    }
  ]
}
```

Tipe: Objek [LogConfiguration](#)

Diperlukan: Tidak

logDriver

Driver log yang akan digunakan untuk tugas. Secara default, AWS Batch mengaktifkan driver `awslogs log`. Nilai valid yang tercantum untuk parameter ini adalah driver log yang dapat berkomunikasi dengan agen penampung Amazon ECS secara default.

Parameter ini sesuai dengan LogConfig di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--log-driver` untuk [docker run](#). Secara default, tugas menggunakan driver pencatatan yang sama dengan yang digunakan daemon Docker. Namun, tugas dapat menggunakan driver pencatatan yang berbeda dari daemon Docker dengan menentukan driver log dengan parameter ini dalam ketentuan tugas. Jika Anda ingin menentukan driver logging lain untuk suatu pekerjaan, sistem log harus dikonfigurasi pada instance container di lingkungan komputasi. Atau, sebagai alternatif, konfigurasi di server log lain untuk menyediakan opsi logging jarak jauh. Untuk informasi lebih lanjut tentang opsi untuk berbagai driver log yang didukung, lihat [Mengonfigurasi driver pencatatan](#) dalam dokumentasi Docker.

Note

AWS Batch saat ini mendukung subset dari driver logging yang tersedia untuk daemon Docker. Driver log tambahan mungkin tersedia di rilis mendatang agen kontainer Amazon ECS.

Driver log yang didukung adalah `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries`, `syslog`, dan `splunk`.

Note

Pekerjaan yang berjalan pada sumber daya Fargate dibatasi untuk driver `splunk log`, `awslogs` dan `log`.

Parameter ini memerlukan Docker Remote API versi 1.18 atau lebih besar pada instans kontainer Anda. Untuk memeriksa versi Docker Remote API pada instans kontainer Anda, masuk ke instans kontainer Anda dan jalankan perintah berikut: `sudo docker version | grep "Server API version"`

Note

Agan penampung Amazon ECS yang berjalan pada instance container harus mendaftarkan driver logging yang tersedia pada instance tersebut dengan variabel `ECS_AVAILABLE_LOGGING_DRIVERS` lingkungan. Jika tidak, kontainer yang ditempatkan di instans tersebut tidak dapat menggunakan opsi konfigurasi log ini.

Untuk informasi lebih lanjut, lihat, [Konfigurasi Agen Kontainer Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

awslogs

Menentukan driver logging Amazon CloudWatch Logs. Untuk informasi selengkapnya, lihat [Menggunakan driver log awslogs](#) dan [driver logging Amazon CloudWatch Logs](#) di dokumentasi Docker.

fluentd

Menentukan driver pencatatan Fluentd. Untuk informasi selengkapnya termasuk penggunaan dan opsi, lihat [Driver logging fluentd](#) di dokumentasi Docker.

gelf

Menentukan driver pencatatan Graylog Extended Format (GELF). Untuk informasi selengkapnya termasuk penggunaan dan opsi, lihat [driver logging Graylog Extended Format](#) di dokumentasi Docker.

journald

Menentukan driver pencatatan journald. Untuk informasi selengkapnya termasuk penggunaan dan opsi, lihat [driver logging Journald](#) di dokumentasi Docker.

json-file

Menentukan driver pencatatan file JSON. Untuk informasi selengkapnya termasuk penggunaan dan opsi, lihat [Driver logging File JSON](#) di dokumentasi Docker.

splunk

Menentukan driver pencatatan Splunk. Untuk informasi selengkapnya termasuk penggunaan dan opsi, lihat [Splunk logging driver](#) di dokumentasi Docker.

syslog

Menentukan driver pencatatan syslog. Untuk informasi selengkapnya termasuk penggunaan dan opsi, lihat [Driver logging Syslog](#) di dokumentasi Docker.

Tipe: String

Diperlukan: Ya

Nilai yang valid: awslogs | fluentd | gelf | journald | json-file | splunk | syslog

Note

Jika Anda memiliki driver khusus yang tidak terdaftar sebelumnya yang ingin Anda gunakan dengan agen kontainer Amazon ECS, Anda dapat melakukan fork proyek agen kontainer Amazon ECS yang [tersedia GitHub](#) dan menyesuaikannya agar berfungsi dengan driver tersebut. Kami mendorong Anda untuk mengirim permintaan tarik untuk perubahan yang ingin Anda sertakan. Namun, Amazon Web Services saat ini tidak mendukung permintaan yang menjalankan salinan modifikasi dari perangkat lunak ini.

options

Opsi konfigurasi log untuk dikirim ke driver log untuk tugas.

Parameter ini memerlukan Docker Remote API versi 1.19 atau lebih besar di instans kontainer Anda.

Tipe: Peta antar string

Diperlukan: Tidak

secretOptions

Objek yang mewakili rahasia untuk diteruskan ke konfigurasi log. Untuk informasi selengkapnya, lihat [Menentukan data sensitif](#).

Tipe: array objek

Diperlukan: Tidak

name

Nama opsi driver log yang akan ditetapkan dalam tugas.

Tipe: String

Diperlukan: Ya

valueFrom

Nama Sumber Daya Amazon (ARN) dari rahasia untuk mengekspos ke konfigurasi log wadah. Nilai yang didukung adalah ARN lengkap dari rahasia Secrets Manager atau ARN lengkap dari parameter di SSM Parameter Store.

Note

Jika parameter SSM Parameter Store ada Wilayah AWS sama dengan tugas yang Anda luncurkan, maka Anda dapat menggunakan ARN lengkap atau nama parameter. Jika parameter ada di Wilayah yang berbeda, ARN lengkap harus disebutkan.

Tipe: String

Diperlukan: Ya

memory

Parameter ini sudah usang, gunakan sebagai gantinya. [resourceRequirements](#)

Jumlah memori MiB yang disediakan untuk pekerjaan itu.

Sebagai contoh untuk cara menggunakan [resourceRequirements](#), jika definisi pekerjaan Anda berisi sintaks yang mirip dengan berikut ini.

```
"containerProperties": {  
  "memory": 512  
}
```

Sintaks yang setara menggunakan [resourceRequirements](#) adalah sebagai berikut.

```
"containerProperties": {  
  "resourceRequirements": [  
    {  
      "type": "MEMORY",  
      "value": "512"  
    }  
  ]  
}
```

Jenis: Integer

Diperlukan: Ya

mountPoints

Titik pemasangan untuk volume data dalam kontainer Anda. Parameter ini sesuai dengan `Volumes` di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--volume` untuk [docker run](#).

```
"mountPoints": [  
  {  
    "sourceVolume": "string",  
    "containerPath": "string",  
    "readOnly": true/false  
  }  
]
```

Tipe: Array objek

Diperlukan: Tidak

`sourceVolume`

Nama volume yang akan dipasang.

Jenis: String

Wajib: Ya, ketika `mountPoints` digunakan.

`containerPath`

Path di kontainer tempat memasang volume host.

Jenis: String

Wajib: Ya, ketika `mountPoints` digunakan.

`readOnly`

Jika nilai ini adalah `true`, kontainer memiliki akses hanya-baca ke volume. Jika nilai ini adalah `false`, kontainer bisa menulis ke volume.

Tipe: Boolean

Wajib: Tidak

Default: Salah

networkConfiguration

Konfigurasi jaringan untuk pekerjaan yang berjalan pada sumber daya Fargate. Tugas yang berjalan pada sumber daya EC2 tidak harus menentukan parameter ini.

```
"networkConfiguration": {  
  "assignPublicIp": "string"  
}
```

Tipe: Array objek

Diperlukan: Tidak

assignPublicIp

Menunjukkan apakah pekerjaan memiliki alamat IP publik. Hal ini diperlukan jika tugas membutuhkan akses jaringan keluar.

Tipe: String

Nilai yang valid: ENABLED | DISABLED

Wajib: Tidak

Default: DISABLED

privileged

Jika parameter ini betul, kontainer akan diberikan izin tingkat tinggi di instans kontainer host (mirip dengan pengguna root). Parameter ini sesuai dengan Privileged di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--privileged` untuk [docker run](#). Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate. Jangan berikan atau tentukan sebagai false.

```
"privileged": true/false
```

Tipe: Boolean

Wajib: Tidak

readonlyRootFilesystem

Jika parameter ini betul, kontainer diberikan akses hanya-baca ke sistem file asalnya. Parameter ini sesuai dengan ReadonlyRootfs di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--read-only` untuk [docker run](#).

```
"readonlyRootFilesystem": true/false
```

Tipe: Boolean

Wajib: Tidak

resourceRequirements

Jenis dan jumlah sumber daya yang akan ditetapkan ke kontainer. Sumber daya yang didukung mencakup GPU, MEMORY, dan VCPU.

```
"resourceRequirements" : [  
  {  
    "type": "GPU",  
    "value": "number"  
  }  
]
```

Tipe: Array objek

Diperlukan: Tidak

type

Jenis sumber daya yang akan ditetapkan ke kontainer. Sumber daya yang didukung mencakup GPU, MEMORY, dan VCPU.

Jenis: String

Wajib: Ya, ketika `resourceRequirements` digunakan.

value

Kuantitas sumber daya yang ditentukan untuk disimpan bagi kontainer. Nilainya bervariasi berdasarkan `type` yang ditentukan.

`type="GPU"`

Jumlah GPU fisik yang akan disimpan untuk kontainer. Jumlah GPU yang dicadangkan untuk semua kontainer dalam suatu pekerjaan tidak dapat melebihi jumlah GPU yang tersedia pada sumber daya komputasi tempat pekerjaan diluncurkan.

`tipe="MEMORY"`

Batas keras (dalam MiB) memori yang diberikan ke kontainer. Jika kontainer Anda mencoba untuk melebihi memori yang ditentukan di sini, kontainer akan dimatikan. Parameter ini sesuai dengan Memory di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--memory` untuk [docker run](#). Anda harus menentukan setidaknya 4 MiB memori untuk sebuah tugas. Hal ini wajib, tetapi dapat ditentukan di beberapa tempat untuk tugas paralel multisimpul (MNP). Ini harus ditentukan untuk setiap simpul setidaknya satu kali. Parameter ini sesuai dengan Memory di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--memory` untuk [docker run](#).

Note

Jika Anda mencoba memaksimalkan pemanfaatan sumber daya dengan menyediakan memori sebanyak mungkin bagi tugas untuk tipe instans tertentu, lihat [Manajemen Memori Sumber Daya Komputasi](#).

Untuk pekerjaan yang berjalan pada sumber daya Fargate, maka `value` harus cocok dengan salah satu nilai yang didukung. Selain itu, VCPU nilai harus menjadi salah satu nilai yang didukung untuk nilai memori tersebut.

VCPU	MEMORY
0,25 vCPU	512, 1024, dan 2048 MiB
0,5 vCPU	1024-4096 MiB dalam peningkatan 1024 MiB
1 vCPU	2048-8192 MiB dalam peningkatan 1024 MiB
2 vCPU	4096-16384 MiB dalam peningkatan 1024 MiB
4 vCPU	8192-30720 MiB dengan peningkatan 1024 MiB
8 vCPU	16384-61440 MiB dengan peningkatan 4096 MiB
16 vCPU	32768-122880 MiB dengan peningkatan 8192 MiB

`tipe="VCPU"`

Jumlah vCPUs yang disimpan untuk tugas. Parameter ini sesuai dengan `CpuShares` di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--cpu-shares` untuk [docker run](#). Setiap vCPU setara dengan 1.024 bagian CPU. Untuk pekerjaan yang berjalan pada sumber daya EC2, Anda harus menentukan setidaknya satu vCPU. Hal ini wajib, tetapi dapat ditentukan di beberapa tempat. Itu harus ditentukan setidaknya sekali untuk setiap node.

Untuk pekerjaan yang berjalan di sumber daya Fargate, `value` harus cocok dengan salah satu nilai yang didukung dan `MEMORY` nilai harus menjadi salah satu nilai yang didukung untuk nilai VCPU tersebut. Nilai yang didukung adalah 0,25, 0,5, 1, 2, 4, 8, dan 16.

Default untuk kuota jumlah sumber daya vCPU Fargate On-Demand adalah 6 vCPU. Untuk informasi lebih lanjut tentang kuota Fargate, lihat kuota [AWS Fargate](#) di. Referensi Umum Amazon Web

Jenis: String

Wajib: Ya, ketika `resourceRequirements` digunakan.

secrets

Rahasia untuk tugas yang diungkap sebagai variabel lingkungan. Untuk informasi selengkapnya, lihat [Menentukan data sensitif](#).

```
"secrets": [  
  {  
    "name": "secretName1",  
    "valueFrom": "secretArn1"  
  },  
  {  
    "name": "secretName2",  
    "valueFrom": "secretArn2"  
  }  
  ...  
]
```

Tipe: Array objek

Diperlukan: Tidak

name

Nama variabel lingkungan yang berisi rahasia.

Jenis: String

Wajib: Ya, ketika secrets digunakan.

valueFrom

Rahasia yang akan diungkap ke kontainer. Nilai yang didukung adalah Nama Sumber Daya Amazon (ARN) lengkap dari rahasia Secrets Manager atau ARN lengkap parameter di SSM Parameter Store.

Note

Jika parameter Penyimpanan Parameter SSM Wilayah AWS sama dengan pekerjaan yang Anda luncurkan, maka Anda dapat menggunakan ARN lengkap atau nama parameter. Jika parameter ada di Wilayah yang berbeda, ARN lengkap harus disebutkan.

Jenis: String

Wajib: Ya, ketika secrets digunakan.

ulimits

Daftar nilai `ulimits` yang akan ditetapkan dalam kontainer. Parameter ini sesuai dengan `Ulimits` di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--ulimit` untuk [docker run](#).

```
"ulimits": [  
  {  
    "name": string,  
    "softLimit": integer,  
    "hardLimit": integer  
  }  
  ...  
]
```

Tipe: Array objek

Diperlukan: Tidak

name

Parameter type dari `ulimit`.

Jenis: String

Wajib: Ya, ketika `ulimits` digunakan.

hardLimit

Batas keras untuk tipe `ulimit`.

Jenis: Integer

Wajib: Ya, ketika `ulimits` digunakan.

softLimit

Batas lunak untuk tipe `ulimit`.

Jenis: Integer

Wajib: Ya, ketika `ulimits` digunakan.

user

Nama pengguna untuk digunakan di dalam kontainer. Parameter ini sesuai dengan `User` di bagian [Buat kontainer](#) di [Docker Remote API](#) dan opsi `--user` untuk [docker run](#).

```
"user": "string"
```

Tipe: String

Wajib: Tidak

vcpus

Parameter ini sudah usang, gunakan sebagai gantinya. [resourceRequirements](#)

Jumlah vCPU yang disimpan untuk kontainer.

Sebagai contoh cara menggunakan `resourceRequirements`, jika definisi pekerjaan Anda berisi baris yang mirip dengan ini:

```
"containerProperties": {  
  "vcpus": 2  
}
```

Garis ekivalen [resourceRequirements](#) yang digunakan adalah sebagai berikut.

```
"containerProperties": {  
  "resourceRequirements": [  
    {  
      "type": "VCPU",  
      "value": "2"  
    }  
  ]  
}
```

Jenis: Integer

Diperlukan: Ya

volumes

Ketika mendaftarkan ketentuan tugas, Anda dapat menentukan daftar volume yang disampaikan ke daemon Docker di instans kontainer. Parameter berikut diperbolehkan dalam properti kontainer:

```
"volumes": [  
  {  
    "name": "string",  
    "host": {  
      "sourcePath": "string"  
    },  
    "efsVolumeConfiguration": {  
      "authorizationConfig": {  
        "accessPointId": "string",  
        "iam": "string"  
      },  
      "fileSystemId": "string",  
      "rootDirectory": "string",  
      "transitEncryption": "string",
```

```
    "transitEncryptionPort": number
  }
}
]
```

name

Nama volume. Maksimum 255 huruf (huruf besar dan huruf kecil), angka, tanda hubung, dan garis bawah diperbolehkan. Nama ini direferensikan dalam parameter `sourceVolume` dari ketentuan kontainer `mountPoints`.

Tipe: String

Wajib: Tidak

host

Isi dari parameter `host` menentukan apakah volume data Anda tetap di instans kontainer `host` atau tidak dan di mana itu disimpan. Jika parameter `host` kosong, daemon Docker menetapkan jalur `host` untuk volume data Anda. Namun, data tidak dijamin bertahan setelah kontainer yang terkait dengannya berhenti berjalan.

Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Tipe: Objek

Diperlukan: Tidak

sourcePath

Jalur di instans kontainer `host` yang disajikan untuk kontainer. Jika parameter ini kosong, daemon Docker akan menetapkan jalur `host` untuk Anda.

Jika parameter `host` berisi lokasi file `sourcePath`, volume data akan tetap di lokasi yang ditentukan di instans kontainer `host` sampai Anda menghapusnya secara manual. Jika nilai `sourcePath` tidak ada di instans kontainer `host`, daemon Docker akan membuatnya. Jika lokasi ada, konten folder jalur sumber akan diekspor.

Tipe: String

Wajib: Tidak

`efsVolumeConfiguration`

Parameter ini ditentukan ketika Anda menggunakan sistem file Amazon Elastic File System untuk penyimpanan tugas. Untuk informasi selengkapnya, lihat [Volume Amazon EFS](#).

Tipe: Objek

Diperlukan: Tidak

`authorizationConfig`

Detail konfigurasi otorisasi untuk sistem file Amazon EFS.

Tipe: String

Wajib: Tidak

`accessPointId`

ID titik akses Amazon EFS yang akan digunakan. Jika jalur akses ditentukan, nilai direktori asal yang ditentukan di `EFSVolumeConfiguration` harus diabaikan atau diatur ke `/`. Ini akan memberlakukan jalur yang ditetapkan pada titik akses EFS. Jika titik akses digunakan, enkripsi transit harus diaktifkan di `EFSVolumeConfiguration`. Untuk informasi lebih lanjut, lihat [Bekerja dengan Titik Akses Amazon EFS](#) dalam Panduan Pengguna Amazon Elastic File System.

Tipe: String

Wajib: Tidak

`iam`

Menentukan apakah akan menggunakan peran IAM AWS Batch job yang ditentukan dalam definisi pekerjaan saat memasang sistem file Amazon EFS. Jika diaktifkan, enkripsi transit harus diaktifkan di `EFSVolumeConfiguration`. Jika parameter ini diabaikan, nilai default `DISABLED` akan digunakan. Untuk informasi selengkapnya, lihat [Menggunakan titik akses Amazon EFS](#).

Tipe: String

Nilai yang valid: `ENABLED` | `DISABLED`

Diperlukan: Tidak

`fileSystemId`


ID sistem file Amazon EFS yang akan digunakan.

Tipe: String

Wajib: Tidak

`rootDirectory`

Direktori dalam sistem file Amazon EFS untuk pemasangan sebagai direktori asal di dalam host. Jika parameter ini diabaikan, asal volume Amazon EFS akan digunakan. Jika Anda menentukan /, efeknya akan sama seperti menghilangkan parameter ini. Panjang maksimum adalah 4.096 karakter.

 Important

Jika titik akses EFS ditentukan dalam `authorizationConfig`, parameter direktori asal harus diabaikan atau diatur ke /. Ini memberlakukan jalur yang ditetapkan di titik akses Amazon EFS.

Tipe: String

Wajib: Tidak

`transitEncryption`

Menentukan apakah akan mengaktifkan enkripsi untuk data Amazon EFS saat transit antara host Amazon ECS dan server Amazon EFS. Enkripsi transit harus diaktifkan jika otorisasi IAM Amazon EFS digunakan. Jika parameter ini diabaikan, nilai default DISABLED akan digunakan. Untuk informasi lebih lanjut, lihat [Mengkripsi data saat transit](#) dalam Panduan Pengguna Amazon Elastic File System.

Tipe: String

Nilai yang valid: ENABLED | DISABLED

Diperlukan: Tidak

transitEncryptionPort

Port yang akan digunakan saat mengirim data terenkripsi antara host Amazon ECS dan server Amazon EFS. Jika Anda tidak menentukan port enkripsi transit, strategi pemilihan port yang digunakan oleh pembantu pemasangan Amazon EFS akan digunakan. Nilai harus antara 0 dan 65.535. Untuk informasi lebih lanjut, lihat [Pembantu Pemasangan EFS](#) dalam Panduan Pengguna Amazon Elastic File System.

Tipe: Integer

Wajib: Tidak

Properti Amazon EKS

Objek dengan berbagai properti yang khusus untuk pekerjaan berbasis Amazon EKS. Ini tidak boleh ditentukan untuk definisi pekerjaan berbasis Amazon ECS.

podProperties

Properti untuk sumber daya Kubernetes pod suatu pekerjaan.

Tipe: Objek [EksPodProperties](#)

Diperlukan: Tidak

containers

Properti wadah yang digunakan pada pod Amazon EKS.

Tipe: Objek [EksContainer](#)

Diperlukan: Tidak

args

Sebuah array argumen ke entrypoint. Jika ini tidak ditentukan, gambar CMD kontainer digunakan. Ini sesuai dengan args anggota di bagian [Entrypoint](#) dari [Pod](#) di Kubernetes. Referensi variabel lingkungan diperluas menggunakan lingkungan wadah.

Jika variabel lingkungan yang direferensikan tidak ada, referensi dalam perintah tidak diubah. Misalnya, jika referensinya adalah `$(NAME1) ""` dan variabel NAME1 lingkungan tidak ada, string perintah akan tetap `"$(NAME1)." $$` diganti dengan `$`, dan string yang

dihasilkan tidak diperluas. Misalnya, `$$ (VAR_NAME)` diteruskan sebagai `$(VAR_NAME)` apakah variabel `VAR_NAME` lingkungan ada atau tidak. Untuk informasi selengkapnya, lihat [CMD](#) di referensi Dockerfile dan [Mendefinisikan perintah dan argumen untuk pod](#) dalam dokumentasi. Kubernetes

Tipe: Array string

Diperlukan: Tidak

command

Titik masuk untuk wadah. Ini tidak dijalankan di dalam shell. Jika ini tidak ditentukan, gambar `ENTRYPOINT` kontainer digunakan. Referensi variabel lingkungan diperluas menggunakan lingkungan wadah.

Jika variabel lingkungan yang direferensikan tidak ada, referensi dalam perintah tidak diubah. Misalnya, jika referensinya adalah `$(NAME1) ""` dan variabel `NAME1` lingkungan tidak ada, string perintah akan tetap `$(NAME1)."` `$$` diganti dengan `$` dan string yang dihasilkan tidak diperluas. Misalnya, `$$ (VAR_NAME)` akan diteruskan sebagai `$(VAR_NAME)` apakah variabel `VAR_NAME` lingkungan ada atau tidak. Entrypoint tidak dapat diperbarui. Untuk informasi selengkapnya, lihat [ENTRYPOINT](#) dalam referensi Dockerfile dan [Mendefinisikan perintah dan argumen untuk wadah](#) dan [Entrypoint](#) dalam dokumentasi. Kubernetes

Tipe: Array string

Diperlukan: Tidak

env

Variabel lingkungan untuk disampaikan ke kontainer.

Note

Variabel lingkungan tidak dapat dimulai dengan `"AWS_BATCH"`. Konvensi penamaan ini dicadangkan untuk variabel yang AWS Batch menetapkan.

Tipe: Array objek [EksContainerEnvironmentVariable](#)

Diperlukan: Tidak

name

Nama variabel lingkungan.

Tipe: String

Diperlukan: Ya

value

Nilai dari variabel lingkungan.

Tipe: String

Wajib: Tidak

image

Gambar Docker digunakan untuk memulai wadah.

Tipe: String

Diperlukan: Ya

imagePullPolicy

Kebijakan tarik gambar untuk wadah. Nilai yang didukung adalah Always, IfNotPresent, dan Never. Parameter ini menjadi default ke IfNotPresent. Namun, jika :latest tag ditentukan, defaultnya. Always Untuk informasi selengkapnya, lihat [Memperbarui gambar](#) dalam Kubernetesdokumentasi.

Tipe: String

Wajib: Tidak

name

Nama kontainer. Jika nama tidak ditentukan, nama default "Default" digunakan. Setiap kontainer dalam pod harus memiliki nama yang unik.

Tipe: String

Wajib: Tidak

resources

Jenis dan jumlah sumber daya untuk ditugaskan ke kontainer. Sumber daya yang didukung mencakup `memory`, `cpu`, dan `nvidia.com/gpu`. Untuk informasi selengkapnya, lihat [Manajemen sumber daya untuk pod dan kontainer](#) dalam Kubernetes dokumentasi.

Tipe: Objek [EksContainerResourceRequirements](#)

Diperlukan: Tidak

limits

Jenis dan jumlah sumber daya yang akan dicadangkan untuk wadah. Nilai bervariasi berdasarkan name yang ditentukan. Sumber daya dapat diminta menggunakan objek `limits` atau `requests` objek.

memori

Batas keras memori (dalam MiB) untuk wadah, menggunakan seluruh bilangan bulat, dengan akhiran "Mi". Jika kontainer Anda mencoba untuk melampaui memori yang ditentukan, kontainer akan dimatikan. Anda harus menentukan setidaknya 4 MiB memori untuk suatu pekerjaan. `memory` dapat ditentukan dalam `limits`, `requests`, atau keduanya. Jika `memory` ditentukan di kedua tempat, maka nilai yang ditentukan `limits` harus sama dengan nilai yang ditentukan dalam `requests`.

Note

Untuk memaksimalkan pemanfaatan sumber daya Anda, berikan pekerjaan Anda memori sebanyak mungkin untuk jenis instance tertentu yang Anda gunakan. Untuk mempelajari caranya, lihat [Manajemen Memori Sumber Daya Komputasi](#).

cpu

Jumlah CPU yang disediakan untuk wadah. Nilai harus kelipatan genap dari 0.25. `cpu` dapat ditentukan dalam `limits`, `requests`, atau keduanya. Jika `cpu` ditentukan di kedua tempat, maka nilai yang ditentukan `limits` harus setidaknya sebesar nilai yang ditentukan dalam `requests`.

`nvidia.com/gpu`

Jumlah GPU yang dicadangkan untuk wadah. Nilai harus berupa bilangan bulat utuh. `memory` dapat ditentukan dalam `limits`, `requests`, atau keduanya. Jika `memory` ditentukan di kedua tempat, maka nilai yang ditentukan `limits` harus sama dengan nilai yang ditentukan dalam `requests`.

Tipe: Peta string ke string

Batasan Panjang Nilai: Panjang minimum 1. Panjang maksimum 256.


Diperlukan: Tidak

`requests`

Jenis dan jumlah sumber daya untuk meminta wadah. Nilai bervariasi berdasarkan `name` yang ditentukan. Sumber daya dapat diminta dengan menggunakan salah satu `limits` atau `requests` objek.

memori

Batas keras memori (dalam MiB) untuk wadah, menggunakan seluruh bilangan bulat, dengan akhiran "Mi". Jika kontainer Anda mencoba untuk melampaui memori yang ditentukan, kontainer akan dimatikan. Anda harus menentukan setidaknya 4 MiB memori untuk suatu pekerjaan. `memory` dapat ditentukan dalam `limits`, `requests`, atau keduanya. Jika `memory` ditentukan di keduanya, maka nilai yang ditentukan `limits` harus sama dengan nilai yang ditentukan dalam `requests`.

 Note

Jika Anda mencoba memaksimalkan pemanfaatan sumber daya dengan menyediakan memori sebanyak mungkin bagi tugas untuk tipe instans tertentu, lihat [Manajemen Memori Sumber Daya Komputasi](#).

`cpu`

Jumlah CPU yang dicadangkan untuk wadah. Nilai harus kelipatan genap dari 0.25. `cpu` dapat ditentukan dalam `limits`, `requests`, atau keduanya. Jika `cpu` ditentukan dalam keduanya, maka nilai yang ditentukan dalam `limits` harus setidaknya sebesar nilai yang ditentukan dalam `requests`.

`nvidia.com/gpu`

Jumlah GPU yang dicadangkan untuk wadah. Nilai harus berupa bilangan bulat utuh. `nvidia.com/gpu` dapat ditentukan dalam `limits`, `requests`, atau keduanya. Jika `nvidia.com/gpu` ditentukan di keduanya, maka nilai yang ditentukan `limits` harus sama dengan nilai yang ditentukan dalam `requests`.

Tipe: Peta string ke string

Batasan Panjang Nilai: Panjang minimum 1. Panjang maksimum 256.

Diperlukan: Tidak

`securityContext`

Konteks keamanan untuk suatu pekerjaan. Untuk informasi selengkapnya, lihat [Mengonfigurasi konteks keamanan untuk pod atau wadah](#) dalam Kubernetes dokumentasi.

Tipe: Objek [EksContainerSecurityContext](#)

Diperlukan: Tidak

`privileged`

Ketika parameter `inittrue`, penampung diberikan izin tinggi pada instance wadah host. Tingkat izin mirip dengan izin `root` pengguna. Nilai default-nya adalah `false`. Parameter ini memetakan ke `privileged` kebijakan dalam [kebijakan keamanan pod Privileged](#) dalam Kubernetes dokumentasi.

Tipe: Boolean

Wajib: Tidak

`readOnlyRootFilesystem`

Ketika parameter `inittrue`, wadah diberikan akses read-only ke sistem file root nya. Nilai default-nya adalah `false`. Parameter ini memetakan ke `ReadOnlyRootFilesystem` kebijakan dalam [kebijakan keamanan Volume dan sistem file pod](#) dalam Kubernetes dokumentasi.

Tipe: Boolean

Wajib: Tidak

runAsGroup

Ketika parameter ini ditentukan, wadah dijalankan sebagai ID grup yang ditentukan (`gid`). Jika parameter ini tidak ditentukan, default adalah grup yang ditentukan dalam metadata gambar. Parameter ini memetakan ke `RunAsGroup` dan `MustRunAs` kebijakan dalam [kebijakan keamanan pod Pengguna dan grup](#) dalam Kubernetesdokumentasi.

Tipe: Panjang

Diperlukan: Tidak

runAsNonRoot

Ketika parameter ini ditentukan, wadah dijalankan sebagai pengguna dengan `uid` selain 0. Jika parameter ini tidak ditentukan, maka aturan tersebut diberlakukan. Parameter ini memetakan ke `RunAsUser` dan `MustRunAsNonRoot` kebijakan dalam [kebijakan keamanan pod Pengguna dan grup](#) dalam Kubernetesdokumentasi.

Tipe: Panjang

Diperlukan: Tidak

runAsUser

Ketika parameter ini ditentukan, wadah dijalankan sebagai ID pengguna yang ditentukan (`uid`). Jika parameter ini tidak ditentukan, default adalah pengguna yang ditentukan dalam metadata gambar. Parameter ini memetakan ke `RunAsUser` dan `MustRunAs` kebijakan dalam [kebijakan keamanan pod Pengguna dan grup](#) dalam Kubernetesdokumentasi.

Tipe: Panjang

Diperlukan: Tidak

volumeMounts

Volume dipasang untuk wadah untuk pekerjaan Amazon EKS. Untuk informasi selengkapnya tentang volume dan volume mountKubernetes, lihat [Volume](#) dalam Kubernetesdokumentasi.

Tipe: Array objek [EksContainerVolumeMount](#)

Diperlukan: Tidak

`mountPath`

Jalur pada wadah tempat volume dipasang.

Tipe: String

Wajib: Tidak

`name`

Nama volume mount. Ini harus cocok dengan nama salah satu volume di pod.

Tipe: String

Wajib: Tidak

`readOnly`

Jika nilai ini adalah `true`, kontainer memiliki akses hanya-baca ke volume. Jika tidak, kontainer bisa menulis ke volume. Nilai default-nya adalah `false`.

Tipe: Boolean

Wajib: Tidak

`dnsPolicy`

Kebijakan DNS untuk pod. Nilai default-nya adalah `ClusterFirst`. Jika `hostNetwork` parameter tidak ditentukan, defaultnya adalah `ClusterFirstWithHostNet`. `ClusterFirst` menunjukkan bahwa kueri DNS apa pun yang tidak cocok dengan akhiran domain cluster yang dikonfigurasi diteruskan ke server nama hulu yang diwarisi dari node. Jika tidak ada nilai yang ditentukan `dnsPolicy` dalam operasi [RegisterJobDefinitionAPI](#), maka tidak ada nilai yang dikembalikan `dnsPolicy` oleh salah satu dari [DescribeJobDefinitions](#) atau operasi [DescribeJobsAPI](#). Pengaturan spesifikasi pod akan berisi salah satu `ClusterFirst` atau `ClusterFirstWithHostNet`, tergantung pada nilai `hostNetwork` parameternya. Untuk informasi selengkapnya, lihat [kebijakan DNS Pod](#) di Kubernetes dokumentasi.

Nilai yang valid: `Default` | `ClusterFirst` | `ClusterFirstWithHostNet`

Tipe: String

Wajib: Tidak

hostNetwork

Menunjukkan apakah pod menggunakan alamat IP jaringan host. Nilai default-nya adalah `true`. Menyetel ini untuk `false` mengaktifkan model jaringan Kubernetes pod. Sebagian besar AWS Batch beban kerja hanya untuk egress-only dan tidak memerlukan overhead alokasi IP untuk setiap pod untuk koneksi yang masuk. Untuk informasi selengkapnya, lihat [Host namespace](#) dan [jaringan Pod](#) dalam dokumentasi. Kubernetes

Tipe: Boolean

Wajib: Tidak

serviceAccountName

Nama akun layanan yang digunakan untuk menjalankan pod. Untuk informasi selengkapnya, lihat [akun Kubernetes layanan](#) dan [Mengonfigurasi akun Kubernetes layanan untuk mengambil peran IAM](#) dalam Panduan Pengguna Amazon EKS dan [Mengonfigurasi akun layanan untuk pod](#) dalam Kubernetes dokumentasi.

Tipe: String

Wajib: Tidak

volumes

Menentukan volume untuk definisi pekerjaan yang menggunakan sumber daya Amazon EKS.

Tipe: Array objek [EksVolume](#)

Diperlukan: Tidak

kosongDir

Menentukan konfigurasi Kubernetes `emptyDir` volume. `emptyDirVolume` pertama kali dibuat ketika sebuah pod ditugaskan ke sebuah node. Itu ada selama pod itu berjalan pada node itu. `emptyDirVolume` awalnya kosong. Semua kontainer dalam pod dapat membaca dan menulis file dalam `emptyDir` volume. Namun, `emptyDir` volume dapat dipasang pada jalur yang sama atau berbeda di setiap wadah. Ketika sebuah pod dihapus dari node karena alasan apa pun, data di dalamnya akan dihapus `emptyDir` secara permanen. Untuk informasi selengkapnya, lihat [emptyDir](#) dalam dokumentasi. Kubernetes

Tipe: Objek [EksEmptyDir](#)

Diperlukan: Tidak

medium

Media untuk menyimpan volume. Nilai default adalah string kosong, yang menggunakan penyimpanan node.

""

(Default) Gunakan penyimpanan disk node.

“Memori”

Gunakan tmpfs volume yang didukung oleh RAM node. Isi volume hilang saat node reboot, dan penyimpanan apa pun pada volume dihitung terhadap batas memori kontainer.

Tipe: String

Wajib: Tidak

SizeLimit

Ukuran maksimum volume. Secara default, tidak ada ukuran maksimum yang ditentukan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Diperlukan: Tidak

HostPath

Menentukan konfigurasi Kubernetes hostPath volume. hostPathVolume memasang file atau direktori yang sudah ada dari sistem file node host ke dalam pod Anda. Untuk informasi selengkapnya, lihat [HostPath](#) dalam Kubernetes dokumentasi.

Tipe: Objek [EksHostPath](#)

Diperlukan: Tidak

path

Jalur file atau direktori pada host untuk dipasang ke kontainer pada pod.

Tipe: String

Wajib: Tidak

name

Nama volume. Nama harus diizinkan sebagai nama subdomain DNS. Untuk informasi selengkapnya, lihat [nama subdomain DNS](#) dalam dokumentasi. Kubernetes

Tipe: String

Diperlukan: Ya

Rahasia

Menentukan konfigurasi Kubernetes `secret volume`. Untuk informasi lebih lanjut, lihat [rahasia](#) dalam Kubernetes dokumentasi.

Tipe: Objek [EksSecret](#)

Diperlukan: Tidak

opsional

Menentukan apakah rahasia atau kunci rahasia harus didefinisikan.

Tipe: Boolean

Wajib: Tidak

SecretName

Nama rahasia. Nama harus diizinkan sebagai nama subdomain DNS. Untuk informasi selengkapnya, lihat [nama subdomain DNS](#) dalam dokumentasi. Kubernetes

Tipe: String

Diperlukan: Ya

Kemampuan platform

platformCapabilities

Kemampuan platform yang diperlukan oleh ketentuan tugas. Jika tidak ada nilai yang ditentukan, default-nya adalah EC2. Untuk tugas yang berjalan di sumber daya Fargate, nilainya adalah FARGATE.

Note

Jika pekerjaan berjalan pada sumber daya Amazon EKS, maka Anda tidak boleh menentukan `platformCapabilities`.

Tipe: String

Nilai yang valid: EC2 | FARGATE

Diperlukan: Tidak

Propagasi tanda

`propagateTags`

Menentukan apakah akan mempropagasi tanda dari tugas atau ketentuan tugas ke tugas Amazon ECS yang sesuai. Jika tidak ada nilai yang ditentukan, tanda tidak dipropagasi. Tanda hanya dapat dipropagasi ke tugas ketika tugas dibuat. Untuk tanda dengan nama yang sama, tanda tugas diprioritaskan daripada tanda ketentuan tugas. Jika jumlah total tanda gabungan dari tugas dan ketentuan tugas lebih dari 50, tugas dipindahkan ke status FAILED.

Note

Jika pekerjaan berjalan pada sumber daya Amazon EKS, maka Anda tidak boleh menentukan `propagateTags`.

Tipe: Boolean

Wajib: Tidak

Properti simpul

`nodeProperties`

Ketika mendaftarkan ketentuan tugas paralel multisimpul, Anda harus menentukan daftar properti simpul. Properti node ini menentukan jumlah node yang akan digunakan dalam pekerjaan Anda, indeks node utama, dan rentang node yang berbeda untuk digunakan. Jika tugas berjalan di

sumber daya Fargate, Anda tidak dapat menentukan `nodeProperties`. Sebaliknya, gunakan `containerProperties`. Properti simpul berikut diperbolehkan dalam ketentuan tugas. Untuk informasi selengkapnya, lihat [Tugas paralel multisimpul](#).

 Note

Jika pekerjaan berjalan pada sumber daya Amazon EKS, maka Anda tidak boleh menentukannodeProperties.

Tipe: Objek [NodeProperties](#)

Diperlukan: Tidak

`mainNode`

Menentukan indeks simpul untuk simpul utama dari tugas paralel multisimpul. Nilai indeks simpul ini harus lebih kecil dari jumlah simpul.

Jenis: Integer

Diperlukan: Ya

`numNodes`


Jumlah simpul yang terkait dengan tugas paralel multisimpul.

Jenis: Integer

Diperlukan: Ya

`nodeRangeProperties`

Daftar rentang simpul dan propertinya yang terkait dengan tugas paralel multisimpul.

 Note

Grup node adalah grup node pekerjaan yang identik yang semuanya berbagi properti kontainer yang sama. Anda dapat menggunakan AWS Batch untuk menentukan hingga lima grup node yang berbeda untuk setiap pekerjaan.

Tipe: Array objek [NodeRangeProperty](#)

Diperlukan: Ya

`targetNodes`

Rentang simpul, menggunakan nilai indeks simpul. Rentang 0:3 menunjukkan simpul dengan nilai indeks dari 0 sampai 3. Jika nilai rentang awal dihilangkan (:n), maka 0 digunakan untuk memulai rentang. Jika nilai akhir rentang diabaikan (n:), indeks simpul tertinggi akan digunakan untuk mengakhiri rentang. Rentang simpul akumulatif Anda harus memperhitungkan semua simpul (0:n). Anda dapat membuat nest rentang simpul, misalnya 0:10 dan 4:5. Untuk kasus ini, properti rentang 4:5 akan menggantikan properti 0:10.

Tipe: String

Wajib: Tidak

`container`

Detail kontainer untuk rentang simpul. Untuk informasi selengkapnya, lihat [Properti kontainer](#).

Tipe: Objek [ContainerProperties](#)

Diperlukan: Tidak

Strategi percobaan kembali

`retryStrategy`

Ketika mendaftarkan ketentuan tugas, secara opsional Anda dapat menentukan strategi percobaan kembali yang akan digunakan untuk tugas gagal yang dikirimkan dengan ketentuan tugas ini. Strategi coba lagi apa pun yang ditentukan selama [SubmitJob](#) operasi mengesampingkan strategi coba lagi yang ditentukan di sini. Secara default, setiap tugas dicoba satu kali. Jika Anda menentukan lebih dari satu upaya, tugas akan dicoba lagi jika gagal. Contoh upaya yang gagal antara lain tugas yang mengembalikan kode keluar bukan nol atau instans kontainer yang diakhiri. Untuk informasi selengkapnya, lihat [Percobaan kembali tugas secara otomatis](#).

Tipe: Objek [RetryStrategy](#)

Diperlukan: Tidak

attempts

Jumlah upaya untuk memindahkan tugas ke status `RUNNABLE`. Anda dapat menentukan antara 1 sampai 10 upaya. Jika `attempts` lebih besar dari satu, tugas akan dicoba lagi sebanyak jumlah tersebut jika gagal, sampai tugas berpindah ke `RUNNABLE`.

```
"attempts": integer
```

Tipe: Integer

Wajib: Tidak

evaluateOnExit

Array hingga 5 objek yang menentukan kondisi di mana pekerjaan dicoba ulang atau gagal. Jika parameter ini ditentukan, maka `attempts` parameter juga harus ditentukan. Jika `evaluateOnExit` ditentukan tetapi tidak ada entri yang cocok, maka pekerjaan tersebut dicoba lagi.

```
"evaluateOnExit": [  
  {  
    "action": "string",  
    "onExitCode": "string",  
    "onReason": "string",  
    "onStatusReason": "string"  
  }  
]
```

Tipe: Array objek [EvaluateOnExit](#)

Diperlukan: Tidak

action

Menentukan tindakan yang akan diambil jika semua syarat yang ditentukan (`onStatusReason`, `onReason`, dan `onExitCode`) terpenuhi. Nilai ini tidak peka huruf besar dan kecil.

Tipe: String

Diperlukan: Ya

Nilai yang valid: `RETRY` | `EXIT`

`onExitCode`

Berisi pola glob untuk dicocokkan dengan representasi desimal dari `ExitCode` yang dikembalikan untuk tugas. Pola dapat memiliki panjang hingga 512 karakter. Ini hanya bisa berisi angka. Ini tidak dapat berisi huruf atau karakter khusus. Ini secara opsional dapat diakhiri dengan tanda bintang (*) sehingga hanya awal string yang harus sama persis.

Tipe: String

Wajib: Tidak

`onReason`

Berisi pola glob untuk dicocokkan dengan `Reason` yang dikembalikan untuk tugas. Pola dapat memiliki panjang hingga 512 karakter. Ini dapat berisi huruf, angka, titik (.), titik dua (:), dan spasi kosong (spasi, tab). Ini secara opsional dapat diakhiri dengan tanda bintang (*) sehingga hanya awal string yang harus sama persis.

Tipe: String

Wajib: Tidak

`onStatusReason`

Berisi pola glob untuk dicocokkan dengan `StatusReason` yang dikembalikan untuk tugas. Pola dapat memiliki panjang hingga 512 karakter. Ini dapat berisi huruf, angka, titik (.), titik dua (:), dan spasi kosong (spasi, tab). Ini secara opsional dapat diakhiri dengan tanda bintang (*) sehingga hanya awal string yang harus sama persis.

Tipe: String

Wajib: Tidak

Prioritas penjadwalan

`schedulingPriority`

Prioritas penjadwalan untuk pekerjaan yang diserahkan dengan definisi pekerjaan ini. Ini hanya memengaruhi pekerjaan dalam antrian pekerjaan dengan kebijakan pembagian yang adil. Pekerjaan dengan prioritas penjadwalan yang lebih tinggi dijadwalkan sebelum pekerjaan dengan prioritas penjadwalan yang lebih rendah.

Nilai minimum yang didukung adalah 0 dan nilai maksimum yang didukung adalah 9999.

Tipe: Integer

Wajib: Tidak

Tag

tags

Tanda pasangan kunci-nilai untuk dikaitkan dengan ketentuan tugas. Untuk informasi selengkapnya, lihat [Menandai sumber daya AWS Batch Anda](#).

Tipe: Peta antar string

Wajib: Tidak

Waktu habis

timeout

Anda dapat mengonfigurasi durasi batas waktu untuk pekerjaan Anda sehingga jika pekerjaan berjalan lebih lama dari itu, AWS Batch mengakhiri pekerjaan. Untuk informasi selengkapnya, lihat [Batas waktu Job](#). Jika pekerjaan dihentikan karena batas waktu, itu tidak dicoba lagi. Konfigurasi batas waktu apa pun yang ditentukan selama [SubmitJob](#) operasi akan mengganti konfigurasi batas waktu yang ditentukan di sini. Untuk informasi selengkapnya, lihat [Batas waktu Job](#).

Tipe: Objek [JobTimeout](#)

Diperlukan: Tidak

`attemptDurationSeconds`

Durasi waktu dalam detik (diukur dari stempel waktu `startedAt` upaya tugas) setelah AWS Batch mengakhiri tugas yang belum selesai. Nilai minimum untuk timeout adalah 60 detik.

Untuk pekerjaan array, batas waktu berlaku untuk pekerjaan anak, bukan untuk pekerjaan array induk.

Untuk pekerjaan multi-node parallel (MNP), batas waktu berlaku untuk seluruh pekerjaan, bukan untuk masing-masing node.

Tipe: Integer

Wajib: Tidak

Membuat definisi pekerjaan menggunakan EcsProperties

Dengan menggunakan definisi AWS Batch pekerjaan [EcsProperties](#), Anda dapat memodelkan perangkat keras, sensor, lingkungan 3D, dan simulasi lainnya dalam wadah terpisah. Anda dapat menggunakan fitur ini untuk secara logis mengatur komponen beban kerja Anda, dan memisahkannya dari aplikasi utama. Fitur ini dapat digunakan AWS Batch di Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS), dan AWS Fargate

ContainerProperties versus definisi EcsProperties pekerjaan

Anda dapat memilih untuk menggunakan [ContainerProperties](#) atau definisi [EcsProperties](#) pekerjaan sesuai dengan kasus penggunaan Anda. Pada tingkat tinggi, menjalankan AWS Batch pekerjaan dengan EcsProperties mirip dengan menjalankan pekerjaan dengan aContainerProperties.

Struktur definisi pekerjaan lama, menggunakan ContainerProperties, tetap didukung. Jika saat ini Anda memiliki alur kerja menggunakan struktur ini, Anda dapat terus menjalankannya.

Perbedaan utama adalah bahwa ada objek baru yang ditambahkan ke definisi pekerjaan untuk mengakomodasi definisi EcsProperties berbasis.

Misalnya, definisi pekerjaan yang digunakan ContainerProperties di Amazon ECS dan Fargate memiliki struktur berikut:

```
{
  "containerProperties": {
    ...
    "image": "my_ecr_image1",
    ...
  },
  ...
}
```

Definisi pekerjaan yang digunakan EcsProperties di Amazon ECS dan Fargate memiliki struktur berikut:

```
{
  "ecsProperties": {
    "taskProperties": [{
      "containers": [
        {
          ...
          "image": "my_ecr_image1",
          ...
        },
        {
          ...
          "image": "my_ecr_image2",
          ...
        },
      ],
    }
  ]
}
```

Perubahan umum pada AWS Batch API

Berikut ini lebih lanjut menguraikan beberapa perbedaan utama saat menggunakan `EcsProperties` dan tipe data `EcsProperties` API:

- Banyak parameter yang digunakan di dalamnya `ContainerProperties` muncul di dalam `TaskContainerProperties`. Beberapa contoh termasuk `command`, `image`, `privileged`, `secrets`, dan `users`. Mereka semua dapat ditemukan di dalam [TaskContainerProperties](#).
- Beberapa `TaskContainerProperties` parameter tidak memiliki ekuivalen fungsional dalam struktur lama. Beberapa contoh termasuk `dependsOn`, `essential`, `name`, `ipcMode`, dan `pidMode`. Untuk informasi lebih lanjut, lihat [EcsTaskDetails](#) dan [TaskContainerProperties](#).

Selain itu, beberapa `ContainerProperties` parameter tidak memiliki ekuivalen, atau aplikasi, dalam struktur. `EcsProperties` In [taskProperties](#), `container` telah diganti dengan `containers` sehingga objek baru dapat menerima hingga sepuluh elemen. [Untuk informasi lebih lanjut lihat: ContainerProperties dan: containersRegisterJobDefinition. EcsTaskProperties](#)

- `taskRoleArn` secara fungsional setara dengan `jobRoleArn`. Untuk informasi lebih lanjut lihat [EcsTaskProperties: taskRoleArn](#) dan [ContainerProperties: jobRoleArn](#).
- Anda dapat memasukkan dari satu (1) hingga sepuluh (10) wadah dalam `EcsProperties` struktur. [Untuk informasi lebih lanjut lihat: containerEcsTaskProperties](#).
- Objek `taskProperties` dan `InstanceTypes` adalah array, tetapi saat ini hanya menerima satu elemen. [Misalnya, ;TaskProperties dan: instanceTypesEcsProperties. NodeRangeProperty](#)

Definisi pekerjaan multi-kontainer untuk Amazon ECS

Untuk mengakomodasi struktur multi-container untuk Amazon ECS, beberapa tipe data API berbeda. Misalnya,

- [ecsProperties](#) adalah level yang sama seperti `containerProperties` dalam definisi kontainer tunggal. Untuk informasi selengkapnya, lihat [EcsProperties](#) di Panduan Referensi AWS Batch API.
- [taskProperties](#) berisi properti yang ditentukan untuk tugas Amazon ECS. Untuk informasi selengkapnya, lihat [EcsProperties](#) di Panduan Referensi AWS Batch API.
- [containers](#) mencakup informasi serupa dengan definisi kontainer tunggal. `containerProperties` Perbedaan utamanya adalah `containers` memungkinkan Anda menentukan hingga sepuluh kontainer. Untuk informasi selengkapnya, lihat [ECS:Containers TaskProperties](#) di [Panduan AWS Batch Referensi API](#).
- [essential](#) parameter menunjukkan bagaimana wadah mempengaruhi pekerjaan. Semua container essential harus menyelesaikan dengan sukses (keluar sebagai 0) agar pekerjaan dapat berkembang. Jika wadah yang ditandai sebagai essential gagal (keluar sebagai non-0), maka pekerjaan gagal.

Nilai default adalah `true` dan setidaknya satu kontainer harus ditandai sebagai `essential`. Untuk informasi selengkapnya, lihat [essential](#) di Panduan Referensi AWS Batch API.

- Dengan [dependsOn](#) parameter, Anda dapat menentukan daftar dependensi kontainer. Untuk informasi selengkapnya, lihat [dependsOn](#) di Panduan Referensi AWS Batch API.

Note

Kompleksitas `dependsOn` daftar dan runtime kontainer terkait dapat memengaruhi waktu mulai untuk pekerjaan Anda. Jika dependensi membutuhkan waktu lama untuk dijalankan, pekerjaan akan tetap dalam `STARTING` keadaan sampai selesai.

Untuk informasi selengkapnya tentang struktur `ecsProperties` dan, lihat sintaks [RegisterJobDefinition](#) permintaan untuk [ecsProperties](#).

Definisi pekerjaan multi-kontainer untuk Amazon EKS

Untuk mengakomodasi struktur multi-container untuk Amazon EKS, beberapa tipe data API berbeda. Misalnya,

- [name](#) adalah pengidentifikasi unik untuk wadah. Objek ini tidak diperlukan untuk satu kontainer, tetapi diperlukan saat mendefinisikan beberapa kontainer dalam sebuah pod. Ketika name tidak ditentukan untuk kontainer tunggal, nama default `default`, diterapkan.
- [initContainers](#) didefinisikan dalam tipe [eksPodProperties](#) data. They dijalankan sebelum wadah aplikasi, selalu berjalan hingga selesai, dan harus berhasil diselesaikan sebelum penampung berikutnya dimulai.

Kontainer ini terdaftar di agen Amazon EKS Connector dan menyimpan informasi pendaftaran di toko data backend Amazon Elastic Kubernetes Service. `initContainers` objek dapat menerima hingga sepuluh (10) elemen. Untuk informasi selengkapnya, lihat [Init Containers](#) dalam Kubernetes dokumentasi.

Note

`initContainers` objek dapat mempengaruhi waktu mulai untuk pekerjaan Anda. Jika `initContainers` butuh waktu lama untuk menjalankan, pekerjaan akan tetap dalam STARTING keadaan sampai mereka selesai.

- [shareProcessNamespace](#) menunjukkan apakah kontainer di pod dapat berbagi namespace proses yang sama. Nilai default adalah `false`. Menyetel ini `true` untuk mengaktifkan kontainer melihat dan memberi sinyal proses di wadah lain yang terletak di pod yang sama.
- Setiap wadah memiliki kepentingan. Semua kontainer harus berhasil diselesaikan (keluar sebagai 0) agar pekerjaan berhasil. Jika satu kontainer gagal (keluar sebagai selain 0), maka pekerjaan gagal.

Untuk informasi selengkapnya tentang struktur `eksProperties` dan, lihat sintaks [RegisterJobDefinition](#) permintaan untuk [eksProperties](#).

AWS Batch skenario pekerjaan menggunakan EcsProperties

Untuk mengilustrasikan bagaimana definisi AWS Batch pekerjaan yang digunakan `EcsProperties` dapat disusun berdasarkan kebutuhan Anda, topik ini menyajikan

[RegisterJobDefinition](#)muatan berikut. Anda dapat menyalin contoh-contoh ini ke dalam file, menyesuaikannya dengan kebutuhan Anda, dan kemudian menggunakan AWS Command Line Interface (AWS CLI) untuk memanggil `RegisterJobDefinition`.

AWS Batch pekerjaan untuk Amazon Elastic Container Service di Amazon Elastic Compute Cloud

```
{
  "jobDefinitionName": "multicontainer-ecs-ec2",
  "type": "container",
  "ecsProperties": {
    "taskProperties": [
      {
        "containers": [
          {
            "name": "c1",
            "essential": false,
            "command": [
              "echo",
              "hello world"
            ],
            "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
            "resourceRequirements": [
              {
                "type": "VCPU",
                "value": "2"
              },
              {
                "type": "MEMORY",
                "value": "4096"
              }
            ]
          },
          {
            "name": "c2",
            "essential": true,
            "command": [
              "echo",
              "hello world"
            ],
            "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
            "resourceRequirements": [
              {
```

```
        "type": "VCPU",
        "value": "6"
      },
      {
        "type": "MEMORY",
        "value": "12288"
      }
    ]
  }
}
}
```

AWS Batch pekerjaan untuk Amazon ECS di AWS Fargate

```
{
  "jobDefinitionName": "multicontainer-ecs-fargate",
  "type": "container",
  "platformCapabilities": [
    "FARGATE"
  ],
  "ecsProperties": {
    "taskProperties": [
      {
        "containers": [
          {
            "name": "c1",
            "command": [
              "echo",
              "hello world"
            ],
            "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
            "resourceRequirements": [
              {
                "type": "VCPU",
                "value": "2"
              },
              {
                "type": "MEMORY",
                "value": "4096"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

    ]
  },
  {
    "name": "c2",
    "essential": true,
    "command": [
      "echo",
      "hello world"
    ],
    "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
    "resourceRequirements": [
      {
        "type": "VCPU",
        "value": "6"
      },
      {
        "type": "MEMORY",
        "value": "12288"
      }
    ]
  }
],
"executionRoleArn": "arn:aws:iam::1112223333:role/ecsTaskExecutionRole"
}
]
}
}

```

AWS Batch lowongan kerja untuk Amazon Elastic Kubernetes Service

```

{
  "jobDefinitionName": "multicontainer-eks",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "shareProcessNamespace": true,
      "initContainers": [
        {
          "name": "init-container",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo"
          ],

```

```
    "args": [
      "hello world"
    ],
    "resources": {
      "requests": {
        "cpu": "1",
        "memory": "512Mi"
      }
    }
  },
  {
    "name": "init-container-2",
    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
    "command": [
      "echo",
      "my second init container"
    ],
    "resources": {
      "requests": {
        "cpu": "1",
        "memory": "512Mi"
      }
    }
  }
],
"containers": [
  {
    "name": "c1",
    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
    "command": [
      "echo world"
    ],
    "resources": {
      "requests": {
        "cpu": "1",
        "memory": "512Mi"
      }
    }
  },
  {
    "name": "sleep-container",
    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
    "command": [
      "sleep",
```

```
    "20"
  ],
  "resources": {
    "requests": {
      "cpu": "1",
      "memory": "512Mi"
    }
  }
}
]
```

AWS Batch Pekerjaan multi-node parallel (MNP) dengan beberapa kontainer per node

```
{
  "jobDefinitionName": "multicontainer-mnp",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 6,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:5",
        "ecsProperties": {
          "taskProperties": [
            {
              "containers": [
                {
                  "name": "range05-c1",
                  "command": [
                    "echo",
                    "hello world"
                  ],
                  "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                  "resourceRequirements": [
                    {
                      "type": "VCPU",
                      "value": "2"
                    },
                    {
                      "type": "MEMORY",
```

```
        "value": "4096"
      }
    ]
  },
  {
    "name": "range05-c2",
    "command": [
      "echo",
      "hello world"
    ],
    "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
    "resourceRequirements": [
      {
        "type": "VCPU",
        "value": "2"
      },
      {
        "type": "MEMORY",
        "value": "4096"
      }
    ]
  }
]
}
}
}
}
}
}
}
}
}
}
```

Menggunakan driver log awslogs

Secara default, AWS Batch memungkinkan driver `awslogs` log untuk mengirim informasi log ke CloudWatch Log. Anda dapat menggunakan fitur ini untuk melihat berbagai log dari kontainer Anda di satu lokasi yang nyaman dan mencegah log kontainer Anda mengambil ruang disk di instans kontainer Anda. Topik ini membantu Anda mengonfigurasi driver log `awslogs` dalam ketentuan tugas Anda.

Note

Di AWS Batch konsol, Anda dapat mengonfigurasi driver `awslogs` log di bagian konfigurasi Logging saat Anda membuat definisi pekerjaan.

Note

Jenis informasi yang dicatat oleh kontainer dalam tugas Anda sebagian besar tergantung pada perintah `ENTRYPOINT`. Secara default, log yang ditangkap menunjukkan output perintah yang biasanya Anda lihat di terminal interaktif jika Anda menjalankan kontainer secara lokal, yaitu pengaliran I/O `STDOUT` dan `STDERR`. Driver `awslogs` log hanya meneruskan log ini dari Docker ke CloudWatch Logs. Untuk informasi selengkapnya tentang cara log Docker diproses, termasuk cara alternatif untuk menangkap berbagai pengaliran atau data file, lihat [Melihat log untuk kontainer atau layanan](#) dalam dokumentasi Docker.

Untuk mengirim log sistem dari instance kontainer Anda ke CloudWatch Log, lihat [Menggunakan CloudWatch Log dengan AWS Batch](#). Untuk informasi selengkapnya tentang CloudWatch Log, lihat [Memantau Kuota File CloudWatch Log dan Log](#) di Panduan Pengguna CloudWatch Log Amazon.

Opsi driver log `awslogs` yang tersedia

Driver log `awslogs` mendukung opsi berikut dalam ketentuan tugas AWS Batch. Untuk informasi selengkapnya, lihat [Driver logging CloudWatch log](#) di dokumentasi Docker.

`awslogs-region`

Diperlukan: Tidak

Tentukan Wilayah tempat driver log `awslogs` harus mengirim log Docker Anda. Secara default, Wilayah yang digunakan adalah Wilayah yang sama dengan Wilayah untuk tugas. Anda dapat memilih untuk mengirim semua log Anda dari pekerjaan di Wilayah yang berbeda ke satu Wilayah di CloudWatch Log. Hal ini akan memungkinkan semuanya terlihat di satu lokasi. Sebagai alternatif, Anda dapat memisahkannya menurut Wilayah untuk pendekatan yang lebih terperinci. Namun, jika Anda memilih opsi ini, pastikan bahwa grup log yang ditentukan ada di Wilayah yang Anda tentukan.

awslogs-group

Wajib: Opsional

Dengan opsi `awslogs-group`, Anda dapat menentukan grup log yang menjadi tujuan pengiriman pengaliran log oleh driver log `awslogs`. Jika ini tidak ditentukan, `aws/batch/job` akan digunakan.

awslogs-stream-prefix

Wajib: Opsional

Dengan opsi `awslogs-stream-prefix` pilihan, Anda dapat mengaitkan pengaliran log dengan prefiks tertentu, dan ID tugas Amazon ECS dari tugas AWS Batch yang mencakup kontainer tersebut. Jika anda menentukan prefiks dengan pilihan ini, pengaliran log akan mengambil format berikut:

```
prefix-name/default/ecs-task-id
```

awslogs-datetime-format

Diperlukan: Tidak

Opsi ini menentukan pola permulaan multibaris dalam format `strftime` Python. Pesan log terdiri dari baris yang sesuai dengan pola dan baris berikutnya yang tidak cocok dengan pola. Dengan demikian, baris yang cocok adalah pembatas antarpesan log.

Salah satu contoh kasus penggunaan format ini adalah untuk penguraian output seperti buangan tumpukan, yang mungkin bisa tercatat dalam beberapa entri. Pola yang benar akan memungkinkannya ditangkap dalam satu entri.

Untuk informasi lebih lanjut, lihat [awslogs-datetime-format](#).

Opsi ini selalu diutamakan jika `awslogs-datetime-format` dan `awslogs-multiline-pattern` sama-sama dikonfigurasi.

Note

Pencatatan multibaris melakukan penguraian dan pencocokan ekspresi reguler dari semua pesan log. Ini dapat berdampak negatif pada performa pencatatan.

awslogs-multiline-pattern

Diperlukan: Tidak

Opsi ini menentukan pola permulaan multibaris dengan menggunakan ekspresi reguler. Pesan log terdiri dari baris yang sesuai dengan pola dan baris berikutnya yang tidak cocok dengan pola. Dengan demikian, baris yang cocok adalah pembatas antarpesan log.

Untuk informasi selengkapnya, lihat [awslogs-multiline-pattern](#) di dokumentasi Docker.

Opsi ini diabaikan jika `awslogs-datetime-format` juga dikonfigurasi.

Note

Pencatatan multibaris melakukan penguraian dan pencocokan ekspresi reguler dari semua pesan log. Ini mungkin berdampak negatif pada performa pencatatan.

awslogs-create-group

Diperlukan: Tidak

Menentukan apakah Anda ingin grup log dibuat secara otomatis. Jika opsi ini tidak ditentukan, default-nya adalah `false`.

Warning

Opsi ini tidak disarankan. Kami menyarankan Anda membuat grup log terlebih dahulu menggunakan tindakan CloudWatch Logs [CreateLogGroup](#) API karena setiap pekerjaan mencoba membuat grup log, meningkatkan kemungkinan pekerjaan gagal.

Note

Kebijakan IAM untuk peran eksekusi Anda harus menyertakan izin `logs:CreateLogGroup` sebelum Anda mencoba untuk menggunakan `awslogs-create-group`.

Menentukan konfigurasi log dalam ketentuan tugas Anda

Secara default, AWS Batch mengaktifkan driver log `awslogs`. Bagian ini menjelaskan cara menyesuaikan konfigurasi log `awslogs` untuk suatu tugas. Untuk informasi selengkapnya, lihat [Membuat definisi pekerjaan simpul tunggal](#).

Cuplikan JSON konfigurasi log berikut memiliki objek `logConfiguration` yang ditentukan untuk setiap tugas. Salah satunya adalah untuk WordPress pekerjaan yang mengirim log ke grup log yang dipanggil `awslogs-wordpress`, dan yang lainnya adalah untuk wadah MySQL yang mengirim log ke grup log yang dipanggil `awslogs-mysql`. Kedua kontainer menggunakan prefiks pengaliran log `awslogs-example`.

```
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {
    "awslogs-group": "awslogs-wordpress",
    "awslogs-stream-prefix": "awslogs-example"
  }
}
```

```
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {
    "awslogs-group": "awslogs-mysql",
    "awslogs-stream-prefix": "awslogs-example"
  }
}
```

Di konsol AWS Batch, konfigurasi log untuk ketentuan tugas `wordpress` ditentukan seperti yang ditunjukkan dalam gambar berikut.

Log configuration

Log driver

awslogs ▼

Options

Name	Value	
awslogs-group ▼	awslogs-wordpress	Remove option
awslogs-stream-prefix ▼	awslogs-example	Remove option

Add option

Secrets

Add secret

Setelah Anda mendaftarkan definisi tugas dengan driver `awslogs` log dalam konfigurasi log definisi pekerjaan, Anda dapat mengirimkan pekerjaan dengan definisi pekerjaan tersebut untuk mulai mengirim CloudWatch log ke Log. Untuk informasi selengkapnya, lihat [Mengirimkan pekerjaan](#).

Menentukan data sensitif

Dengan AWS Batch, Anda dapat menyuntikkan data sensitif ke dalam tugas Anda dengan menyimpan data sensitif Anda di rahasia AWS Secrets Manager atau parameter Parameter Store AWS Systems Manager, lalu mereferensikannya dalam ketentuan tugas Anda.

Rahasia dapat diungkap ke tugas dengan cara berikut:

- Untuk menyuntikkan data sensitif ke dalam kontainer Anda sebagai variabel lingkungan, gunakan parameter ketentuan tugas `secrets`.
- Untuk mereferensikan informasi sensitif dalam konfigurasi log tugas, gunakan parameter ketentuan tugas `secretOptions`.

Topik

- [Menentukan data sensitif menggunakan Secrets Manager](#)

- [Menentukan data sensitif menggunakan Systems Manager Parameter Store](#)

Menentukan data sensitif menggunakan Secrets Manager

Dengan AWS Batch, Anda dapat menyuntikkan data sensitif ke dalam pekerjaan Anda dengan menyimpan data sensitif Anda dalam AWS Secrets Manager rahasia dan kemudian mereferensikannya dalam definisi pekerjaan Anda. Data sensitif yang disimpan dalam rahasia Secrets Manager dapat diungkap ke tugas sebagai variabel lingkungan atau sebagai bagian dari konfigurasi log.

Ketika Anda menyuntikkan rahasia sebagai variabel lingkungan, Anda dapat menentukan kunci JSON atau versi rahasia yang akan disuntikkan. Proses ini membantu Anda mengontrol data sensitif yang diungkap ke tugas Anda. Untuk informasi selengkapnya tentang versioning rahasia, lihat [Konsep dan Istilah Penting untuk AWS Secrets Manager](#) dalam Panduan Pengguna AWS Secrets Manager .

Pertimbangan untuk menentukan data sensitif menggunakan Secrets Manager

Hal-hal berikut ini harus dipertimbangkan ketika menggunakan Secrets Manager untuk menentukan data sensitif untuk tugas.

- Untuk menyuntikkan rahasia menggunakan kunci JSON atau versi rahasia tertentu, instans kontainer di lingkungan komputasi Anda harus memiliki agen kontainer Amazon ECS. versi 1.37.0 atau yang lebih baru. Namun, sebaiknya gunakan versi agen kontainer terbaru. Untuk informasi tentang memeriksa versi agen Anda dan memperbarui ke versi terbaru, lihat [Memperbarui agen kontainer Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

Untuk menyuntikkan konten lengkap rahasia sebagai variabel lingkungan atau untuk menyuntikkan rahasia dalam konfigurasi log, instance container Anda harus memiliki versi 1.23.0 atau yang lebih baru dari agen kontainer.

- Hanya rahasia yang menyimpan data teks, yang merupakan rahasia yang dibuat dengan SecretString parameter [CreateSecret](#)API, yang didukung. Rahasia yang menyimpan data biner, yang merupakan rahasia yang dibuat dengan SecretBinary parameter [CreateSecret](#)API tidak didukung.
- Saat menggunakan ketentuan tugas yang mereferensikan rahasia Secrets Manager untuk mengambil data sensitif untuk tugas Anda, jika Anda juga menggunakan VPC endpoint antarmuka, Anda harus membuat VPC endpoint antarmuka untuk Secrets Manager. Untuk informasi

selengkapnya, lihat [Menggunakan Secrets Manager dengan VPC Endpoint](#) dalam Panduan Pengguna AWS Secrets Manager .

- Data sensitif disuntikkan ke tugas Anda saat tugas pertama kali dimulai. Jika rahasia kemudian diperbarui atau dirotasi, tugas tidak menerima nilai yang diperbarui secara otomatis. Anda harus meluncurkan tugas baru untuk memaksa layanan meluncurkan tugas baru dengan nilai rahasia yang diperbarui.

Izin IAM yang diperlukan untuk rahasia AWS Batch

Untuk menggunakan fitur ini, Anda harus memiliki peran eksekusi dan mereferensikannya dalam ketentuan tugas Anda. Hal ini memungkinkan agen kontainer untuk menarik sumber daya Secrets Manager yang diperlukan. Untuk informasi selengkapnya, lihat [AWS Batch eksekusi peran IAM](#).

Untuk menyediakan akses ke rahasia Secrets Manager yang Anda buat, tambahkan izin berikut secara manual ke peran eksekusi sebagai kebijakan inline. Untuk informasi lebih lanjut, lihat [Menambahkan dan Menghapus Kebijakan IAM](#) dalam Panduan Pengguna IAM.

- `secretsmanager:GetSecretValue`—Wajib jika Anda mereferensikan rahasia Secrets Manager.
- `kms:Decrypt`—Wajib hanya jika rahasia Anda menggunakan kunci KMS khusus dan bukan kunci default. ARN untuk kunci khusus Anda harus ditambahkan sebagai sumber daya.

Contoh kebijakan inline berikut menambahkan izin yang diperlukan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:<secret_name>",
        "arn:aws:kms:<region>:<aws_account_id>:key/<key_id>"
      ]
    }
  ]
}
```

Menyuntikkan data sensitif sebagai variabel lingkungan

Dalam ketentuan tugas, Anda dapat menentukan item berikut:

- Objek `secrets` yang berisi nama variabel lingkungan untuk diatur dalam tugas
- Amazon Resource Name (ARN) rahasia Secrets Manager
- Parameter tambahan yang berisi data sensitif untuk diberikan ke tugas

Contoh berikut menunjukkan sintaks lengkap yang harus ditentukan untuk rahasia Secrets Manager.

```
arn:aws:secretsmanager:region:aws_account_id:secret:secret-name:json-key:version-stage:version-id
```

Bagian berikut menjelaskan parameter tambahan. Parameter ini bersifat opsional. Namun, jika Anda tidak menggunakannya, Anda harus menyertakan titik dua : untuk menggunakan nilai default. Contoh disediakan di bawah ini sebagai konteks.

`json-key`

Menentukan nama kunci dalam pasangan kunci-nilai dengan nilai yang ingin Anda tetapkan sebagai nilai variabel lingkungan. Hanya nilai dalam format JSON yang didukung. Jika Anda tidak menentukan kunci JSON, konten lengkap dari rahasia akan digunakan.

`version-stage`

Menentukan label tahapan dari versi rahasia yang ingin Anda gunakan. Jika label tahapan versi ditentukan, Anda tidak dapat menentukan ID versi. Jika tidak ada tahap versi yang ditentukan, perilaku default-nya adalah mengambil rahasia dengan label tahapan `AWSCURRENT`.

Label tahapan digunakan untuk melacak berbagai versi rahasia saat diperbarui atau dirotasi. Setiap versi rahasia memiliki satu atau beberapa label tahapan dan satu ID. Untuk informasi selengkapnya, lihat [Persyaratan dan Konsep Utama untuk AWS Secrets Manager](#) di Panduan AWS Secrets Manager Pengguna.

`version-id`

Menentukan pengidentifikasi unik dari versi rahasia yang ingin Anda gunakan. Jika ID versi ditentukan, Anda tidak dapat menentukan label tahapan versi. Jika tidak ada ID versi yang ditentukan, perilaku default-nya adalah mengambil rahasia dengan label tahapan `AWSCURRENT`.

ID versi digunakan untuk melacak berbagai versi rahasia saat diperbarui atau dirotasi. Setiap versi rahasia memiliki satu ID. Untuk informasi selengkapnya, lihat [Persyaratan dan Konsep Utama untuk AWS Secrets Manager](#) di Panduan AWS Secrets Manager Pengguna.

Contoh ketentuan kontainer

Contoh berikut menunjukkan cara yang dapat Anda gunakan untuk mereferensikan rahasia Secrets Manager dalam ketentuan kontainer Anda.

Example mereferensikan rahasia penuh

Berikut ini adalah cuplikan ketentuan tugas yang menunjukkan format ketika mereferensikan teks lengkap dari rahasia Secrets Manager.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-AbCdEf"
    }]
  }]
}
```

Example mereferensikan kunci tertentu dalam sebuah rahasia

Berikut ini menunjukkan contoh output dari [get-secret-value](#) perintah yang menampilkan isi rahasia bersama dengan label pementasan versi dan ID versi yang terkait dengannya.

```
{
  "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
  "Name": "appauthexample",
  "VersionId": "871d9eca-18aa-46a9-8785-981dd39ab30c",
  "SecretString": "{\"username1\":\"password1\",\"username2\":\"password2\", \"username3\":\"password3\"}",
  "VersionStages": [
    "AWSCURRENT"
  ],
  "CreateDate": 1581968848.921
}
```

Referensi kunci tertentu dari output sebelumnya dalam ketentuan kontainer dengan menentukan nama kunci di akhir ARN.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf:username1:~"
    }]
  }]
}
```

Example mereferensikan versi rahasia tertentu

Berikut ini menunjukkan contoh output dari perintah [describe-secret](#) yang menampilkan konten rahasia yang tidak terenkripsi bersama dengan metadata untuk semua versi rahasia.

```
{
  "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
  "Name": "appauthexample",
  "Description": "Example of a secret containing application authorization data.",
  "RotationEnabled": false,
  "LastChangedDate": 1581968848.926,
  "LastAccessedDate": 1581897600.0,
  "Tags": [],
  "VersionIdsToStages": {
    "871d9eca-18aa-46a9-8785-981dd39ab30c": [
      "AWSCURRENT"
    ],
    "9d4cb84b-ad69-40c0-a0ab-cead36b967e8": [
      "AWSPREVIOUS"
    ]
  }
}
```

Referensi label tahapan versi tertentu dari output sebelumnya dalam ketentuan kontainer dengan menentukan nama kunci di akhir ARN.

```
{
  "containerProperties": [{
```

```

"secrets": [{
  "name": "environment_variable_name",
  "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf::AWSPREVIOUS:"
}]
}]
}

```

Referensi ID versi tertentu dari output sebelumnya dalam ketentuan kontainer dengan menentukan nama kunci di akhir ARN.

```

{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}

```

Example mereferensikan kunci dan label tahapan versi tertentu dari suatu rahasia

Bagian berikut menunjukkan cara mereferensikan kunci tertentu dalam rahasia dan label tahapan versi tertentu.

```

{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1:AWSPREVIOUS:"
    }]
  }]
}

```

Untuk menentukan kunci dan versi ID tertentu, gunakan sintaks berikut.

```

{
  "containerProperties": [{
    "secrets": [{

```

```

    "name": "environment_variable_name",
    "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
  }]
}]
}

```

Menyuntikkan data sensitif dalam konfigurasi log

Dalam ketentuan tugas Anda, ketika menentukan `logConfiguration` Anda dapat menentukan `secretOptions` dengan nama opsi driver log yang akan ditetapkan dalam kontainer dan ARN lengkap dari rahasia Secrets Manager yang berisi data sensitif untuk diberikan ke kontainer.

Berikut ini adalah cuplikan ketentuan tugas yang menunjukkan format ketika mereferensikan rahasia Secrets Manager.

```

{
  "containerProperties": [{
    "logConfiguration": [{
      "logDriver": "splunk",
      "options": {
        "splunk-url": "https://cloud.splunk.com:8080"
      },
      "secretOptions": [{
        "name": "splunk-token",
        "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-
AbCdEf"
      }]
    }]
  }]
}

```

Menciptakan AWS Secrets Manager rahasia

Anda dapat menggunakan konsol Secrets Manager untuk membuat rahasia untuk data sensitif Anda. Untuk informasi lebih lanjut, lihat [Membuat Rahasia Dasar](#) dalam Panduan Pengguna AWS Secrets Manager .

Untuk membuat rahasia dasar

Gunakan Secrets Manager untuk membuat rahasia untuk data sensitif Anda.

1. Buka konsol Secrets Manager di <https://console.aws.amazon.com/secretsmanager/>.
2. Pilih Store a new secret (Simpan rahasia baru).
3. Untuk Select secret type (Pilih tipe rahasia), pilih Other type of secrets (Tipe rahasia lainnya).
4. Tentukan detail rahasia khusus Anda sebagai pasangan Key (Kunci) dan Value (Nilai). Misalnya, Anda dapat menentukan kunci Username, lalu memberikan nama pengguna yang sesuai sebagai nilainya. Tambahkan kunci kedua dengan nama Password dan teks kata sandi sebagai nilainya. Anda juga dapat menambahkan entri untuk nama basis data, alamat server, atau port TCP. Anda dapat menambahkan pasangan sebanyak yang Anda butuhkan untuk menyimpan informasi yang Anda butuhkan.

Sebagai alternatif, Anda dapat memilih tab Plaintext dan masukkan nilai rahasia dengan cara apa pun yang Anda sukai.

5. Pilih kunci AWS KMS enkripsi yang ingin Anda gunakan untuk mengenkripsi teks yang dilindungi secara rahasia. Jika Anda tidak memilihnya, Secrets Manager akan memeriksa apakah ada kunci default untuk akun tersebut, dan menggunakannya jika ada. Jika kunci default tidak ada, Secrets Manager akan membuatnya untuk Anda secara otomatis. Anda juga dapat memilih Tambahkan kunci baru untuk membuat kunci KMS khusus untuk rahasia ini. Untuk membuat kunci KMS Anda sendiri, Anda harus memiliki izin untuk membuat kunci KMS di akun Anda.
6. Pilih Selanjutnya.
7. Untuk Secret name (Nama rahasia), ketik nama dan jalur opsional, misalnya **production/MyAwesomeAppSecret** atau **development/TestSecret**, lalu pilih Next (Berikutnya). Anda dapat menambahkan deskripsi secara opsional untuk membantu Anda mengingat tujuan dari rahasia ini nanti.

Nama rahasia harus merupakan huruf, digit, atau salah satu karakter ASCII berikut: /_+ =.@-

8. (Opsional) Di sini, Anda dapat mengonfigurasi rotasi untuk rahasia Anda. Untuk prosedur ini, biarkan tetap Disable automatic rotation (Nonaktifkan rotasi otomatis) dan pilih Next (Berikutnya).

Untuk informasi tentang cara mengonfigurasi rotasi pada rahasia baru atau yang sudah ada, lihat [Memutar AWS Secrets Manager Rahasia Anda](#).

9. Tinjau pengaturan Anda, lalu pilih Store secret (Simpan rahasia) untuk menyimpan semua yang Anda masukkan sebagai rahasia baru di Secrets Manager.

Menentukan data sensitif menggunakan Systems Manager Parameter Store

Dengan AWS Batch, Anda dapat menyuntikkan data sensitif ke dalam wadah Anda dengan menyimpan data sensitif Anda dalam AWS Systems Manager parameter Parameter Store dan kemudian mereferensikannya dalam definisi kontainer Anda.

Topik

- [Pertimbangan untuk menentukan data sensitif menggunakan Systems Manager Parameter Store](#)
- [Izin IAM yang diperlukan untuk rahasia AWS Batch](#)
- [Menyuntikkan data sensitif sebagai variabel lingkungan](#)
- [Menyuntikkan data sensitif dalam konfigurasi log](#)
- [Membuat AWS Systems Manager parameter Parameter Store](#)

Pertimbangan untuk menentukan data sensitif menggunakan Systems Manager Parameter Store

Hal-hal berikut harus dipertimbangkan ketika menentukan data sensitif untuk kontainer menggunakan parameter Systems Manager Parameter Store.

- Fitur ini mengharuskan instance container Anda memiliki versi 1.23.0 atau yang lebih baru dari agen kontainer. Akan tetapi, kami merekomendasikan untuk menggunakan versi agen kontainer terbaru. Untuk informasi tentang memeriksa versi agen Anda dan memperbarui ke versi terbaru, lihat [Memperbarui agen kontainer Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.
- Data sensitif disuntikkan ke dalam kontainer untuk tugas Anda ketika kontainer pertama kali dimulai. Jika rahasia atau parameter Parameter Store kemudian diperbarui atau dirotasi, kontainer tidak menerima nilai yang diperbarui secara otomatis. Anda harus meluncurkan tugas baru untuk memaksa peluncuran tugas baru dengan rahasia yang diperbarui.

Izin IAM yang diperlukan untuk rahasia AWS Batch

Untuk menggunakan fitur ini, Anda harus memiliki peran eksekusi dan mereferensikannya dalam ketentuan tugas Anda. Ini memungkinkan agen kontainer Amazon ECS untuk menarik AWS Systems Manager sumber daya yang diperlukan. Untuk informasi selengkapnya, lihat [AWS Batch eksekusi peran IAM](#).

Untuk memberikan akses ke AWS Systems Manager parameter Parameter Store yang Anda buat, tambahkan izin berikut secara manual sebagai kebijakan inline ke peran eksekusi. Untuk informasi lebih lanjut, lihat [Menambahkan dan Menghapus Kebijakan IAM](#) dalam Panduan Pengguna IAM.

- `ssm:GetParameters`—Wajib jika Anda mereferensikan parameter Systems Manager Parameter Store dalam ketentuan tugas.
- `secretsmanager:GetSecretValue`—Wajib jika Anda mereferensikan rahasia Secrets Manager secara langsung atau jika parameter Systems Manager Parameter Store Anda mereferensikan rahasia Secrets Manager dalam ketentuan tugas.
- `kms:Decrypt`—Wajib hanya jika rahasia Anda menggunakan kunci KMS khusus dan bukan kunci default. ARN untuk kunci khusus Anda harus ditambahkan sebagai sumber daya.

Contoh kebijakan inline berikut menambahkan izin yang diperlukan:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:<region>:<aws_account_id>:parameter/<parameter_name>",
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:<secret_name>",
        "arn:aws:kms:<region>:<aws_account_id>:key/<key_id>"
      ]
    }
  ]
}
```

Menyuntikkan data sensitif sebagai variabel lingkungan

Dalam ketentuan kontainer Anda, tentukan `secrets` dengan nama variabel lingkungan yang akan ditetapkan dalam kontainer dan ARN lengkap parameter Systems Manager Parameter Store yang berisi data sensitif untuk diberikan ke kontainer.

Berikut ini adalah cuplikan ketentuan tugas yang menunjukkan format ketika mereferensikan parameter Systems Manager Parameter Store. Jika parameter Systems Manager Parameter Store ada di wilayah yang sama dengan tugas yang Anda luncurkan, Anda dapat menggunakan ARN lengkap atau nama parameter. Jika parameter ada di Wilayah yang berbeda, ARN lengkap harus disebutkan.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }]
  }]
}
```

Menyuntikkan data sensitif dalam konfigurasi log

Dalam ketentuan kontainer Anda, ketika menentukan `logConfiguration`, Anda dapat menentukan `secretOptions` dengan nama opsi driver log yang kan ditetapkan dalam kontainer dan ARN lengkap parameter Systems Manager Parameter Store yang berisi data sensitif untuk diberikan ke kontainer.

Important

Jika parameter Systems Manager Parameter Store berada di wilayah yang sama dengan tugas yang Anda luncurkan, Anda dapat menggunakan ARN lengkap atau nama parameter. Jika parameter ada di Wilayah yang berbeda, ARN lengkap harus disebutkan.

Berikut ini adalah cuplikan ketentuan tugas yang menunjukkan format ketika mereferensikan parameter Systems Manager Parameter Store.

```
{
  "containerProperties": [{
    "logConfiguration": [{
      "logDriver": "fluentd",
      "options": {
        "tag": "fluentd demo"
      },
      "secretOptions": [{
```



```

    "name": "fluentd-address",
    "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
  }}
}}
}}
}
```

Membuat AWS Systems Manager parameter Parameter Store

Anda dapat menggunakan AWS Systems Manager konsol untuk membuat parameter Systems Manager Parameter Store untuk data sensitif Anda. Untuk informasi lebih lanjut, lihat [Panduan: Membuat dan menggunakan Parameter dalam Perintah \(konsol\)](#) dalam Panduan Pengguna AWS Systems Manager .

Untuk membuat parameter Parameter Store

1. Buka AWS Systems Manager konsol di <https://console.aws.amazon.com/systems-manager/>.
2. Di panel navigasi, pilih Parameter Store, Create parameter (Buat parameter).
3. Untuk Name (Nama), ketik hierarki dan nama parameter. Sebagai contoh, ketik `test/database_password`.
4. Untuk Description (Deskripsi), ketik deskripsi opsional.
5. Untuk Type, pilih String, StringList, atau SecureString.

Note

- Jika Anda memilih SecureString, bidang ID Kunci KMS akan muncul. Jika Anda tidak memberikan ID kunci KMS, ARN kunci KMS, nama alias, atau alias ARN, maka sistem menggunakan `alias/aws/ssm` Ini adalah kunci KMS default untuk Systems Manager. Untuk menghindari penggunaan kunci ini, pilih tombol kustom. Untuk informasi lebih lanjut, lihat [Menggunakan Parameter String Aman](#) dalam Panduan Pengguna AWS Systems Manager .
- Saat Anda membuat parameter string aman di konsol dengan menggunakan `key-id` parameter dengan nama alias kunci KMS kustom atau alias ARN, Anda harus menentukan awalan sebelum alias. `alias/` Berikut ini adalah contoh ARN:

```
arn:aws:kms:us-east-2:123456789012:alias/MyAliasName
```

Berikut ini adalah contoh nama alias:

```
alias/MyAliasName
```

6. Untuk Value (Nilai), ketik nilai. Misalnya, `MyFirstParameter`. Jika Anda memilih `SecureString`, nilainya ditutupi persis seperti yang Anda masukkan.
7. Pilih `Create parameter (Buat parameter)`.

Otentikasi registri pribadi untuk pekerjaan

Otentikasi registri pribadi untuk pekerjaan yang menggunakan AWS Secrets Manager memungkinkan Anda untuk menyimpan kredensi Anda dengan aman dan kemudian mereferensikannya dalam definisi pekerjaan Anda. Ini menyediakan cara untuk mereferensikan gambar kontainer yang ada di pendaftar pribadi di luar AWS yang memerlukan otentikasi dalam definisi pekerjaan Anda. Fitur ini didukung oleh pekerjaan yang dihosting di instans Amazon EC2 dan Fargate.

Important

Jika definisi pekerjaan Anda mereferensikan gambar yang disimpan di Amazon ECR, topik ini tidak berlaku. Untuk informasi selengkapnya, lihat [Menggunakan Gambar Amazon ECR dengan Amazon ECS](#) di Panduan Pengguna Amazon Elastic Container Registry.

Untuk pekerjaan yang dihosting di instans Amazon EC2, fitur ini memerlukan versi `1.19.0` atau yang lebih baru dari agen penampung. Akan tetapi, kami merekomendasikan untuk menggunakan versi agen kontainer terbaru. Untuk informasi tentang cara memeriksa versi agen Anda dan memperbarui ke versi terbaru, lihat [Memperbarui agen penampung Amazon ECS](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.

Untuk pekerjaan yang dihosting di Fargate, fitur ini memerlukan versi platform `1.2.0` atau yang lebih baru. Untuk selengkapnya, lihat [versi platform AWS Fargate Linux](#) di Panduan Pengembang Layanan Amazon Elastic Container.

Dalam definisi kontainer Anda, tentukan `repositoryCredentials` objek dengan detail rahasia yang Anda buat. Rahasia yang Anda referensikan bisa dari akun yang berbeda Wilayah AWS atau berbeda dari pekerjaan yang menggunakannya.

Note

Saat menggunakan AWS Batch API, AWS CLI, atau AWS SDK, jika rahasia ada Wilayah AWS sama dengan pekerjaan yang Anda luncurkan maka Anda dapat menggunakan ARN lengkap atau nama rahasia. Jika rahasia ada di akun yang berbeda, ARN penuh rahasia harus ditentukan. Saat menggunakan AWS Management Console, ARN penuh rahasia harus ditentukan selalu.

Berikut ini adalah cuplikan definisi pekerjaan yang menunjukkan parameter yang diperlukan:

```
"containerProperties": [  
  {  
    "image": "private-repo/private-image",  
    "repositoryCredentials": {  
      "credentialsParameter":  
        "arn:aws:secretsmanager:region:123456789012:secret:secret_name"  
    }  
  }  
]
```

Izin IAM yang diperlukan untuk otentikasi registri pribadi

Peran eksekusi diperlukan untuk menggunakan fitur ini. Hal ini mengizinkan agen kontainer untuk menarik citra kontainer. Untuk informasi selengkapnya, lihat [AWS Batch eksekusi peran IAM](#).

Untuk memberikan akses ke rahasia yang Anda buat, tambahkan izin berikut sebagai kebijakan inline ke peran eksekusi. Untuk informasi selengkapnya, lihat [Menambahkan dan Menghapus Kebijakan IAM](#).

- `secretsmanager:GetSecretValue`
- `kms:Decrypt`—Diperlukan hanya jika kunci Anda menggunakan kunci KMS khusus dan bukan kunci default. Nama Sumber Daya Amazon (ARN) untuk kunci kustom Anda harus ditambahkan sebagai sumber daya.

Berikut ini adalah contoh kebijakan inline yang menambahkan izin.

```
{  
  "Version": "2012-10-17",
```

```

"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "kms:Decrypt",
      "secretsmanager:GetSecretValue"
    ],
    "Resource": [
      "arn:aws:secretsmanager:region:123456789012:secret:secret_name",
      "arn:aws:kms:region:123456789012:key/key_id"
    ]
  }
]
}

```

Menggunakan otentikasi registri pribadi

Untuk membuat rahasia dasar

Gunakan AWS Secrets Manager untuk membuat rahasia untuk kredensial registri pribadi Anda.

1. Buka AWS Secrets Manager konsol di <https://console.aws.amazon.com/secretsmanager/>.
2. Pilih Store a new secret (Simpan rahasia baru).
3. Untuk Pilih tipe rahasia, pilih Jenis rahasia lainnya.
4. Pilih Plaintext dan masukkan kredensial registri privat Anda menggunakan format berikut:

```

{
  "username" : "privateRegistryUsername",
  "password" : "privateRegistryPassword"
}

```

5. Pilih Selanjutnya.
6. Untuk nama Rahasia, masukkan jalur dan nama opsional, seperti **production/MyAwesomeAppSecret** atau **development/TestSecret**, dan pilih Berikutnya. Anda dapat menambahkan deskripsi secara opsional untuk membantu Anda mengingat tujuan dari rahasia ini nanti.

Nama rahasia harus huruf ASCII, digit, atau salah satu karakter berikut: `/_+=.@-`

7. (Opsional) Di sini, Anda dapat mengonfigurasi rotasi untuk rahasia Anda. Untuk prosedur ini, biarkan tetap Disable automatic rotation (Nonaktifkan rotasi otomatis) dan pilih Next (Berikutnya).

Untuk petunjuk tentang cara mengonfigurasi rotasi pada rahasia baru atau yang sudah ada, lihat [Memutar AWS Secrets Manager Rahasia Anda](#).

8. Tinjau pengaturan Anda, lalu pilih Simpan rahasia untuk menyimpan semua yang Anda masukkan sebagai rahasia baru di Secrets Manager.

Daftarkan definisi pekerjaan dan di bawah Registri pribadi, aktifkan Otentikasi registri pribadi. Kemudian, di Secrets Manager ARN atau nama, masukkan Amazon Resource Name (ARN) dari rahasia tersebut. Lihat informasi yang lebih lengkap di [Izin IAM yang diperlukan untuk otentikasi registri pribadi](#).

Volume Amazon EFS

Amazon Elastic File System (Amazon EFS) menyediakan penyimpanan file sederhana yang dapat diskalakan untuk digunakan dengan tugas AWS Batch Anda. Kapasitas penyimpanan Amazon EFS bersifat elastis. Penyimpanan akan menskalakan secara otomatis saat Anda menambahkan dan menghapus file. Aplikasi Anda dapat memiliki penyimpanan yang dibutuhkan, kapan pun dibutuhkan.

Anda dapat menggunakan sistem file Amazon EFS dengan AWS Batch untuk mengekspor data sistem file di seluruh armada instans kontainer Anda. Dengan begitu, tugas Anda memiliki akses ke penyimpanan tetap yang sama. Namun, Anda harus mengonfigurasi AMI instans kontainer untuk memasang sistem file Amazon EFS sebelum daemon Docker dimulai. Selain itu, ketentuan tugas Anda harus mereferensikan pemasangan volume di instans kontainer agar dapat menggunakan sistem file. Bagian berikut akan membantu Anda mulai menggunakan Amazon EFS dengan AWS Batch.

Pertimbangan volume Amazon EFS

Hal-hal berikut harus dipertimbangkan ketika menggunakan volume Amazon EFS:

- Untuk tugas yang menggunakan sumber daya EC2, dukungan sistem file Amazon EFS ditambahkan sebagai pratinjau publik dengan AMI yang dioptimalkan untuk Amazon ECS versi 20191212 dengan agen kontainer versi 1.35.0. Namun, dukungan sistem file Amazon EFS memasuki ketersediaan umum dengan AMI yang dioptimalkan untuk Amazon ECS versi 20200319 dengan agen kontainer versi 1.38.0, yang berisi titik akses Amazon EFS dan fitur otorisasi IAM. Kami menyarankan agar Anda menggunakan AMI yang dioptimalkan untuk Amazon ECS versi 20200319 atau yang lebih baru untuk memanfaatkan fitur ini. Untuk informasi lebih

lanjut, lihat [Versi AMI yang dioptimalkan untuk Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

Note

Jika Anda membuat AMI sendiri, Anda harus menggunakan agen kontainer 1.38.0 atau yang lebih baru, `ecs-init` versi 1.38.0-1 atau yang lebih baru, dan jalankan perintah berikut di instans Amazon EC2 Anda. Ini semua untuk mengaktifkan plugin volume Amazon ECS. Perintah akan tergantung pada apakah Anda menggunakan Amazon Linux 2 atau Amazon Linux sebagai citra dasar Anda.

Amazon Linux 2

```
$ yum install amazon-efs-utils
systemctl enable --now amazon-ecs-volume-plugin
```

Amazon Linux

```
$ yum install amazon-efs-utils
sudo shutdown -r now
```

- Untuk tugas yang menggunakan sumber daya Fargate, dukungan sistem file Amazon EFS ditambahkan saat menggunakan platform versi 1.4.0 atau yang lebih baru. Untuk informasi lebih lanjut, lihat [Versi platform AWS Fargate](#) dalam Panduan Developer Amazon Elastic Container Service.
- Ketika menentukan volume Amazon EFS dalam tugas yang menggunakan sumber daya Fargate, Fargate membuat kontainer pengawas yang bertanggung jawab untuk mengelola volume Amazon EFS. Kontainer pengawas menggunakan sejumlah kecil memori tugas. Kontainer pengawas akan terlihat saat dilakukan kueri titik akhir metadata tugas versi 4. Untuk informasi selengkapnya, lihat [Titik akhir metadata tugas versi 4](#) di Panduan Pengguna Amazon Elastic Container Service untuk AWS Fargate

Menggunakan titik akses Amazon EFS

Titik akses Amazon EFS adalah titik masuk khusus aplikasi ke dalam sistem file EFS yang membantu Anda mengelola akses aplikasi ke set data bersama. Untuk informasi lebih lanjut tentang titik akses

Amazon EFS dan cara mengontrol akses ke sana, lihat [Bekerja dengan Titik Akses Amazon EFS](#) dalam Panduan Pengguna Amazon Elastic File System.

Titik akses dapat menerapkan identitas pengguna, termasuk grup POSIX pengguna, pada semua permintaan sistem file yang dibuat melalui titik akses. Titik akses juga dapat menerapkan direktori asal yang berbeda untuk sistem file sehingga klien hanya dapat mengakses data dalam direktori tertentu atau subdirektornya.

Note

Saat membuat titik akses EFS, Anda menentukan jalur di sistem file yang akan berfungsi sebagai direktori asal. Ketika Anda mereferensikan sistem file EFS dengan ID titik akses di ketentuan tugas AWS Batch Anda, direktori asal harus dihilangkan atau diatur ke /. Ini akan memberlakukan jalur yang ditetapkan pada titik akses EFS.

Anda dapat menggunakan IAM role tugas AWS Batch untuk memberlakukan penggunaan aplikasi tertentu atas titik akses tertentu. Dengan menggabungkan kebijakan IAM dengan titik akses, Anda dapat dengan mudah memberikan akses yang aman ke set data tertentu untuk aplikasi Anda. Fitur ini menggunakan IAM role Amazon ECS untuk fungsionalitas tugas. Untuk informasi lebih lanjut, lihat [IAM Role untuk Tugas](#) dalam Panduan Developer Amazon Elastic Container Service.

Menentukan sistem file Amazon EFS dalam ketentuan tugas Anda

Untuk menggunakan volume sistem file Amazon EFS untuk kontainer Anda, Anda harus menentukan konfigurasi volume dan titik pemasangan dalam ketentuan tugas Anda. Cuplikan JSON ketentuan tugas berikut menunjukkan sintaks untuk objek `volumes` dan `mountPoints` untuk sebuah kontainer:

```
{
  "containerProperties": [
    {
      "image": "amazonlinux:2",
      "command": [
        "ls",
        "-la",
        "/mount/efs"
      ],
      "mountPoints": [
        {
          "sourceVolume": "myEfsVolume",
```

```
        "containerPath": "/mount/efs",
        "readOnly": true
    }
],
"volumes": [
    {
        "name": "myEfsVolume",
        "efsVolumeConfiguration": {
            "fileSystemId": "fs-12345678",
            "rootDirectory": "/path/to/my/data",
            "transitEncryption": "ENABLED",
            "transitEncryptionPort": integer,
            "authorizationConfig": {
                "accessPointId": "fsap-1234567890abcdef1",
                "iam": "ENABLED"
            }
        }
    }
]
}
}
```

efsVolumeConfiguration

Tipe: Objek

Diperlukan: Tidak

Parameter ini ditentukan saat menggunakan volume Amazon EFS.

fileSystemId

Tipe: String

Diperlukan: Ya


ID sistem file Amazon EFS yang akan digunakan.

rootDirectory

Tipe: String

Wajib: Tidak

Direktori dalam sistem file Amazon EFS untuk pemasangan sebagai direktori asal di dalam host. Jika parameter ini diabaikan, asal volume Amazon EFS akan digunakan. Menentukan / memiliki efek yang sama seperti mengabaikan parameter ini. Ini dapat memiliki panjang hingga 4.096 karakter.

 Important

Jika titik akses EFS ditentukan dalam `authorizationConfig`, parameter direktori asal harus diabaikan atau diatur ke `/`. Ini akan memberlakukan jalur yang ditetapkan pada titik akses EFS.

`transitEncryption`

Tipe: String

Nilai yang valid: ENABLED | DISABLED

Diperlukan: Tidak

Menentukan apakah akan mengaktifkan enkripsi untuk data Amazon EFS yang transit di antara host AWS Batch dan server Amazon EFS. Enkripsi transit harus diaktifkan jika otorisasi IAM Amazon EFS digunakan. Jika parameter ini diabaikan, nilai default DISABLED akan digunakan. Untuk informasi lebih lanjut, lihat [Mengkripsi data saat transit](#) dalam Panduan Pengguna Amazon Elastic File System.

`transitEncryptionPort`

Tipe: Integer

Wajib: Tidak

Port yang akan digunakan saat mengirim data terenkripsi antara host AWS Batch dan server Amazon EFS. Jika Anda tidak menentukan port enkripsi transit, strategi pemilihan port yang digunakan oleh pembantu pemasangan Amazon EFS akan digunakan. Nilai harus antara 0 dan 65.535. Untuk informasi lebih lanjut, lihat [Pembantu Pemasangan EFS](#) dalam Panduan Pengguna Amazon Elastic File System.

`authorizationConfig`

Tipe: Objek

Diperlukan: Tidak

Detail konfigurasi otorisasi untuk sistem file Amazon EFS.

`accessPointId`

Tipe: String

Wajib: Tidak

ID titik akses yang akan digunakan. Jika jalur akses ditentukan, nilai direktori asal di `efsVolumeConfiguration` harus diabaikan atau diatur ke `/`. Ini akan memberlakukan jalur yang ditetapkan pada titik akses EFS. Jika titik akses digunakan, enkripsi transit harus diaktifkan di `EFSSVolumeConfiguration`. Untuk informasi lebih lanjut, lihat [Bekerja dengan Titik Akses Amazon EFS](#) dalam Panduan Pengguna Amazon Elastic File System.

`iam`

Tipe: String

Nilai yang valid: ENABLED | DISABLED

Diperlukan: Tidak

Menentukan apakah akan menggunakan IAM role tugas AWS Batch yang didefinisikan dalam ketentuan tugas ketika memasang sistem file Amazon EFS. Jika diaktifkan, enkripsi transit harus diaktifkan di `EFSSVolumeConfiguration`. Jika parameter ini diabaikan, nilai default DISABLED akan digunakan. Untuk informasi lebih lanjut tentang IAM role eksekusi, lihat [AWS Batch eksekusi peran IAM](#).

Contoh ketentuan tugas

Contoh ketentuan tugas berikut menggambarkan cara menggunakan pola umum, seperti variabel lingkungan, substitusi parameter, dan pemasangan volume.

Menggunakan variabel lingkungan

Contoh ketentuan tugas berikut menggunakan variabel lingkungan untuk menentukan jenis file dan URL Amazon S3. Contoh ini diambil dari postingan blog komputasi [Membuat Tugas AWS Batch "Fetch & Run" Sederhana](#). Skrip `fetch_and_run.sh` yang dijelaskan di postingan blog tersebut menggunakan variabel lingkungan ini untuk mengunduh skrip `myjob.sh` dari S3 dan menyatakan jenis file-nya.

Meskipun perintah dan lingkungan variabel bersifat hardcoded dalam ketentuan tugas di contoh ini, Anda dapat menentukan penempatan perintah dan variabel lingkungan untuk membuat ketentuan tugas menjadi lebih fleksibel.

```
{
  "jobDefinitionName": "fetch_and_run",
  "type": "container",
  "containerProperties": {
    "image": "123456789012.dkr.ecr.us-east-1.amazonaws.com/fetch_and_run",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "2000"
      },
      {
        "type": "VCPU",
        "value": "2"
      }
    ],
    "command": [
      "myjob.sh",
      "60"
    ],
    "jobRoleArn": "arn:aws:iam::123456789012:role/AWSBatchS3ReadOnly",
    "environment": [
      {
        "name": "BATCH_FILE_S3_URL",
        "value": "s3://my-batch-scripts/myjob.sh"
      },
      {
        "name": "BATCH_FILE_TYPE",
        "value": "script"
      }
    ],
    "user": "nobody"
  }
}
```

Menggunakan substitusi parameter

Contoh ketentuan tugas berikut menggambarkan cara untuk memungkinkan substitusi parameter dan mengatur nilai default.

Deklarasi `Ref` :: di bagian `command` digunakan untuk mengatur placeholder untuk substitusi parameter. Ketika Anda mengirim tugas dengan ketentuan tugas ini, Anda menentukan penyimpanan parameter untuk mengisi nilai-nilai tersebut, seperti `inputfile` dan `outputfile`. Bagian `parameters` yang mengikutinya akan menetapkan default untuk codec, tetapi Anda dapat mengganti parameter tersebut sesuai kebutuhan.

Untuk informasi selengkapnya, lihat [Parameter-parameter](#).

```
{
  "jobDefinitionName": "ffmpeg_parameters",
  "type": "container",
  "parameters": {"codec": "mp4"},
  "containerProperties": {
    "image": "my_repo/ffmpeg",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "2000"
      },
      {
        "type": "VCPU",
        "value": "2"
      }
    ],
    "command": [
      "ffmpeg",
      "-i",
      "Ref::inputfile",
      "-c",
      "Ref::codec",
      "-o",
      "Ref::outputfile"
    ],
    "jobRoleArn": "arn:aws:iam::123456789012:role/ECSTask-S3FullAccess",
    "user": "nobody"
  }
}
```

Menguji fungsionalitas GPU

Contoh ketentuan tugas berikut menguji apakah beban kerja GPU AMI yang dideskripsikan dalam [Menggunakan AMI beban kerja GPU](#) sudah dikonfigurasi dengan benar. [Contoh definisi pekerjaan ini menjalankan contoh pengklasifikasi MNIST TensorFlow mendalam dari GitHub](#)

```
{
  "containerProperties": {
    "image": "tensorflow/tensorflow:1.8.0-devel-gpu",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "32000"
      },
      {
        "type": "VCPU",
        "value": "8"
      }
    ],
    "command": [
      "sh",
      "-c",
      "cd /tensorflow/tensorflow/examples/tutorials/mnist; python mnist_deep.py"
    ]
  },
  "type": "container",
  "jobDefinitionName": "tensorflow_mnist_deep"
}
```

Anda dapat membuat file dengan teks JSON sebelumnya yang bernama `tensorflow_mnist_deep.json`, lalu mendaftarkan ketentuan tugas AWS Batch dengan perintah berikut:

```
aws batch register-job-definition --cli-input-json file://tensorflow_mnist_deep.json
```

Tugas paralel multisimpul

Contoh ketentuan tugas berikut menggambarkan tugas paralel multisimpul. Untuk informasi lebih lanjut, lihat [Membangun alur kerja dinamika molekuler yang digabungkan erat dengan tugas paralel multisimpul di AWS Batch](#) dalam blog AWS Compute.

```
{
  "jobDefinitionName": "gromacs-jobdef",
  "jobDefinitionArn": "arn:aws:batch:us-east-2:123456789012:job-definition/gromacs-
jobdef:1",
  "revision": 6,
  "status": "ACTIVE",
  "type": "multinode",
  "parameters": {},
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:1",
        "container": {
          "image": "123456789012.dkr.ecr.us-east-2.amazonaws.com/gromacs_mpi:latest",
          "resourceRequirements": [
            {
              "type": "MEMORY",
              "value": "24000"
            },
            {
              "type": "VCPU",
              "value": "8"
            }
          ],
          "command": [],
          "jobRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
          "ulimits": [],
          "instanceType": "p3.2xlarge"
        }
      }
    ]
  }
}
```

Antrean tugas

Tugas dikirim ke antrean tugas yang menjadi tempatnya berada sampai tugas dapat dijadwalkan untuk berjalan di lingkungan komputasi. Suatu akun AWS dapat memiliki beberapa antrean tugas. Misalnya, Anda dapat membuat antrean yang menggunakan instans Sesuai Permintaan Amazon EC2 untuk tugas prioritas tinggi dan antrean lain yang menggunakan Instans Spot Amazon EC2 untuk tugas prioritas rendah. Antrean tugas memiliki prioritas yang digunakan oleh penjadwal untuk menentukan tugas mana di antrean mana yang harus dievaluasi untuk eksekusi terlebih dahulu.

Topik

- [Membuat antrean tugas](#)
- [Parameter antrean tugas](#)

Membuat antrean tugas

Sebelum Anda dapat mengirimkan tugas di AWS Batch, Anda harus membuat antrean tugas. Saat membuat antrean pekerjaan, Anda mengaitkan satu atau beberapa lingkungan komputasi ke antrian dan menetapkan urutan preferensi.


Anda juga menetapkan prioritas ke antrean pekerjaan yang menentukan urutan yang ditempatkan oleh penjadwal AWS Batch. Ini berarti bahwa, jika lingkungan komputasi dikaitkan dengan lebih dari satu antrian pekerjaan, antrian pekerjaan dengan prioritas lebih tinggi diberikan preferensi.

Membuat antrian pekerjaan Fargate

Untuk membuat antrian pekerjaan Fargate

1. Buka konsol AWS Batch di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Antrian pekerjaan.
4. Pilih Buat.
5. Untuk tipe Orkestrasi, pilih Fargate.
6. Untuk Nama, masukkan nama unik untuk antrian pekerjaan Anda. Namanya bisa mencapai 128 karakter, dan dapat berisi huruf besar dan kecil, angka, dan garis bawah (_).

7. Untuk Prioritas, masukkan nilai bilangan bulat untuk prioritas antrian pekerjaan. Antrian pekerjaan dengan prioritas lebih tinggi dijalankan sebelum antrian pekerjaan prioritas rendah yang terkait dengan lingkungan komputasi yang sama. Prioritas ditentukan dalam urutan menurun. Misalnya, antrean tugas dengan nilai prioritas 10 diberikan preferensi penjadwalan di atas antrean tugas dengan nilai prioritas 1.
8. (Opsional) Untuk kebijakan Penjadwalan Nama Sumber Daya Amazon (ARN), pilih kebijakan penjadwalan yang ada.
9. Untuk lingkungan komputasi Terhubung, pilih satu atau beberapa lingkungan komputasi dari daftar yang akan dikaitkan dengan antrian pekerjaan. Pilih lingkungan komputasi dalam urutan yang Anda inginkan antrean untuk mencoba penempatan antrian pekerjaan. Penjadwal pekerjaan menggunakan urutan tempat Anda memilih lingkungan komputasi untuk menentukan lingkungan komputasi mana yang memulai pekerjaan tertentu. Sebelum Anda dapat mengaitkannya dengan antrian pekerjaan, lingkungan komputasi harus dalam status. VALID Anda dapat mengaitkan hingga tiga lingkungan komputasi dengan antrean tugas.

 Note


Semua lingkungan komputasi yang terkait dengan antrian pekerjaan harus berbagi model penyediaan yang sama. AWS Batch tidak mendukung pencampuran model penyediaan dalam satu antrian pekerjaan.

10. Untuk urutan lingkungan Compute, pilih panah atas dan bawah untuk mengonfigurasi urutan yang Anda inginkan.
11. Pilih Buat antrean pekerjaan untuk menyelesaikan dan membuat antrian pekerjaan Anda.

Membuat antrian pekerjaan Amazon EC2

Untuk membuat antrean pekerjaan Amazon EC2

1. Buka konsol AWS Batch di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Antrian pekerjaan.
4. Pilih Buat.
5. Untuk jenis Orkestrasi, pilih Amazon Elastic Compute Cloud (Amazon EC2).

6. Untuk Nama, masukkan nama unik untuk antrian pekerjaan Anda. Namanya bisa mencapai 128 karakter, dan dapat berisi huruf besar dan kecil, angka, dan garis bawah (_).
 7. Untuk Prioritas, masukkan nilai bilangan bulat untuk prioritas antrian pekerjaan. Antrian pekerjaan dengan prioritas lebih tinggi dijalankan sebelum antrian pekerjaan prioritas rendah yang terkait dengan lingkungan komputasi yang sama. Prioritas ditentukan dalam urutan menurun. Misalnya, antrean tugas dengan nilai prioritas 10 diberikan preferensi penjadwalan di atas antrean tugas dengan nilai prioritas 1.
 8. (Opsional) Untuk kebijakan Penjadwalan Nama Sumber Daya Amazon (ARN), pilih kebijakan penjadwalan yang ada.
 9. Untuk lingkungan komputasi Terhubung, pilih satu atau beberapa lingkungan komputasi dari daftar yang akan dikaitkan dengan antrian pekerjaan. Pilih lingkungan komputasi dalam urutan yang Anda inginkan antrean untuk mencoba penempatan antrian pekerjaan. Penjadwal pekerjaan menggunakan urutan tempat Anda memilih lingkungan komputasi untuk menentukan lingkungan komputasi mana yang memulai pekerjaan tertentu. Sebelum Anda dapat mengaitkannya dengan antrian pekerjaan, lingkungan komputasi harus dalam status. VALID Anda dapat mengaitkan hingga tiga lingkungan komputasi dengan antrean tugas. Jika Anda tidak memiliki lingkungan komputasi yang ada, pilih Buat lingkungan komputasi
-  **Note**


Semua lingkungan komputasi yang terkait dengan antrian pekerjaan harus berbagi model penyediaan yang sama. AWS Batch tidak mendukung pencampuran model penyediaan dalam satu antrian pekerjaan.
10. Untuk urutan lingkungan Compute, pilih panah atas dan bawah untuk mengonfigurasi urutan yang Anda inginkan.
 11. Pilih Buat antrean pekerjaan untuk menyelesaikan dan membuat antrian pekerjaan Anda.

Membuat antrian pekerjaan Amazon EKS


Untuk membuat antrean pekerjaan Amazon EKS

1. Buka konsol AWS Batch di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Antrian pekerjaan.
4. Pilih Buat.

5. Untuk jenis Orkestrasi, pilih Amazon Elastic Kubernetes Service (Amazon EKS).
6. Untuk Nama, masukkan nama unik untuk antrian pekerjaan Anda. Namanya bisa mencapai 128 karakter, dan dapat berisi huruf besar dan kecil, angka, dan garis bawah (_).
7. Untuk Priority (Prioritas), masukkan nilai integer untuk prioritas antrean tugas. Antrian pekerjaan dengan prioritas lebih tinggi dijalankan sebelum antrian pekerjaan prioritas rendah yang terkait dengan lingkungan komputasi yang sama. Prioritas ditentukan dalam urutan menurun. Misalnya, antrean tugas dengan nilai prioritas 10 diberikan preferensi penjadwalan di atas antrean tugas dengan nilai prioritas 1.
8. (Opsional) Untuk kebijakan Penjadwalan Nama Sumber Daya Amazon (ARN), pilih kebijakan penjadwalan yang ada.
9. Untuk lingkungan komputasi Terhubung, pilih satu atau beberapa lingkungan komputasi dari daftar yang akan dikaitkan dengan antrian pekerjaan. Pilih lingkungan komputasi dalam urutan yang Anda inginkan antrean untuk mencoba penempatan antrian pekerjaan. Penjadwal pekerjaan menggunakan urutan tempat Anda memilih lingkungan komputasi untuk menentukan lingkungan komputasi mana yang memulai pekerjaan tertentu. Sebelum Anda dapat mengaitkannya dengan antrian pekerjaan, lingkungan komputasi harus dalam status. VALID Anda dapat mengaitkan hingga tiga lingkungan komputasi dengan antrean tugas.

 Note

Semua lingkungan komputasi yang terkait dengan antrian pekerjaan harus berbagi model penyediaan yang sama. AWS Batch tidak mendukung pencampuran model penyediaan dalam satu antrian pekerjaan.

 Note

Semua lingkungan komputasi yang dikaitkan dengan antrean tugas harus berbagi arsitektur yang sama. AWS Batch tidak mendukung pencampuran jenis arsitektur lingkungan komputasi dalam satu antrean tugas.

10. Untuk urutan lingkungan Compute, pilih panah atas dan bawah untuk mengonfigurasi urutan yang Anda inginkan.
11. Pilih Buat antrean pekerjaan untuk menyelesaikan dan membuat antrian pekerjaan Anda.

Templat antrian tugas

Berikut ini adalah template antrian pekerjaan kosong. Anda dapat menggunakan template ini untuk membuat antrian pekerjaan Anda. Anda kemudian dapat menyimpan antrian pekerjaan ini ke file dan menggunakannya dengan AWS CLI `--cli-input-json` opsi. Untuk informasi selengkapnya tentang parameter ini, lihat [CreateJobQueue](#) di Referensi AWS Batch API.

```
{
  "computeEnvironmentOrder": [
    {
      "computeEnvironment": "",
      "order": 0
    }
  ],
  "jobQueueName": "",
  "jobStateTimeLimitActions": [
    {
      "state": "RUNNABLE",
      "action": "CANCEL",
      "maxTimeSeconds": 0,
      "reason": ""
    }
  ],
  "priority": 0,
  "schedulingPolicyArn": "",
  "state": "ENABLED",
  "tags": {
    "KeyName": ""
  }
}
```

Note

Anda dapat menghasilkan template antrian pekerjaan sebelumnya dengan perintah berikut.
AWS CLI

```
$ aws batch create-job-queue --generate-cli-skeleton
```

Parameter antrean tugas

Antrian pekerjaan dibagi menjadi empat komponen dasar: name, state, priority, dan compute environment order. Bagian ini membahas komponen-komponen yang terkait dengan komponen-komponen ini.

Topik

- [Nama antrean tugas](#)
- [Tindakan batas waktu status antrian pekerjaan](#)
- [Prioritas](#)
- [Kebijakan penjadwalan](#)
- [Status](#)
- [Urutan lingkungan komputasi](#)
- [Tag](#)

Nama antrean tugas

[jobQueueName](#)

Nama untuk antrean tugas Anda. Maksimum 128 huruf (huruf besar dan huruf kecil), angka, dan garis bawah diperbolehkan.

Tipe: String

Diperlukan: Ya

Tindakan batas waktu status antrian pekerjaan

[jobStateTimeLimitActions](#)

Serangkaian tindakan yang AWS Batch dilakukan pada pekerjaan yang tetap berada di kepala antrian pekerjaan dalam keadaan yang ditentukan lebih lama dari waktu yang ditentukan. AWS Batch akan melakukan setiap tindakan setelah `maxTimeSeconds` berlalu. (Catatan: Nilai minimum untuk `maxTimeSeconds` adalah 600 (10 menit) dan nilai maksimumnya adalah 86.400 (24 jam).)

Tipe: Array objek `JobStateTimeLimitActions`

Diperlukan: Tidak

Prioritas

[priority](#)

Prioritas antrean tugas. Antrean tugas dengan prioritas yang lebih tinggi (atau nilai integer yang lebih tinggi untuk parameter `priority`) akan dievaluasi pertama ketika dikaitkan dengan lingkungan komputasi yang sama. Prioritas ditentukan dalam urutan turun. Misalnya, antrean tugas dengan nilai prioritas 10 mendapat preferensi penjadwalan di atas antrean tugas dengan nilai prioritas 1. Semua lingkungan komputasi harus berupa Amazon EC2 EC2 (SPOT atau) atau FARGATE Fargate (atau). `FARGATE_SPOT` Lingkungan komputasi Amazon EC2 dan Fargate tidak dapat dicampur.

Jenis: Integer

Diperlukan: Ya

Kebijakan penjadwalan

[schedulingPolicyArn](#)

Nama Sumber Daya Amazon (ARN) dari kebijakan penjadwalan untuk antrean pekerjaan. Antrian pekerjaan yang tidak memiliki kebijakan penjadwalan dijadwalkan dalam model first-in, first-out (FIFO). Setelah antrian pekerjaan memiliki kebijakan penjadwalan, itu dapat diganti tetapi tidak dapat dihapus. Antrian pekerjaan tanpa kebijakan penjadwalan dijadwalkan sebagai antrian pekerjaan FIFO dan tidak dapat menambahkan kebijakan penjadwalan. Antrian pekerjaan dengan kebijakan penjadwalan dapat memiliki maksimum 500 pengidentifikasi saham adil aktif. Ketika batas telah tercapai, pengajuan pekerjaan apa pun yang menambahkan pengenal saham adil baru gagal.

Tipe: String

Wajib: Tidak

Status

state

Status antrean tugas. Jika status antrean tugas adalah ENABLED (nilai default), antrean dapat menerima tugas. Jika status antrean tugas adalah DISABLED, tugas baru tidak dapat ditambahkan ke antrean, tetapi tugas yang sudah ada dalam antrean dapat diselesaikan.

Tipe: String

Nilai yang valid: ENABLED | DISABLED

Diperlukan: Tidak

Urutan lingkungan komputasi

computeEnvironmentOrder

Set lingkungan komputasi yang dipetakan ke antrean tugas dan urutannya yang relatif terhadap satu sama lain. Penjadwal tugas menggunakan parameter ini untuk menentukan lingkungan komputasi mana yang harus menjalankan tugas tertentu. Lingkungan komputasi harus berada di tahapan VALID sebelum Anda dapat mengaitkannya dengan antrean tugas. Anda dapat mengaitkan hingga tiga lingkungan komputasi dengan antrean tugas. Semua lingkungan komputasi harus berupa Amazon EC2 (SPOT atau) atau FARGATE (atau). FARGATE_SPOT Lingkungan komputasi Amazon EC2 dan Fargate tidak dapat dicampur.

Note

Semua lingkungan komputasi yang terkait dengan antrian pekerjaan harus berbagi arsitektur yang sama. AWS Batch tidak mendukung pencampuran tipe arsitektur lingkungan komputasi dalam satu antrian pekerjaan.

Tipe: Array objek [ComputeEnvironmentOrder](#)

Diperlukan: Ya

`computeEnvironment`

Amazon Resource Name (ARN) lingkungan komputasi.

Tipe: String

Diperlukan: Ya

`order`

Urutan lingkungan komputasi. Lingkungan komputasi dicoba dalam urutan naik. Sebagai contoh, jika dua lingkungan komputasi dikaitkan dengan suatu antrean tugas, lingkungan komputasi dengan nilai integer `order` yang lebih rendah akan dicoba untuk penempatan tugas terlebih dahulu.

Tag

[tags](#)

Tanda pasangan kunci-nilai untuk dikaitkan dengan antrean tugas. Untuk informasi selengkapnya, lihat [Menandai sumber daya AWS Batch Anda](#).

Tipe: Peta antar string

Diperlukan: Tidak

Penjadwalan Job

AWS BatchPenjadwal mengevaluasi kapan, di mana, dan bagaimana menjalankan pekerjaan yang dikirimkan ke antrian pekerjaan. Jika Anda tidak menentukan kebijakan penjadwalan saat membuat antrian pekerjaan, penjadwal AWS Batch pekerjaan secara default akan menggunakan strategi first-in, first-out (FIFO). Strategi FIFO dapat menyebabkan pekerjaan penting “terjebak” di belakang pekerjaan yang diajukan sebelumnya. Dengan menentukan kebijakan penjadwalan yang berbeda, Anda dapat mengalokasikan sumber daya komputasi sesuai dengan kebutuhan spesifik Anda.

Note

Jika Anda ingin menjadwalkan urutan spesifik tempat pekerjaan dijalankan, gunakan [dependsOn](#) parameter [SubmitJob](#) untuk menentukan dependensi untuk setiap pekerjaan.

Jika Anda membuat kebijakan penjadwalan dan melampirkannya ke antrian pekerjaan, penjadwalan pembagian yang adil akan diaktifkan. Jika antrian pekerjaan memiliki kebijakan penjadwalan, kebijakan penjadwalan menentukan urutan pekerjaan yang dijalankan. Untuk informasi selengkapnya, lihat [Kebijakan penjadwalan](#).

Bagikan pengidentifikasi

Anda dapat menggunakan pengidentifikasi berbagi untuk menandai pekerjaan dan membedakan antara pengguna dan beban kerja. AWS BatchPenjadwal melacak penggunaan untuk setiap pengidentifikasi saham yang adil dengan menggunakan ($T * weightFactor$) rumus, di mana T penggunaan vCPU dari waktu ke waktu. Penjadwal memilih pekerjaan dengan penggunaan terendah dari pengidentifikasi saham. Anda dapat menggunakan pengenal saham yang adil tanpa mengesampingkannya.

Note

Pengidentifikasi berbagi unik dalam antrian pekerjaan dan tidak digabungkan di seluruh antrian pekerjaan.

Anda dapat mengatur prioritas penjadwalan untuk mengonfigurasi urutan pekerjaan yang dijalankan pada pengenal berbagi. Pekerjaan dengan prioritas penjadwalan yang lebih tinggi dijadwalkan

terlebih dahulu. Jika Anda tidak menentukan kebijakan penjadwalan, semua pekerjaan yang dikirimkan ke antrian pekerjaan dijadwalkan dalam urutan FIFO. Saat mengirimkan pekerjaan, Anda tidak dapat menentukan pengenal berbagi atau prioritas penjadwalan.

Note

Sumber daya komputasi terlampir dialokasikan secara merata di antara semua pengidentifikasi saham kecuali secara eksplisit diganti.

Penjadwalan saham yang adil

Penjadwalan saham yang adil menyediakan serangkaian kontrol untuk membantu menjadwalkan pekerjaan.

Note

Untuk informasi selengkapnya tentang parameter kebijakan penjadwalan, lihat [Parameter kebijakan penjadwalan](#).

- **Share decay seconds** — Periode waktu (dalam detik) yang digunakan AWS Batch penjadwal untuk menghitung persentase saham yang adil untuk setiap pengidentifikasi saham yang adil. Nilai nol menunjukkan bahwa hanya penggunaan saat ini yang diukur. Waktu pembusukan yang lebih lama memberi lebih banyak bobot waktu.

Note

Periode waktu pembusukan dihitung sebagai: $shareDecaySeconds + OrderMinutes$ di $OrderMinutes$ mana waktu dalam urutan dalam menit.

- **Komputasi reservasi** — Mencegah pekerjaan dalam satu pengidentifikasi saham menggunakan semua sumber daya yang dilampirkan ke antrian pekerjaan. Rasio cadangan `ActiveFairShares` adalah $computeReservation/100)^{ActiveFairShares}$ di mana jumlah pengidentifikasi saham adil aktif.

Note

Jika pengidentifikasi saham memiliki pekerjaan diSUBMITTED,,, PENDING RUNNABLESTARTING, atau RUNNING negara bagian, itu dianggap sebagai pengidentifikasi saham aktif. Setelah periode waktu pembusukan berakhir, pengidentifikasi saham dianggap tidak aktif.

- Faktor berat - Faktor berat untuk pengidentifikasi saham. Nilai default adalah 1. Nilai yang lebih rendah memungkinkan pekerjaan dari pengidentifikasi berbagi berjalan atau memberikan runtime tambahan ke pengidentifikasi berbagi. Misalnya, pekerjaan yang menggunakan pengidentifikasi saham dengan faktor bobot 0,125 (1/8) ditugaskan delapan kali sumber daya komputasi pekerjaan yang menggunakan pengidentifikasi saham dengan faktor bobot 1.

Note

Anda hanya perlu mendefinisikan atribut ini ketika Anda perlu memperbarui faktor bobot default 1.

Lingkungan komputasi

Antrean tugas dipetakan ke satu atau beberapa lingkungan komputasi. Lingkungan komputasi berisi instans kontainer Amazon ECS yang digunakan untuk menjalankan tugas batch dalam kontainer. Lingkungan komputasi tertentu juga dapat dipetakan ke satu atau lebih dari satu antrean tugas. Dalam antrian pekerjaan, lingkungan komputasi terkait masing-masing memiliki urutan yang digunakan oleh penjadwal untuk menentukan di mana pekerjaan yang siap dijalankan akan dijalankan. Jika lingkungan komputasi pertama memiliki status VALID dan memiliki sumber daya yang tersedia, pekerjaan dijadwalkan ke instance kontainer dalam lingkungan komputasi tersebut. Jika lingkungan komputasi pertama memiliki status INVALID atau tidak dapat menyediakan sumber daya komputasi yang sesuai, penjadwal mencoba menjalankan pekerjaan di lingkungan komputasi berikutnya.


Topik

- [Lingkungan komputasi terkelola](#)
- [Lingkungan komputasi tidak terkelola](#)
- [AMI sumber daya komputasi](#)
- [Luncurkan dukungan templat](#)
- [Membuat lingkungan komputasi](#)
- [Templat lingkungan komputasi](#)
- [Parameter lingkungan komputasi](#)
- [Konfigurasi EC2](#)
- [Strategi alokasi](#)
- [Memperbarui lingkungan komputasi](#)
- [Lingkungan komputasi Amazon EKS](#)
- [Manajemen Memori Sumber Daya Komputasi](#)

Lingkungan komputasi terkelola

Anda dapat menggunakan lingkungan komputasi terkelola untuk AWS Batch mengelola kapasitas dan jenis instans sumber daya komputasi dalam lingkungan. Ini didasarkan pada spesifikasi sumber daya komputasi yang Anda tentukan saat membuat lingkungan komputasi. Anda dapat memilih untuk menggunakan Instans Sesuai Permintaan Amazon EC2 dan Instans Spot Amazon EC2.


Selain itu, Anda dapat menggunakan kapasitas Fargate dan Fargate Spot di lingkungan komputasi terkelola Anda. Saat menggunakan Instans Spot, Anda dapat menetapkan harga maksimum secara opsional. Dengan cara ini, Instans Spot hanya diluncurkan ketika harga Instans Spot berada di bawah persentase tertentu dari harga On-Demand.

 Important

Instans Fargate Spot tidak didukung pada Windows containers on AWS Fargate. Antrian pekerjaan akan diblokir jika FargateWindows pekerjaan dikirimkan ke antrian pekerjaan yang hanya menggunakan lingkungan komputasi Fargate Spot.

Lingkungan komputasi terkelola meluncurkan instans Amazon EC2 ke dalam VPC dan subnet yang Anda tentukan, lalu mendaftarkannya dengan kluster Amazon ECS. Instans Amazon EC2 memerlukan akses jaringan eksternal untuk berkomunikasi dengan titik akhir layanan Amazon ECS. Beberapa subnet tidak menyediakan instans Amazon EC2 dengan alamat IP publik. Jika instans Amazon EC2 Anda tidak memiliki alamat IP publik, mereka harus menggunakan terjemahan alamat jaringan (NAT) untuk mendapatkan akses ini. Untuk informasi lebih lanjut, lihat [Gateway NAT](#) dalam Panduan Pengguna Amazon VPC. Untuk informasi lebih lanjut tentang cara membuat VPC, lihat [Membuat cloud pribadi virtual](#).

Secara default, lingkungan komputasi AWS Batch terkelola menggunakan versi terbaru yang disetujui dari Amazon ECS AMI yang dioptimalkan untuk sumber daya komputasi. Namun, Anda mungkin perlu membuat AMI Anda sendiri untuk digunakan di lingkungan komputasi terkelola Anda karena berbagai alasan. Untuk informasi selengkapnya, lihat [AMI sumber daya komputasi](#).

 Note

AWS Batch tidak secara otomatis memutakhirkan AMI di lingkungan komputasi setelah dibuat. Misalnya, AMI tidak memperbarui AMI di lingkungan komputasi Anda saat AMI Amazon ECS yang dioptimalkan versi yang lebih baru dirilis. Anda bertanggung jawab atas manajemen sistem operasi tamu. Ini termasuk pembaruan dan patch keamanan. Anda juga bertanggung jawab atas perangkat lunak aplikasi tambahan atau utilitas yang Anda instal di sumber daya komputasi. Ada dua cara untuk menggunakan AMI baru untuk AWS Batch pekerjaan Anda. Metode asli adalah menyelesaikan langkah-langkah ini:

1. Buat lingkungan komputasi baru dengan AMI baru.
2. Tambahkan lingkungan komputasi ke antrean tugas yang ada.

3. Hapus lingkungan komputasi sebelumnya dari antrean tugas Anda.
4. Hapus lingkungan komputasi sebelumnya.

Pada April 2022, AWS Batch menambahkan dukungan yang ditingkatkan untuk memperbarui lingkungan komputasi. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#). Untuk menggunakan pembaruan lingkungan komputasi yang disempurnakan untuk memperbarui AMI, ikuti aturan berikut:

- Jangan menyetel parameter service role ([serviceRole](#)) atau setel ke peran `AWSServiceRoleForBatch` terkait layanan.
- Tetapkan parameter strategi alokasi ([allocationStrategy](#)) ke `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` atau `SPOT_PRICE_CAPACITY_OPTIMIZED`.
- Setel pembaruan ke parameter versi gambar terbaru ([updateToLatestImageVersion](#)) ke `true`.
- Jangan tentukan ID AMI di [imageId](#), [imageIdOverride](#) (in [ec2Configuration](#)), atau di template peluncuran ([launchTemplate](#)). Dalam hal ini, AWS Batch pilih AMI Amazon ECS terbaru yang dioptimalkan yang didukung AWS Batch pada saat pembaruan infrastruktur dimulai. Atau, Anda dapat menentukan ID AMI di [imageIdOverride](#) parameter [imageId](#) atau, atau template peluncuran yang diidentifikasi oleh [LaunchTemplate](#) properti. Mengubah salah satu properti ini memulai pembaruan infrastruktur. Jika ID AMI ditentukan dalam template peluncuran, ID AMI tidak dapat diganti dengan menentukan ID AMI di [imageIdOverride](#) parameter [imageId](#) atau. Itu hanya dapat diganti dengan menentukan template peluncuran yang berbeda. Atau, jika versi template peluncuran disetel ke `$Default` atau `$Latest`, dengan menyetel versi default baru untuk template peluncuran (jika ada `$Default`) atau dengan menambahkan versi baru ke templat peluncuran (jika ada `$Latest`).

Jika aturan ini diikuti, pembaruan apa pun yang memulai pembaruan infrastruktur akan menyebabkan ID AMI dipilih kembali. Jika [version](#) pengaturan dalam template peluncuran ([launchTemplate](#)) disetel ke `$Latest` atau `$Default`, versi terbaru atau default dari template peluncuran dievaluasi pada saat pembaruan infrastruktur, meskipun tidak [launchTemplate](#) diperbarui.

Pertimbangan saat membuat pekerjaan paralel multi-node

AWS Batch merekomendasikan pembuatan lingkungan komputasi khusus untuk menjalankan pekerjaan multi-node paralel (MNP) dan pekerjaan non-MNP. Ini karena cara kapasitas komputasi dibuat di lingkungan komputasi terkelola Anda. Saat membuat lingkungan komputasi terkelola baru, jika Anda menentukan `minvCpu` nilai yang lebih besar dari nol maka buat kumpulan AWS Batch instance untuk digunakan dengan pekerjaan non-MNP saja. Jika pekerjaan paralel multi-node dikirimkan, AWS Batch buat kapasitas instance baru untuk menjalankan tugas paralel multi-node. Dalam kasus di mana ada pekerjaan paralel single-node dan multi-node yang berjalan di lingkungan komputasi yang sama di mana `maxvCpus` nilai `minvCpus` atau ditetapkan, jika sumber daya komputasi yang diperlukan tidak tersedia AWS Batch akan menunggu pekerjaan saat ini selesai sebelum membuat sumber daya komputasi yang diperlukan untuk menjalankan pekerjaan baru.

Lingkungan komputasi tidak terkelola

Dalam lingkungan komputasi yang tidak terkelola, Anda mengelola sumber daya komputasi Anda sendiri. Anda harus memverifikasi bahwa AMI yang Anda gunakan untuk sumber daya komputasi Anda memenuhi spesifikasi AMI instans kontainer Amazon ECS. Lihat informasi yang lebih lengkap di [Spesifikasi AMI sumber daya komputasi](#) dan [Membuat AMI sumber daya komputasi](#).

Note

AWS Sumber daya Fargate tidak didukung di lingkungan komputasi yang tidak dikelola.

Setelah membuat lingkungan komputasi yang tidak dikelola, gunakan operasi [DescribeComputeEnvironments](#) API untuk melihat detail lingkungan komputasi. Cari kluster Amazon ECS yang terkait dengan lingkungan, lalu secara manual luncurkan instans kontainer Anda ke kluster Amazon ECS tersebut.

AWS CLI Perintah berikut juga menyediakan ARN cluster Amazon ECS.

```
$ aws batch describe-compute-environments \
  --compute-environments unmanagedCE \
  --query "computeEnvironments[].ecsClusterArn"
```

Untuk informasi selengkapnya, lihat [Meluncurkan Instans Kontainer Amazon ECS](#) di Panduan Developer Layanan Kontainer Amazon Elastic. Saat meluncurkan sumber daya komputasi,

tentukan ARN klaster Amazon ECS yang didaftarkan sumber daya dengan data pengguna Amazon EC2 berikut. Ganti `ecsClusterArn` dengan ARN cluster yang Anda peroleh dengan perintah sebelumnya.

```
#!/bin/bash
echo "ECS_CLUSTER=ecsClusterArn" >> /etc/ecs/ecs.config
```

AMI sumber daya komputasi

Secara default, lingkungan komputasi AWS Batch dikelola menggunakan versi terbaru yang disetujui dari Amazon ECS AMI yang dioptimalkan untuk sumber daya komputasi. Namun, Anda mungkin ingin membuat AMI sendiri untuk digunakan untuk lingkungan komputasi dikelola dan tidak dikelola. Jika Anda memerlukan salah satu dari berikut ini, kami sarankan Anda membuat AMI Anda sendiri:

- Meningkatkan ukuran penyimpanan root AMI atau volume data Anda
- Menambahkan volume penyimpanan instans untuk jenis instans Amazon EC2 yang didukung
- Menyesuaikan agen kontainer Amazon ECS
- Menyesuaikan Docker
- Mengonfigurasi AMI beban kerja GPU untuk memungkinkan kontainer mengakses perangkat keras GPU pada jenis instans Amazon EC2 yang didukung

Note

Setelah lingkungan komputasi dibuat, AWS Batch tidak memutakhirkan AMI di lingkungan komputasi. AWS Batch juga tidak memperbarui AMI di lingkungan komputasi Anda saat versi AMI yang dioptimalkan Amazon ECS yang lebih baru tersedia. Anda bertanggung jawab atas manajemen sistem operasi tamu. Ini termasuk pembaruan dan patch keamanan. Anda juga bertanggung jawab atas perangkat lunak aplikasi tambahan atau utilitas yang Anda instal di sumber daya komputasi. Untuk menggunakan AMI baru untuk AWS Batch pekerjaan Anda, lakukan hal berikut:

1. Buat lingkungan komputasi baru dengan AMI baru.
2. Tambahkan lingkungan komputasi ke antrean tugas yang ada.
3. Hapus lingkungan komputasi sebelumnya dari antrean tugas Anda.
4. Hapus lingkungan komputasi sebelumnya.

Pada April 2022, AWS Batch menambahkan dukungan yang ditingkatkan untuk memperbarui lingkungan komputasi. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#). Untuk menggunakan pembaruan lingkungan komputasi yang disempurnakan untuk memperbarui AMI, ikuti aturan berikut:

- Jangan menyetel parameter service role ([serviceRole](#)) atau setel ke peran `AWSServiceRoleForBatch` terkait layanan.
- Tetapkan parameter strategi alokasi ([allocationStrategy](#)) ke `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED`, atau `SPOT_PRICE_CAPACITY_OPTIMIZED`.
- Setel pembaruan ke parameter versi gambar terbaru ([updateToLatestImageVersion](#)) ke `true`.
- Jangan tentukan ID AMI di [imageId](#), [imageIdOverride](#) (in [ec2Configuration](#)), atau di template peluncuran ([launchTemplate](#)). Jika Anda tidak menentukan ID AMI, AWS Batch pilih AMI Amazon ECS terbaru yang dioptimalkan yang AWS Batch mendukung pada saat pembaruan infrastruktur dimulai. Atau, Anda dapat menentukan ID AMI di [imageIdOverride](#) parameter [imageId](#) atau. Atau, Anda dapat menentukan template peluncuran yang diidentifikasi oleh `LaunchTemplate` properti. Mengubah salah satu properti ini memulai pembaruan infrastruktur. Jika ID AMI ditentukan dalam template peluncuran, ID AMI tidak dapat diganti dengan menentukan ID AMI di salah satu [imageIdOverride](#) parameter [imageId](#) atau. ID AMI hanya dapat diganti dengan menentukan template peluncuran yang berbeda. Jika versi template peluncuran disetel ke `$Default` atau `$Latest`, ID AMI dapat diganti dengan menyetel versi default baru untuk template peluncuran (jika `$Default`) atau dengan menambahkan versi baru ke template peluncuran (jika `$Latest`).

Jika aturan ini diikuti, setiap pembaruan yang memulai pembaruan infrastruktur menyebabkan ID AMI dipilih kembali. Jika [version](#) pengaturan dalam template peluncuran ([launchTemplate](#)) disetel ke `$Latest` atau `$Default`, versi terbaru atau default dari template peluncuran dievaluasi pada saat pembaruan infrastruktur, meskipun [launchTemplate](#) tidak diperbarui.

Topik

- [Spesifikasi AMI sumber daya komputasi](#)
- [Membuat AMI sumber daya komputasi](#)
- [Menggunakan AMI beban kerja GPU](#)
- [Penghentian Amazon Linux](#)

Spesifikasi AMI sumber daya komputasi

Spesifikasi AMI sumber daya AWS Batch komputasi dasar terdiri dari yang berikut:

Diperlukan

- Distribusi Linux modern yang menjalankan setidaknya versi 3.10 kernel Linux pada AMI jenis virtualisasi HVM. Kontainer Windows tidak didukung.

Important

Tugas paralel multisimpul hanya dapat berjalan pada sumber daya komputasi yang diluncurkan di instans Amazon Linux dengan paket `ecs-init` terinstal. Kami menyarankan agar Anda menggunakan AMI yang dioptimalkan untuk Amazon ECS default ketika membuat lingkungan komputasi Anda. Anda dapat melakukan ini dengan tidak menentukan AMI khusus. Untuk informasi selengkapnya, lihat [Tugas paralel multisimpul](#).

- Agen kontainer Amazon ECS. Kami menyarankan agar Anda menggunakan versi terbaru. Untuk informasi lebih lanjut, lihat [Menginstal Agen Kontainer Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.
- Driver log `awslogs` harus ditentukan sebagai driver log yang tersedia dengan variabel lingkungan `ECS_AVAILABLE_LOGGING_DRIVERS` ketika agen kontainer Amazon ECS dimulai. Untuk informasi lebih lanjut, lihat, [Konfigurasi Agen Kontainer Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.
- Daemon Docker yang menjalankan setidaknya versi 1.9, dan dependensi waktu aktif Docker. Untuk informasi lebih lanjut, lihat [Memeriksa dependensi waktu aktif](#) dalam dokumentasi Docker.

Note

Kami merekomendasikan versi Docker yang dikirimkan dan diuji dengan versi agen Amazon ECS yang sesuai yang Anda gunakan. Amazon ECS menyediakan changelog

untuk varian Linux dari AMI yang dioptimalkan Amazon ECS. GitHub Untuk informasi lebih lanjut, lihat [Changelog](#).

Direkomendasikan

- Proses pengasuhan dan inialisasi untuk menjalankan dan memantau agen Amazon ECS. AMI yang dioptimalkan untuk Amazon ECS menggunakan proses `upstart ecs-init`, dan sistem operasi lainnya mungkin menggunakan `systemd`. Untuk informasi dan contoh selengkapnya, lihat [Contoh container instance User Data Configuration Script](#) di Amazon Elastic Container Service Developer Guide. Untuk informasi selengkapnya `ecs-init`, lihat [ecs-initproyek](#) di GitHub. Minimalnya, lingkungan komputasi terkelola memerlukan agen Amazon ECS untuk memulai saat boot. Jika agen Amazon ECS tidak berjalan di sumber daya komputasi Anda, maka agen Amazon ECS tidak dapat menerima pekerjaan. AWS Batch

AMI yang dioptimalkan untuk Amazon ECS telah dikonfigurasi dengan persyaratan dan rekomendasi ini. Kami menyarankan Anda menggunakan AMI Amazon ECS yang dioptimalkan atau AMI Amazon Linux dengan `ecs-init` paket yang diinstal untuk sumber daya komputasi Anda. Pilih AMI lain jika aplikasi Anda memerlukan sistem operasi tertentu atau versi Docker yang belum tersedia di AMI tersebut. Untuk informasi lebih lanjut, lihat [AMI yang Dioptimalkan untuk Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

Membuat AMI sumber daya komputasi

Anda dapat membuat sendiri sumber daya komputasi kustom AMI untuk digunakan untuk lingkungan komputasi terkelola dan tidak terkelola. Untuk instruksi, lihat [Spesifikasi AMI sumber daya komputasi](#). Kemudian, setelah membuat AMI kustom, Anda dapat membuat lingkungan komputasi yang menggunakan AMI yang dapat Anda kaitkan dengan antrean pekerjaan. Terakhir, mulailah mengirimkan pekerjaan ke antrian itu.

Untuk membuat AMI sumber daya komputasi khusus

1. Pilih AMI dasar untuk memulai. Basis AMI harus menggunakan virtualisasi HVM. AMI dasar tidak bisa menjadi AMI Windows.

Note

AMI yang Anda pilih untuk lingkungan komputasi harus cocok dengan arsitektur tipe instance yang ingin Anda gunakan untuk lingkungan komputasi tersebut. Misalnya, jika lingkungan komputasi Anda menggunakan tipe A1 instans, AMI sumber daya komputasi yang Anda pilih harus mendukung Arm instans. Amazon ECS menjual keduanya x86 dan Arm versi Amazon ECS yang dioptimalkan Amazon Linux 2 AMI. Untuk informasi lebih lanjut, lihat [AMI Amazon Linux 2 yang dioptimalkan untuk Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

AMI Amazon Linux 2 yang dioptimalkan untuk Amazon ECS adalah AMI default untuk sumber daya komputasi di lingkungan komputasi terkelola. AMI Amazon Linux 2 yang dioptimalkan untuk Amazon ECS telah dikonfigurasi dan diuji di AWS Batch oleh teknisi AWS. Ini adalah AMI minimal yang dapat Anda mulai dan untuk mendapatkan sumber daya komputasi Anda yang berjalan AWS dengan cepat. Untuk informasi lebih lanjut, lihat [AMI yang dioptimalkan untuk Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

Atau, Anda dapat memilih varian Amazon Linux 2 lainnya dan menginstal `ecs-init` paket dengan perintah berikut. Untuk informasi lebih lanjut, lihat [Menginstal agen kontainer Amazon ECS di instans EC2 Amazon Linux 2](#) dalam Panduan Developer Amazon Elastic Container:

```
$ sudo amazon-linux-extras disable docker
$ sudo amazon-linux-extras install ecs-init
```

Misalnya, jika Anda ingin menjalankan beban kerja GPU pada sumber daya AWS Batch komputasi, Anda dapat memulai dengan [Amazon Linux Deep Learning](#) AMI. Kemudian, konfigurasi AMI untuk menjalankan AWS Batch pekerjaan. Untuk informasi selengkapnya, lihat [Menggunakan AMI beban kerja GPU](#).

Important

Anda dapat memilih AMI dasar yang tidak mendukung `ecs-init` paket. Namun, jika Anda melakukannya, Anda harus mengonfigurasi cara untuk memulai agen Amazon ECS saat boot dan tetap berjalan. Anda juga dapat melihat beberapa contoh skrip konfigurasi data pengguna yang digunakan `systemd` untuk memulai dan memantau agen penampung Amazon ECS. Untuk informasi selengkapnya, lihat [Contoh skrip](#)

[konfigurasi data pengguna instance container](#) di Panduan Pengembang Layanan Amazon Elastic Container.

2. Luncurkan instans dari AMI dasar pilihan Anda dengan opsi penyimpanan yang sesuai untuk AMI Anda. Anda dapat mengonfigurasi ukuran dan jumlah volume Amazon EBS terlampir, atau volume penyimpanan instans jika jenis instans yang Anda pilih mendukungnya. Untuk informasi lebih lanjut, lihat [Meluncurkan Instans](#) dan [Penyimpanan Instans Amazon EC2](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.
3. Connect ke instans Anda dengan SSH dan lakukan tugas konfigurasi yang diperlukan. Ini mungkin meliputi salah satu atau semua langkah berikut:
 - Menginstal agen kontainer Amazon ECS. Untuk informasi lebih lanjut, lihat [Menginstal Agen Kontainer Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.
 - Mengonfigurasi skrip untuk memformat volume penyimpanan instans.
 - Menambahkan volume penyimpanan instans atau sistem file Amazon EFS ke file `/etc/fstab` sehingga dapat dipasang saat boot.
 - Mengonfigurasi opsi Docker, seperti mengaktifkan debug atau menyesuaikan ukuran gambar dasar.
 - Menginstal paket atau menyalin file.

Untuk informasi lebih lanjut, lihat [Menyambungkan ke Instans Linux Anda Menggunakan SSH](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

4. Jika Anda telah memulai agen kontainer Amazon ECS di instans, Anda harus menghentikannya dan menghapus file titik pemeriksaan data persisten sebelum membuat AMI Anda. Jika Anda tidak melakukan ini, agen tidak akan mulai di instans yang diluncurkan dari AMI Anda.
 - a. Hentikan agen kontainer Amazon ECS.

- AMI Amazon Linux 2 yang dioptimalkan untuk Amazon ECS:

```
sudo systemctl stop ecs
```

- AMI Amazon Linux yang dioptimalkan untuk Amazon ECS:

```
sudo stop ecs
```

- b. Hapus file titik pemeriksaan data persisten. Secara default, file tersebut ada di direktori `/var/lib/ecs/data/`. Gunakan perintah berikut untuk menghapus file-file ini, jika ada.

```
sudo rm -rf /var/lib/ecs/data/*
```

5. Buat AMI baru dari instans yang berjalan. Untuk informasi selengkapnya, lihat [Membuat AMI Linux yang Didukung Amazon EBS](#) di panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk menggunakan AMI baru dengan AWS Batch

1. Setelah AMI baru dibuat, buat lingkungan komputasi dengan AMI baru. Untuk melakukan ini, pilih jenis gambar dan masukkan ID AMI kustom di ID Gambar timpa kotak saat Anda membuat lingkungan AWS Batch komputasi. Untuk informasi lebih lanjut, lihat [the section called “Untuk membuat lingkungan komputasi terkelola menggunakan sumber daya EC2”](#).

Note

AMI yang Anda pilih untuk lingkungan komputasi harus cocok dengan arsitektur tipe instance yang ingin Anda gunakan untuk lingkungan komputasi tersebut. Misalnya, jika lingkungan komputasi Anda menggunakan tipe A1 instans, AMI sumber daya komputasi yang Anda pilih harus mendukung Arm instans. Amazon ECS menjual keduanya x86 dan Arm versi Amazon ECS yang dioptimalkan Amazon Linux 2 AMI. Untuk informasi lebih lanjut, lihat [AMI Amazon Linux 2 yang dioptimalkan untuk Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

2. Buat antrean tugas dan kaitkan lingkungan komputasi baru Anda. Untuk informasi selengkapnya, lihat [Membuat antrean tugas](#).

Note

Semua lingkungan komputasi yang dikaitkan dengan suatu antrean tugas harus memiliki arsitektur yang sama. AWS Batch tidak mendukung pencampuran tipe arsitektur lingkungan komputasi dalam satu antrean tugas.

3. (Opsional) Kirim sampel tugas ke antrean tugas baru Anda. Lihat informasi selengkapnya di [Contoh ketentuan tugas](#), [Membuat definisi pekerjaan simpul tunggal](#), dan [Mengirimkan pekerjaan](#).

Menggunakan AMI beban kerja GPU

Untuk menjalankan beban kerja GPU di sumber daya komputasi AWS Batch, Anda harus menggunakan AMI dengan dukungan GPU. Untuk informasi lebih lanjut, lihat [Bekerja dengan GPU di Amazon ECS](#) dan [AMI yang dioptimalkan untuk Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

Di lingkungan komputasi terkelola, jika lingkungan komputasi menentukan jenis `p2`, `p3`, `p4`, `p5`, `g3g3s`, atau kelompok `g5` instans apa pun `g4`, gunakan AMI yang dioptimalkan AWS Batch untuk GPU Amazon ECS.

Dalam lingkungan komputasi yang tidak terkelola, AMI yang dioptimalkan untuk GPU Amazon ECS lebih disarankan. Anda dapat menggunakan AWS Command Line Interface atau AWS Systems Manager Parameter Store [GetParameter](#)/[GetParameters](#), dan [GetParametersByPath](#) operasi untuk mengambil metadata untuk AMI yang dioptimalkan oleh GPU Amazon ECS yang direkomendasikan.

Note

Keluarga `p5` instans hanya didukung pada versi yang sama atau lebih lambat `20230912` dari AMI yang dioptimalkan oleh GPU Amazon ECS dan tidak kompatibel `p2` dengan `g2` dan jenis instans. Jika Anda perlu menggunakan `p5` instance, pastikan bahwa lingkungan komputasi Anda tidak berisi `p2` atau `g2` instance dan gunakan AMI Batch AMI default terbaru. Membuat lingkungan komputasi baru akan menggunakan AMI terbaru tetapi jika Anda memperbarui lingkungan komputasi untuk disertakan `p5`, Anda dapat memastikan bahwa Anda menggunakan AMI terbaru dengan menyetel [updateToLatestImageVersion](#) ke `true` dalam `ComputeResource` properti. Untuk informasi selengkapnya tentang kompatibilitas AMI dengan instans GPU, lihat [Bekerja dengan GPU di Amazon ECS di Amazon](#) Elastic Container Service Developer Guide.

Contoh berikut menunjukkan cara menggunakan [GetParameter](#) perintah.

AWS CLI

```
$ aws ssm get-parameter --name /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended \
    --region us-east-2 --output json
```

Outputnya mencakup informasi AMI dalam `Value` parameter.

```
{
  "Parameter": {
    "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended",
    "LastModifiedDate": 1555434128.664,
    "Value": "{\"schema_version\":1,\"image_name\": \"amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-eks\", \"image_id\": \"ami-083c800fe4211192f\", \"os\": \"Amazon Linux 2\", \"ecs_runtime_version\": \"Docker version 18.06.1-ce\", \"ecs_agent_version\": \"1.27.0\"}",
    "Version": 9,
    "Type": "String",
    "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended"
  }
}
```

Python

```
from __future__ import print_function

import json
import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameter(Name='/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended')
jsonVal = json.loads(response['Parameter']['Value'])
print("image_id  = " + jsonVal['image_id'])
print("image_name = " + jsonVal['image_name'])
```

Output hanya mencakup ID AMI dan nama AMI:

```
image_id  = ami-083c800fe4211192f
image_name = amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-eks
```

Contoh berikut menunjukkan penggunaan [GetParameters](#).

AWS CLI

```
$ aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name \
```

```

/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_id \
--region us-east-2 --output json

```

Output mencakup metadata lengkap untuk setiap parameter:

```

{
  "InvalidParameters": [],
  "Parameters": [
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
      "LastModifiedDate": 1555434128.749,
      "Value": "ami-083c800fe4211192f",
      "Version": 9,
      "Type": "String",
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
    },
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
      "LastModifiedDate": 1555434128.712,
      "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
      "Version": 9,
      "Type": "String",
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
    }
  ]
}

```

Python

```

from __future__ import print_function

import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters(
    Names=['/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name',

```



```

        '/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id'])
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])

```

Outputnya mencakup ID AMI dan nama AMI, menggunakan jalur lengkap untuk nama.

```

/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
ami-083c800fe4211192f
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs

```

Contoh berikut menunjukkan cara menggunakan [GetParametersByPath](#) perintah.

AWS CLI

```

$ aws ssm get-parameters-by-path --path /aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended \
                                --region us-east-2 --output json

```

Outputnya mencakup metadata lengkap untuk semua parameter di bawah jalur yang ditentukan.

```

{
  "Parameters": [
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_agent_version",
      "LastModifiedDate": 1555434128.801,
      "Value": "1.27.0",
      "Version": 8,
      "Type": "String",
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_agent_version"
    },
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_runtime_version",
      "LastModifiedDate": 1548368308.213,
      "Value": "Docker version 18.06.1-ce",
      "Version": 1,
      "Type": "String",

```

```

        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_runtime_version"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
        "LastModifiedDate": 1555434128.749,
        "Value": "ami-083c800fe4211192f",
        "Version": 9,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
        "LastModifiedDate": 1555434128.712,
        "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-eks",
        "Version": 9,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
os",
        "LastModifiedDate": 1548368308.143,
        "Value": "Amazon Linux 2",
        "Version": 1,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/os"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
schema_version",
        "LastModifiedDate": 1548368307.914,
        "Value": "1",
        "Version": 1,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/schema_version"
    }
]

```

```
}
```

Python

```
from __future__ import print_function

import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters_by_path(Path='/aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended')
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])
```

Outputnya mencakup nilai semua nama parameter di jalur yang ditentukan, menggunakan jalur lengkap untuk nama.

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_agent_version =
1.27.0
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_runtime_version =
Docker version 18.06.1-ce
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
ami-083c800fe4211192f
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebc
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/os = Amazon Linux 2
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/schema_version = 1
```

Untuk informasi lebih lanjut, lihat [Mengambil Metadata AMI yang dioptimalkan untuk Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

Penghentian Amazon Linux

Amazon Linux AMI (juga disebut Amazon Linux 1) mencapai akhir hayatnya pada 31 Desember 2023. AWS Batch telah mengakhiri dukungan untuk Amazon Linux AMI karena tidak akan menerima pembaruan keamanan atau perbaikan bug mulai 1 Januari 2024. Untuk informasi selengkapnya tentang Amazon Linux end-of-life, lihat [FAQ AL](#).

Kami menyarankan Anda memperbarui lingkungan komputasi berbasis Amazon Linux yang ada ke Amazon Linux 2023 untuk mencegah gangguan beban kerja yang tidak terduga, dan terus menerima keamanan dan pembaruan lainnya.

Lingkungan komputasi Anda menggunakan Amazon Linux AMI dapat terus berfungsi setelah tanggal 31 Desember 2023 end-of-life . Namun, lingkungan komputasi ini tidak akan lagi menerima pembaruan perangkat lunak baru, patch keamanan, atau perbaikan bug dari. AWS Anda bertanggung jawab untuk memelihara lingkungan komputasi ini di Amazon Linux AMI setelahnya end-of-life. Sebaiknya migrasi lingkungan AWS Batch komputasi ke Amazon Linux 2023 atau Amazon Linux 2 untuk menjaga kinerja dan keamanan yang optimal.

Untuk bantuan migrasi AWS Batch dari Amazon Linux AMI ke Amazon Linux 2023 atau Amazon Linux 2, lihat [Memperbarui lingkungan komputasi](#) -. AWS Batch

Luncurkan dukungan templat

AWS Batch mendukung penggunaan templat peluncuran Amazon EC2 dengan lingkungan komputasi EC2 Anda. Dengan templat peluncuran, Anda dapat memodifikasi konfigurasi default sumber daya komputasi AWS Batch tanpa perlu membuat AMI khusus.

Note

Templat peluncuran tidak didukung di sumber daya AWS Fargate.

Anda harus membuat templat peluncuran sebelum dapat mengaitkannya dengan lingkungan komputasi. Anda dapat membuat template peluncuran di konsol Amazon EC2. Atau, Anda dapat menggunakan AWS CLI atau AWS SDK. Sebagai contoh, file JSON berikut merupakan templat peluncuran yang mengubah ukuran volume data Docker untuk AMI sumber daya komputasi AWS Batch default dan mengaturnya agar menjadi terenkripsi.

```
{
  "LaunchTemplateName": "increase-container-volume-encrypt",
  "LaunchTemplateData": {
    "BlockDeviceMappings": [
      {
        "DeviceName": "/dev/xvda",
        "Ebs": {
          "Encrypted": true,

```

```
        "VolumeSize": 100,  
        "VolumeType": "gp2"  
    }  
  ]  
}
```

Anda dapat membuat template peluncuran sebelumnya dengan menyimpan JSON ke file yang dipanggil `lt-data.json` dan menjalankan AWS CLI perintah berikut.

```
aws ec2 --region <region> create-launch-template --cli-input-json file://lt-data.json
```

Untuk informasi lebih lanjut tentang templat peluncuran, lihat [Meluncurkan Instans dari Templat Peluncuran](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Jika Anda menggunakan templat peluncuran untuk membuat lingkungan komputasi, Anda dapat memindahkan parameter lingkungan komputasi yang sudah ada berikut ini ke templat peluncuran Anda:

Note

Misalkan salah satu parameter ini (kecuali tag Amazon EC2) ditentukan baik dalam template peluncuran maupun dalam konfigurasi lingkungan komputasi. Kemudian, parameter lingkungan komputasi diutamakan. Tanda Amazon EC2 digabung antara templat peluncuran dan konfigurasi lingkungan komputasi. Jika ada tabrakan pada kunci tag, nilai dalam konfigurasi lingkungan komputasi akan diutamakan.

- Pasangan kunci Amazon EC2
- ID AMI Amazon EC2
- ID grup keamanan
- Tanda Amazon EC2

Parameter templat peluncuran berikut diabaikan oleh AWS Batch:

- Tipe instans (tentukan tipe instans yang Anda inginkan ketika membuat lingkungan komputasi)
- Peran instans (tentukan peran instans yang Anda inginkan ketika membuat lingkungan komputasi)

- Subnet antarmuka jaringan (tentukan subnet yang Anda inginkan saat membuat lingkungan komputasi)
- Opsi pasar instans (AWS Batch harus mengontrol konfigurasi Instans Spot)
- Nonaktifkan pengakhiran API (AWS Batch harus mengontrol siklus hidup instans)

AWS Batch hanya memperbarui template peluncuran dengan versi template peluncuran baru selama pembaruan infrastruktur. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Data pengguna Amazon EC2 dalam templat peluncuran

Anda dapat memberikan data pengguna Amazon EC2 dalam templat peluncuran Anda yang dijalankan oleh [cloud-init](#) saat instans Anda diluncurkan. Data pengguna Anda dapat melakukan skenario konfigurasi umum, termasuk namun tidak terbatas pada hal-hal berikut:

- [Termasuk pengguna atau grup](#)
- [Menginstal paket](#)
- [Membuat partisi dan sistem file](#)

Data pengguna Amazon EC2 dalam templat peluncuran harus memiliki format [arsip multibagian MIME](#). Ini karena data pengguna Anda digabung dengan data pengguna AWS Batch yang diperlukan untuk mengonfigurasi sumber daya komputasi Anda. Anda dapat menggabungkan beberapa blok data pengguna menjadi satu file multi-bagian MIME. Misalnya, Anda mungkin perlu menggabungkan boothook cloud yang mengonfigurasi daemon Docker dengan skrip shell data pengguna yang menulis informasi konfigurasi untuk agen kontainer Amazon ECS.

Jika Anda menggunakan AWS CloudFormation, [AWS::CloudFormation::Init](#) tipe tersebut dapat digunakan dengan skrip pembantu [cfn-init](#) untuk melakukan skenario konfigurasi umum.

File multi-bagian MIME terdiri dari komponen berikut:

- Jenis konten dan deklarasi batas bagian: `Content-Type: multipart/mixed; boundary="==BOUNDARY=="`
- Deklarasi versi MIME: `MIME-Version: 1.0`
- Satu atau beberapa blok data pengguna yang berisi komponen berikut:
 - Batas pembuka yang menandakan awal dari blok data pengguna: `--==BOUNDARY==` Anda harus menjaga garis sebelum batas ini kosong.

- Deklarasi tipe konten untuk blok: Content-Type: *text/cloud-config*; charset="us-ascii". Untuk informasi lebih lanjut tentang tipe konten, lihat [Dokumentasi Cloud-Init](#). Anda harus menjaga baris setelah deklarasi tipe konten kosong.
- Isi data pengguna, seperti daftar perintah atau `cloud-init` arahan shell.
- Batas penutupan yang menandakan akhir file multi-bagian MIME: `--==BOUNDARY==--` Anda harus menjaga garis sebelum batas penutupan kosong.

Berikut ini adalah contoh file multi-bagian MIME yang dapat Anda gunakan untuk membuat file Anda sendiri.

Note

Jika Anda menambahkan data pengguna ke template peluncuran di konsol Amazon EC2, Anda dapat menempelkannya sebagai teks biasa. Atau, Anda dapat mengunggahnya dari file. Jika Anda menggunakan AWS CLI atau AWS SDK, Anda harus terlebih dahulu base64 menyandikan data pengguna dan mengirimkan string itu sebagai nilai `UserData` parameter saat Anda memanggil [CreateLaunchTemplate](#), seperti yang ditunjukkan dalam file JSON ini.

```
{
  "LaunchTemplateName": "base64-user-data",
  "LaunchTemplateData": {
    "UserData":
      "ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWZzS1jb250YWluZXItZm9sdW..."
  }
}
```

Contoh-contoh

- [Contoh: Memasang sistem file Amazon EFS](#)
- [Contoh: Menimpa konfigurasi agen kontainer Amazon ECS default](#)
- [Contoh: Memasang sistem file Amazon FSx for Lustre yang sudah ada](#)

Contoh: Memasang sistem file Amazon EFS

Example

Contoh file multi-bagian MIME ini mengonfigurasi sumber daya komputasi untuk menginstal paket `amazon-efs-utils` dan memasang sistem file Amazon EFS yang sudah ada di `/mnt/efs`.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs tls,_netdev" >> /etc/fstab
- mount -a -t efs defaults

--==MYBOUNDARY==--
```

Contoh: Menimpa konfigurasi agen kontainer Amazon ECS default

Example

Contoh file multi-bagian MIME ini menimpa pengaturan pembersihan gambar Docker default untuk suatu sumber daya komputasi.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
echo ECS_IMAGE_CLEANUP_INTERVAL=60m >> /etc/ecs/ecs.config
echo ECS_IMAGE_MINIMUM_CLEANUP_AGE=60m >> /etc/ecs/ecs.config
```



```
--==MYBOUNDARY==--
```

Contoh: Memasang sistem file Amazon FSx for Lustre yang sudah ada

Example

Contoh file multi-bagian MIME ini mengonfigurasi sumber daya komputasi untuk menginstal `lustre2.10` paket dari Perpustakaan Ekstra dan memasang sistem file FSx for Lustre yang ada di dan nama mount dari `/scratch fsx` Contoh ini untuk Amazon Linux 2. Untuk petunjuk instalasi distribusi Linux yang lain, lihat [Memasang Klien Lustre](#) dalam Panduan Pengguna Amazon FSx for Lustre. Untuk informasi selengkapnya, lihat [Memasang sistem file Amazon FSx Anda secara otomatis di Panduan](#) Pengguna Amazon FSx for Lustre.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- file_system_id_01=fs-0abcdef1234567890
- region=us-east-2
- fsx_directory=/scratch
- amazon-linux-extras install -y lustre2.10
- mkdir -p ${fsx_directory}
- mount -t lustre ${file_system_id_01}.fsx.${region}.amazonaws.com@tcp:fsx
  ${fsx_directory}

--==MYBOUNDARY==--
```

Titik pemasangan harus dipetakan ke dalam container di anggota [volumes](#) dan [mountPoints](#) properti container.

```
{
  "volumes": [
    {
      "host": {
        "sourcePath": "/scratch"
      },
      "name": "Scratch"
    }
  ]
}
```

```
    }
  ],
  "mountPoints": [
    {
      "containerPath": "/scratch",
      "sourceVolume": "Scratch"
    }
  ],
}
```

Membuat lingkungan komputasi

Sebelum Anda dapat menjalankan pekerjaan di AWS Batch, Anda perlu membuat lingkungan komputasi. Anda dapat membuat lingkungan komputasi terkelola tempat AWS Batch mengelola instans Amazon EC2 AWS atau sumber daya Fargate dalam lingkungan berdasarkan spesifikasi Anda. Sebagai alternatif, Anda dapat membuat lingkungan komputasi terkelola tempat Anda menangani konfigurasi instans Amazon EC2 dalam lingkungan tersebut.

Important

Instance Fargate Spot tidak didukung dalam skenario berikut:

- Pada wadah Amazon Linux dengan arsitektur ARM64.
- Windows containers on AWS Fargate

Antrian pekerjaan akan diblokir dalam skenario ini jika pekerjaan dikirimkan ke antrian pekerjaan yang hanya menggunakan lingkungan komputasi Fargate Spot.

Daftar Isi

- [Untuk membuat lingkungan komputasi terkelola menggunakan sumber daya AWS Fargate](#)
- [Untuk membuat lingkungan komputasi terkelola menggunakan sumber daya EC2](#)
- [Untuk membuat lingkungan komputasi tidak terkelola menggunakan sumber daya EC2](#)
- [Untuk membuat lingkungan komputasi terkelola menggunakan sumber daya Amazon EKS](#)

Untuk membuat lingkungan komputasi terkelola menggunakan sumber daya AWS Fargate

1. Buka AWS Batch konsol di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Compute environments (Lingkungan komputasi).
4. Pilih Buat.
5. Konfigurasi lingkungan komputasi.

Note

Lingkungan komputasi untuk Windows containers on AWS Fargate pekerjaan harus setidaknya satu vCPU.

- a. Untuk konfigurasi lingkungan Compute, pilih Fargate.
 - b. Untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat berisi maksimum 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
 - c. Untuk peran Layanan, pilih peran terkait layanan yang memungkinkan AWS Batch layanan melakukan panggilan ke operasi AWS API yang diperlukan atas nama Anda. Misalnya, pilih [AWSServiceRoleForBatch](#). Untuk informasi selengkapnya, lihat [Izin peran terkait layanan untuk AWS Batch](#).
 - d. (Opsional) Perluas Tag. Untuk menambahkan tanda, pilih Tambahkan tanda. Kemudian, masukkan nama Kunci dan Nilai opsional. Pilih Tambahkan tanda.
 - e. Pilih Halaman berikutnya.
6. Di bagian konfigurasi Instans:
 - a. (Opsional) Untuk Menggunakan kapasitas Fargate Spot, nyalakan Fargate Spot. Untuk informasi tentang Fargate Spot, lihat [Menggunakan Amazon EC2 Spot dan Fargate_spot](#).
 - b. Untuk Maximum vCPUs (vCPUs maksimum), pilih jumlah maksimum penskalaan keluar vCPUs yang dapat dilakukan oleh lingkungan komputasi Anda, terlepas dari permintaan antrean tugas.
 - c. Pilih Halaman berikutnya.

7. Konfigurasi jaringan.

Important

Sumber daya komputasi memerlukan akses untuk berkomunikasi dengan titik akhir layanan Amazon ECS. Ini dapat dilakukan melalui VPC endpoint antarmuka atau melalui sumber daya komputasi yang memiliki alamat IP publik.

Untuk informasi lebih lanjut tentang VPC endpoint antarmuka, lihat [VPC Endpoint Antarmuka Amazon ECS \(AWS PrivateLink\)](#) dalam Panduan Developer Amazon Elastic Container Service.

Jika Anda tidak memiliki VPC endpoint yang dikonfigurasi dan sumber daya komputasi Anda tidak memiliki alamat IP publik, network address translation (NAT) harus digunakan untuk menyediakan akses ini. Untuk informasi lebih lanjut, lihat [Gateway NAT](#) dalam Panduan Pengguna Amazon VPC. Untuk informasi selengkapnya, lihat [the section called "Buat VPC"](#).

- a. Untuk Virtual Private Cloud (VPC) ID, pilih VPC tempat Anda ingin meluncurkan instans.
- b. Untuk Subnet, pilih subnet yang akan digunakan. Secara default, semua subnet dalam VPC yang dipilih tersedia.

Note

AWS Batch di Fargate saat ini tidak mendukung Local Zones. Untuk informasi selengkapnya, lihat [cluster Amazon ECS di Local Zones, Wavelength Zones, dan AWS Outposts di Amazon Elastic Container Service Developer Guide](#).

- c. Untuk Security groups (Grup keamanan), pilih grup keamanan untuk dilampirkan ke instans Anda. Secara default, grup keamanan default untuk VPC Anda akan dipilih.
 - d. Pilih Halaman berikutnya.
8. Untuk Tinjauan, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat lingkungan komputasi.

Untuk membuat lingkungan komputasi terkelola menggunakan sumber daya EC2

1. Buka AWS Batch konsol di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Compute environments (Lingkungan komputasi).
4. Pilih Buat.
5. Konfigurasi lingkungan.
 - a. Untuk konfigurasi lingkungan Compute, pilih Amazon Elastic Compute Cloud (Amazon EC2).
 - b. Untuk jenis Orkestrasi, pilih Dikelola.
 - c. Untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat berisi maksimum 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
 - d. (Opsional) Untuk peran Layanan, pilih peran terkait layanan yang memungkinkan AWS Batch layanan melakukan panggilan ke operasi AWS API yang diperlukan atas nama Anda. Misalnya, pilih `AWSServiceRoleForBatch`. Untuk informasi selengkapnya, lihat [Izin peran terkait layanan untuk AWS Batch](#).
 - e. Untuk Instance role (Peran instans), pilih untuk membuat profil instans baru atau gunakan profil instans yang sudah ada yang dilampiri izin IAM yang diperlukan. Profil instans ini memungkinkan instans container Amazon ECS yang dibuat untuk lingkungan komputasi Anda untuk melakukan panggilan ke operasi AWS API yang diperlukan atas nama Anda. Untuk informasi selengkapnya, lihat [Peran instans Amazon ECS](#). Jika Anda memilih untuk membuat profil instans baru, peran yang diperlukan (`ecsInstanceRole`) akan dibuat untuk Anda.
 - f. (Opsional) Perluas Tag.
 - g. (Opsional) Untuk tag EC2, pilih Tambahkan tag untuk menambahkan tag ke sumber daya yang diluncurkan di lingkungan komputasi. Kemudian, masukkan nama Kunci dan Nilai opsional. Pilih Tambahkan tanda.
 - h. (Opsional) Untuk Tag, pilih Tambahkan tag. Kemudian, masukkan nama Kunci dan Nilai opsional. Pilih Tambahkan tanda.

Untuk informasi selengkapnya, lihat [Menandai sumber daya AWS Batch Anda](#).
 - i. Pilih Halaman berikutnya.

6. Di bagian konfigurasi Instans:


- a. (Opsional) Untuk Aktifkan menggunakan instans Spot, aktifkan Spot. Untuk informasi selengkapnya, lihat [Instans Spot](#).
- b. (Hanya spot) Untuk harga Maksimum% sesuai permintaan, pilih persentase maksimum harga Instans Spot jika dibandingkan dengan harga Sesuai Permintaan untuk jenis instans tersebut sebelum instans diluncurkan. Misalnya, jika harga maksimum Anda adalah 20%, harga Spot harus kurang dari 20% dari harga Sesuai Permintaan saat ini untuk instans EC2 tersebut. Anda selalu membayar harga (pasar) terendah dan tidak pernah lebih dari persentase maksimum Anda. Jika Anda membiarkan bidang ini kosong, nilai default adalah 100% dari harga Sesuai Permintaan.
- c. (Khusus spot) Untuk peran armada Spot, pilih peran IAM Armada Spot Amazon EC2 yang ada untuk diterapkan ke lingkungan komputasi Spot Anda. Jika Anda belum memiliki peran IAM Armada Spot Amazon EC2, Anda harus membuatnya terlebih dahulu. Untuk informasi selengkapnya, lihat [Peran armada spot Amazon EC2](#).

Important


Untuk menandai Instans Spot saat pembuatan, peran IAM Armada Spot Amazon EC2 Anda harus menggunakan kebijakan terkelola AmazonEC2 yang lebih baru SpotFleetTaggingRole. Kebijakan SpotFleetRole terkelola AmazonEC2 tidak memiliki izin yang diperlukan untuk menandai Instans Spot. Lihat informasi yang lebih lengkap di [Instans Spot tidak ditandai pada pembuatan](#) dan [the section called "Penandaan sumber daya Anda"](#).

- d. Untuk vCPU Minimum, pilih jumlah minimum vCPU yang dipertahankan oleh lingkungan komputasi Anda, terlepas dari permintaan antrian pekerjaan.
- e. Untuk vCPU yang Diinginkan, pilih jumlah vCPU yang diluncurkan oleh lingkungan komputasi Anda. Seiring meningkatnya permintaan antrean tugas Anda, AWS Batch dapat meningkatkan jumlah vCPUs yang diinginkan di lingkungan komputasi Anda dan menambahkan instans EC2, hingga vCPUs maksimum. Seiring menurunnya permintaan, AWS Batch dapat mengurangi jumlah vCPUs yang diinginkan di lingkungan komputasi Anda dan menghapus instans sampai ke vCPUs minimum.
- f. Untuk Maximum vCPUs (vCPUs maksimum), pilih jumlah maksimum penskalaan keluar vCPUs yang dapat dilakukan oleh lingkungan komputasi Anda, terlepas dari permintaan antrean tugas.


- g. Untuk `Allowed instance types` (Tipe instans yang diizinkan), pilih tipe instans Amazon EC2 yang dapat diluncurkan. Anda dapat menentukan keluarga instans untuk meluncurkan tipe instans apa pun dalam keluarga tersebut (misalnya, `c5`, `c5n`, atau `p3`). Selain itu, Anda dapat menentukan ukuran tertentu dalam suatu keluarga (seperti `c5.8xlarge`). Jenis instans logam tidak ada dalam keluarga instance. Misalnya, `c5` tidak mencakup `c5.metal`. Anda juga dapat memilih `optimal` untuk memilih jenis instans (dari `C4`, `M4`, dan keluarga `R4` instance) yang sesuai dengan permintaan antrian pekerjaan Anda.

 Note

Ketika Anda membuat lingkungan komputasi, tipe instans yang Anda pilih untuk lingkungan komputasi harus memiliki arsitektur yang sama. Misalnya, Anda tidak dapat mencampur instans x86 dan ARM di lingkungan komputasi yang sama.

 Note

AWS Batch akan menskalakan GPU berdasarkan jumlah yang diperlukan dalam antrian pekerjaan Anda. Untuk menggunakan penjadwalan GPU, lingkungan komputasi harus menyertakan jenis instance dari `p2`, `p3`, `p4`, `p5`, `g3`, `g3sg4`, atau keluarga `g5`.

 Note

Saat ini, `optimal` menggunakan tipe instance dari keluarga `C4`, `M4`, dan `R4` instance. Dalam Wilayah AWS hal itu tidak memiliki tipe instance dari keluarga instance tersebut, tipe instance dari `C5`, `M5`, dan keluarga `R5` instance digunakan.

- h. Perluas `Additional configuration` (Konfigurasi tambahan).
- i. (Opsional) Untuk grup Penempatan, masukkan nama grup penempatan untuk mengelompokkan sumber daya di lingkungan komputasi.
- j. (Opsional) Untuk key pair EC2, pilih public dan private key pair sebagai security credentials saat Anda terhubung ke instans. Untuk informasi selengkapnya tentang pasangan kunci Amazon EC2, lihat pasangan kunci [Amazon EC2 dan instans Linux](#).

- k. Untuk Allocation strategy (Strategi alokasi), pilih strategi alokasi yang akan digunakan ketika memilih tipe instans dari daftar tipe instans yang diizinkan. `BEST_FIT_PROGRESSIVE` biasanya merupakan pilihan yang lebih baik untuk lingkungan komputasi EC2 On-Demand, `SPOT_CAPACITY_OPTIMIZED`, dan `SPOT_PRICE_CAPACITY_OPTIMIZED` untuk lingkungan komputasi EC2 Spot. Untuk informasi selengkapnya, lihat [the section called “Strategi alokasi”](#).
- l. (Opsional) Untuk konfigurasi EC2 pilih Nilai penggantian tipe gambar dan ID Gambar untuk memberikan informasi untuk AWS Batch memilih Amazon Machine Images (AMI) untuk instance di lingkungan komputasi. Jika penggantian ID Gambar tidak ditentukan untuk setiap jenis Gambar, AWS Batch pilih AMI [Amazon ECS terbaru yang dioptimalkan](#). Jika tidak ada jenis Gambar yang ditentukan, defaultnya adalah Amazon Linux 2 untuk instance non-GPU, non AWS Graviton.

 Important

Untuk menggunakan AMI kustom, pilih jenis gambar lalu masukkan ID AMI kustom di kotak penggantian ID Gambar.

[Amazon Linux 2](#)


Default untuk semua keluarga instance AWS berbasis Graviton (misalnya,, C6g M6gR6g, danT4g) dan dapat digunakan untuk semua jenis instans non-GPU.

[Amazon Linux 2 \(GPU\)](#)

Default untuk semua keluarga instance GPU (misalnya P4 danG4) dan dapat digunakan untuk semua jenis instance berbasis non AWS Graviton.

Amazon Linux


Dapat digunakan untuk keluarga instance non-GPU, non AWS Graviton. Dukungan standar untuk Amazon Linux AMI telah berakhir. Untuk informasi lebih lanjut, lihat [Amazon Linux AMI](#).

 Note

AMI yang Anda pilih untuk lingkungan komputasi harus cocok dengan arsitektur tipe instance yang ingin Anda gunakan untuk lingkungan komputasi tersebut. Misalnya,


jika lingkungan komputasi Anda menggunakan tipe A1 instans, AMI sumber daya komputasi yang Anda pilih harus mendukung Arm instance. Amazon ECS menjual keduanya x86 dan Arm versi Amazon ECS yang dioptimalkan Amazon Linux 2 AMI. Untuk informasi lebih lanjut, lihat [AMI Amazon Linux 2 yang dioptimalkan untuk Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

- m. (Opsional) Untuk Launch template (Templat peluncuran), pilih templat peluncuran Amazon EC2 yang sudah ada untuk mengonfigurasi sumber daya komputasi Anda. Versi default templat akan diisi secara otomatis. Untuk informasi selengkapnya, lihat [Luncurkan dukungan templat](#).

 Note

Dalam template peluncuran, Anda dapat menentukan AMI kustom yang Anda buat.


- n. (Opsional) Untuk Launch template version (Versi templat peluncuran), masukkan `$Default`, `$Latest`, atau nomor versi tertentu yang akan digunakan.

 Important

Jika parameter versi template peluncuran adalah `$Default` atau `$Latest`, versi default atau terbaru dari template peluncuran yang ditentukan dievaluasi selama pembaruan infrastruktur. Jika ID AMI yang berbeda dipilih secara default atau versi terbaru dari template peluncuran dipilih, ID AMI tersebut digunakan dalam pembaruan. Untuk informasi selengkapnya, lihat [the section called “Memperbarui ID AMI”](#).

- o. Pilih Halaman berikutnya.

7. Di bagian Konfigurasi Jaringan:


 Important

Sumber daya komputasi memerlukan akses untuk berkomunikasi dengan titik akhir layanan Amazon ECS. Ini dapat dilakukan melalui VPC endpoint antarmuka atau melalui sumber daya komputasi yang memiliki alamat IP publik.

Untuk informasi lebih lanjut tentang VPC endpoint antarmuka, lihat [VPC Endpoint Antarmuka Amazon ECS \(AWS PrivateLink\)](#) dalam Panduan Developer Amazon Elastic Container Service.

Jika Anda tidak memiliki VPC endpoint yang dikonfigurasi dan sumber daya komputasi Anda tidak memiliki alamat IP publik, network address translation (NAT) harus digunakan untuk menyediakan akses ini. Untuk informasi lebih lanjut, lihat [Gateway NAT](#) dalam Panduan Pengguna Amazon VPC. Untuk informasi selengkapnya, lihat [the section called “Buat VPC”](#).

- a. Untuk ID Virtual Private Cloud (VPC), pilih VPC tempat meluncurkan instans Anda.
- b. Untuk Subnet, pilih subnet yang akan digunakan. Secara default, semua subnet dalam VPC yang dipilih tersedia.

 Note

AWS Batch di Amazon EC2 mendukung Local Zones. Untuk informasi selengkapnya, lihat [Local Zones](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux dan cluster [Amazon ECS di Local Zones, Wavelength Zones, dan di Panduan Pengembang Layanan Amazon Elastic Container](#). AWS Outposts

- c. (Opsional) Untuk grup Keamanan, pilih grup keamanan untuk dilampirkan ke instans Anda. Secara default, grup keamanan default untuk VPC Anda akan dipilih.
8. Pilih Halaman berikutnya.
 9. Untuk Tinjauan, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat lingkungan komputasi.

Untuk membuat lingkungan komputasi tidak terkelola menggunakan sumber daya EC2

1. Buka AWS Batch konsol di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Pada halaman Compute Environments, pilih Create.
4. Konfigurasi lingkungan.
 - a. Untuk konfigurasi lingkungan Compute, pilih Amazon Elastic Compute Cloud (Amazon EC2).
 - b. Untuk jenis Orkestrasi, pilih Tidak dikelola.

5. Untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
6. (Opsional) Untuk peran Layanan, pilih peran yang memungkinkan AWS Batch layanan melakukan panggilan ke operasi AWS API yang diperlukan atas nama Anda. Misalnya, pilih `AWSBatchServiceRole`. Untuk informasi selengkapnya, lihat [the section called “Menggunakan Peran Tertaut Layanan”](#).
7. Untuk Maximum vCPUs (vCPUs maksimum), pilih jumlah maksimum penskalaan keluar vCPUs yang dapat dilakukan oleh lingkungan komputasi Anda, terlepas dari permintaan antrean tugas.
8. (Opsional) Perluas Tag. Untuk menambahkan tanda, pilih Tambahkan tanda. Kemudian, masukkan nama Kunci dan Nilai opsional. Pilih Tambahkan tanda. Untuk informasi selengkapnya, lihat [Menandai sumber daya AWS Batch Anda](#).
9. Pilih Halaman berikutnya.
10. Untuk Tinjauan, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat lingkungan komputasi.

Untuk membuat lingkungan komputasi terkelola menggunakan sumber daya Amazon EKS

1. Buka AWS Batch konsol di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Compute environments (Lingkungan komputasi).
4. Pilih Buat.
5. Untuk konfigurasi lingkungan Compute, pilih Amazon Elastic Kubernetes Service (Amazon EKS).
6. Untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).
7. Untuk peran Instance, pilih profil instans yang sudah ada yang memiliki izin IAM yang diperlukan.

Note

Untuk membuat lingkungan komputasi di AWS Batch konsol, pilih profil instance yang memiliki `eks:DescribeCluster` izin `eks:ListClusters` dan.

8. Untuk kluster EKS, pilih kluster Amazon EKS yang ada.
9. Untuk Namespace, masukkan Kubernetes namespace untuk mengelompokkan AWS Batch proses Anda di cluster.
10. (Opsional) Perluas Tag. Pilih Tambah tag dan kemudian masukkan pasangan kunci-nilai.
11. Pilih Halaman berikutnya.
12. (Opsional) Untuk Menggunakan Instans Spot EC2, aktifkan Aktifkan menggunakan instans Spot untuk menggunakan Instans Spot Amazon EC2.
13. (Hanya spot) Untuk harga Maksimum% sesuai permintaan, pilih persentase maksimum harga Instans Spot jika dibandingkan dengan harga Sesuai Permintaan untuk jenis instans tersebut sebelum instans diluncurkan. Misalnya, jika harga maksimum Anda adalah 20%, harga Spot harus kurang dari 20% dari harga Sesuai Permintaan saat ini untuk instans EC2 tersebut. Anda selalu membayar harga (pasar) terendah dan tidak pernah lebih dari persentase maksimum Anda. Jika Anda membiarkan bidang ini kosong, nilai default adalah 100% dari harga Sesuai Permintaan.
14. (Khusus spot) Untuk peran armada Spot, pilih peran IAM armada Amazon EC2 Spot untuk lingkungan komputasi. SPOT

 Important

Peran ini diperlukan jika strategi alokasi diatur ke BEST_FIT atau tidak ditentukan.

15. (Opsional) Untuk vCPU Minimum, pilih jumlah minimum vCPU yang dipertahankan oleh lingkungan komputasi Anda, terlepas dari permintaan antrian pekerjaan.
16. (Opsional) Untuk vCPU Maksimum, pilih jumlah maksimum vCPU yang dapat ditingkatkan oleh lingkungan komputasi Anda, terlepas dari permintaan antrian pekerjaan.
17. Untuk Allowed instance types (Tipe instans yang diizinkan), pilih tipe instans Amazon EC2 yang dapat diluncurkan. Anda dapat menentukan keluarga instans untuk meluncurkan tipe instans apa pun dalam keluarga tersebut (misalnya, c5, c5n, atau p3). Atau, Anda dapat menentukan ukuran tertentu dalam keluarga (misalnya, c5.8xlarge). Jenis instans logam tidak ada dalam keluarga instance. Misalnya, c5 tidak mencakup c5.meta1. Anda juga dapat memilih optimal untuk memilih jenis instans (dari C4, M4, dan keluarga R4 instance) karena Anda perlu yang sesuai dengan permintaan antrian pekerjaan Anda.

Note

Ketika Anda membuat lingkungan komputasi, tipe instans yang Anda pilih untuk lingkungan komputasi harus memiliki arsitektur yang sama. Misalnya, Anda tidak dapat mencampur instans x86 dan ARM di lingkungan komputasi yang sama.

Note

AWS Batch skala GPU berdasarkan jumlah yang diperlukan dalam antrian pekerjaan Anda. Untuk menggunakan penjadwalan GPU, lingkungan komputasi harus menyertakan jenis instance dari p2,,p3,p4,, p5 g3 g3sg4, atau keluarga. g5

Note

Saat ini, optimal menggunakan tipe instans dari keluarga instans C4, M4, dan R4. Dalam Wilayah AWS hal itu tidak memiliki tipe instance dari keluarga instance tersebut, tipe instance dari C5M5,, dan keluarga R5 instance digunakan.

18. (Opsional) Perluas Additional configuration (Konfigurasi tambahan).

- a. (Opsional) Untuk grup Penempatan, masukkan nama grup penempatan untuk mengelompokkan sumber daya di lingkungan komputasi.
- b. Untuk strategi Alokasi, pilih BEST_FIT_PROGRESSIVE.
- c. (Opsional) Untuk Konfigurasi Gambar Mesin Amazon (AMI), pilih konfigurasi Tambahkan gambar mesin amazon (amis). Kemudian, pilih Jenis Gambar, masukkan penggantian ID Gambar, dan Kubernetesversi.

Important

Untuk menggunakan AMI kustom, pilih jenis gambar lalu masukkan ID AMI kustom di kotak penggantian ID Gambar.

Note

Jika penggantian ID Gambar tidak ditentukan untuk setiap jenis Gambar, AWS Batch pilih AMI [Amazon ECS terbaru yang dioptimalkan](#). Jika tidak ada jenis Gambar yang ditentukan, defaultnya adalah Amazon Linux 2 untuk instance non-GPU, non AWS Graviton.

Amazon Linux 2

Default untuk semua keluarga instance AWS berbasis Graviton (misalnya,, C6g M6gR6g, danT4g) dan dapat digunakan untuk semua jenis instans non-GPU.

Amazon Linux 2 (GPU)

Default untuk semua keluarga instance GPU (misalnya, P4 danG4) dan dapat digunakan untuk semua jenis instance AWS berbasis non Graviton.

- d. (Opsional) Untuk template Launch, pilih template peluncuran yang ada.
 - e. (Opsional) Untuk versi Template Luncurkan **\$Default\$Latest**, masukkan, atau nomor versi.
19. Pilih Halaman berikutnya.
 20. Untuk Virtual Private Cloud (VPC) ID, pilih VPC tempat meluncurkan instance.
 21. Untuk Subnet, pilih subnet yang akan digunakan. Secara default, semua subnet dalam VPC yang dipilih tersedia.

Note

AWS Batch di Amazon EKS mendukung Local Zones. Untuk informasi selengkapnya, lihat [Amazon EKS dan AWS Local Zones](#) di Panduan Pengguna Amazon EKS.

22. (Opsional) Untuk grup Keamanan, pilih grup keamanan untuk dilampirkan ke instans Anda. Secara default, grup keamanan default untuk VPC Anda dipilih.
23. Pilih Halaman berikutnya.
24. Untuk Tinjauan, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat lingkungan komputasi.

Templat lingkungan komputasi

Contoh berikut menunjukkan templat lingkungan komputasi kosong. Anda dapat menggunakan templat ini untuk membuat lingkungan komputasi Anda yang kemudian dapat disimpan ke file dan digunakan dengan opsi AWS CLI `--cli-input-json`. Untuk informasi selengkapnya tentang parameter ini, lihat [CreateComputeEnvironment](#) di Referensi AWS Batch API.

```
{
  "computeEnvironmentName": "",
  "type": "UNMANAGED",
  "state": "DISABLED",
  "unmanagedvCpus": 0,
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 0,
    "desiredvCpus": 0,
    "instanceTypes": [
      ""
    ],
    "imageId": "",
    "subnets": [
      ""
    ],
    "securityGroupIds": [
      ""
    ],
    "ec2KeyPair": "",
    "instanceRole": "",
    "tags": {
      "KeyName": ""
    },
    "placementGroup": "",
    "bidPercentage": 0,
    "spotIamFleetRole": "",
    "launchTemplate": {
      "launchTemplateId": "",
      "launchTemplateName": "",
      "version": ""
    },
    "ec2Configuration": [
      {
```

```
        "imageType": "",
        "imageIdOverride": "",
        "imageKubernetesVersion": ""
    }
]
},
"serviceRole": "",
"tags": {
    "KeyName": ""
},
"eksConfiguration": {
    "eksClusterArn": "",
    "kubernetesNamespace": ""
}
}
```

Note

Anda dapat menghasilkan templat lingkungan komputasi sebelumnya dengan perintah AWS CLI berikut.

```
$ aws batch create-compute-environment --generate-cli-skeleton
```

Parameter lingkungan komputasi

Lingkungan komputasi dibagi menjadi beberapa komponen dasar: nama, jenis, dan status lingkungan komputasi, definisi sumber daya komputasi (jika lingkungan komputasi terkelola), konfigurasi Amazon EKS (jika menggunakan sumber daya Amazon EKS), peran layanan yang digunakan untuk memberikan izin IAMAWS Batch, dan tag untuk lingkungan komputasi.

Topik

- [Nama lingkungan komputasi](#)
- [Tipe](#)
- [Status](#)
- [Sumber daya komputasi](#)
- [Konfigurasi Amazon EKS](#)
- [Peran layanan](#)

- [Tag](#)

Nama lingkungan komputasi

`computeEnvironmentName`

Nama untuk lingkungan komputasi Anda. Nama dapat memiliki panjang hingga 128 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).

Tipe: String

Diperlukan: Ya

Tipe

`type`

Tipe lingkungan komputasi. Pilih MANAGED agar AWS Batch mengelola sumber daya komputasi EC2 atau Fargate yang Anda tentukan. Untuk informasi selengkapnya, lihat [Sumber daya komputasi](#). Pilih UNMANAGED untuk mengelola sumber daya komputasi EC2 Anda sendiri.

Tipe: String

Nilai yang valid: MANAGED | UNMANAGED

Diperlukan: Ya

Status

`state`

Status lingkungan komputasi.

Jika statusnya adalah ENABLED, penjadwal AWS Batch akan berupaya menempatkan tugas dalam lingkungan. Tugas ini berasal dari antrean tugas terkait pada sumber daya komputasi. Jika lingkungan komputasi dikelola, instans akan keluar atau masuk secara otomatis berdasarkan permintaan antrian pekerjaan.

Jika statusnya adalah DISABLED, penjadwal AWS Batch tidak akan berupaya menempatkan tugas dalam lingkungan. Pekerjaan yang berada di RUNNING negara bagian STARTING atau

terus berkembang secara normal. Lingkungan komputasi terkelola yang berada dalam DISABLED status tidak diskalakan.

Note

Lingkungan komputasi dalam suatu DISABLED negara bagian mungkin terus dikenakan biaya penagihan. Untuk mencegah biaya tambahan, matikan lalu hapus lingkungan komputasi. Untuk informasi selengkapnya, lihat [DeleteComputeEnvironment](#) di Referensi AWS Batch API dan [Menghindari tagihan tak terduga](#) di Panduan AWS Billing Pengguna.

Ketika sebuah instance menganggur, skala instance turun ke `minvCpus` nilainya. Namun, ukuran instance tidak berubah. Misalnya, pertimbangkan `c5.8xlarge` contoh dengan `minvCpus` nilai 4 dan `desiredvCpus` nilai 36. Instance ini tidak menurunkan skala ke `c5.large` instance.

Tipe: String

Nilai yang valid: ENABLED | DISABLED

Diperlukan: Tidak

Sumber daya komputasi

`computeResources`

Detail sumber daya komputasi yang dikelola oleh lingkungan komputasi. Untuk informasi selengkapnya, lihat [Lingkungan komputasi](#).

Tipe: Objek [ComputeResource](#)

Wajib: Parameter ini wajib untuk lingkungan komputasi terkelola

`type`

Tipe lingkungan komputasi. Anda dapat memilih untuk menggunakan Instans Sesuai Permintaan EC2 (EC2) dan Instans Spot EC2 (SPOT), atau menggunakan kapasitas Fargate (FARGATE) dan kapasitas Fargate Spot (FARGATE_SPOT) di lingkungan komputasi terkelola Anda. Jika Anda memilih SPOT, Anda juga harus menentukan peran Armada Spot Amazon EC2 dengan parameter `spotIamFleetRole`. Untuk informasi selengkapnya, lihat [Peran armada spot Amazon EC2](#).

Nilai yang valid: EC2 | SPOT | FARGATE | FARGATE_SPOT

Diperlukan: Ya

allocationStrategy

Strategi alokasi yang akan digunakan untuk sumber daya komputasi jika tipe instans EC2 yang sesuai tidak cukup untuk dapat dialokasikan. Ini mungkin karena ketersediaan jenis instans di Wilayah AWS atau batas [layanan Amazon EC2](#). Untuk informasi selengkapnya, lihat [Strategi alokasi](#).

Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

BEST_FIT (default)

AWS Batch memilih tipe instans yang paling sesuai dengan kebutuhan tugas dengan mengutamakan tipe instans berbiaya terendah. Jika instans tambahan dari tipe instans yang dipilih tidak tersedia, AWS Batch menunggu instans tambahan tersedia. Jika tidak ada cukup instans yang tersedia, atau jika Anda mencapai batas [layanan Amazon EC2](#), pekerjaan tambahan tidak akan berjalan sampai pekerjaan yang sedang berjalan selesai. Strategi alokasi ini membuat biaya lebih rendah tetapi dapat membatasi penskalaan. Jika Anda menggunakan Armada SpotBEST_FIT, Peran IAM Armada Spot harus ditentukan. Sumber daya komputasi yang menggunakan strategi BEST_FIT alokasi tidak mendukung pembaruan infrastruktur dan tidak dapat memperbarui beberapa parameter. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Note

BEST_FITtidak didukung untuk lingkungan komputasi yang menggunakan sumber daya Amazon EKS.

BEST_FIT_PROGRESSIVE

Gunakan jenis instance tambahan yang cukup besar untuk memenuhi persyaratan pekerjaan dalam antrian. Lebih suka jenis instance dengan biaya lebih rendah untuk setiap

unit vCPU. Jika instans tambahan dari tipe instans yang dipilih sebelumnya tidak tersedia, AWS Batch akan memilih tipe instans baru.

SPOT_CAPACITY_OPTIMIZED

(Hanya tersedia untuk sumber daya komputasi Instans Spot) Gunakan jenis instans tambahan yang cukup besar untuk memenuhi persyaratan pekerjaan dalam antrian. Lebih suka jenis instance yang cenderung tidak terganggu.

SPOT_PRICE_CAPACITY_OPTIMIZED

(Hanya tersedia untuk sumber daya komputasi Instans Spot) 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.

Note

Kami menyarankan Anda menggunakan SPOT_PRICE_CAPACITY_OPTIMIZED daripada SPOT_CAPACITY_OPTIMIZED dalam kebanyakan kasus.

Dengan BEST_FIT_PROGRESSIVE, SPOT_CAPACITY_OPTIMIZED, dan SPOT_PRICE_CAPACITY_OPTIMIZED strategi menggunakan Instans Sesuai Permintaan atau Spot dan BEST_FIT strategi menggunakan Instans Spot, AWS Batch mungkin perlu melebihi maxvCpus untuk memenuhi persyaratan kapasitas Anda. Dalam kondisi ini, AWS Batch tidak pernah melampaui maxvCpus sebanyak lebih dari satu instans.

Nilai yang valid: BEST_FIT | BEST_FIT_PROGRESSIVE | SPOT_CAPACITY_OPTIMIZED | SPOT_PRICE_CAPACITY_OPTIMIZED

Diperlukan: Tidak

minvCpus

Jumlah minimum vCPU yang dipertahankan lingkungan bahkan jika lingkungan komputasi.
DISABLED

Note


Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Tipe: Integer

Wajib: Tidak

`maxvCpus`

Jumlah maksimum vCPU yang dapat didukung oleh AWS Batch lingkungan komputasi.

 Note


Dengan `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED`, dan strategi `SPOT_PRICE_CAPACITY_OPTIMIZED` alokasi menggunakan Instans Sesuai Permintaan atau Spot dan `BEST_FIT` strategi menggunakan Instans Spot, AWS Batch mungkin perlu melebihi `maxvCpus` untuk memenuhi persyaratan kapasitas Anda. Dalam hal ini, AWS Batch tidak pernah melebihi `maxvCpus` lebih dari satu instans. Misalnya, AWS Batch menggunakan tidak lebih dari satu instans dari antara instans yang ditentukan dalam lingkungan komputasi Anda.

Tipe: Integer

Wajib: Tidak

`desiredvCpus`

Jumlah vCPU yang diinginkan di lingkungan komputasi. AWS Batch memodifikasi nilai ini antara nilai minimum dan maksimum berdasarkan permintaan antrian pekerjaan.

 Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.


Tipe: Integer

Wajib: Tidak


`instanceTypes`

Tipe instans yang dapat diluncurkan. Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate. Jangan tentukan itu. Anda dapat menentukan keluarga instans untuk meluncurkan tipe instans apa pun dalam keluarga tersebut (misalnya, `c5`, `c5n`, atau `p3`). Selain itu, Anda dapat menentukan ukuran tertentu dalam suatu keluarga (seperti

c5.8xlarge). Perhatikan bahwa tipe instans metal tidak tercakup dalam keluarga instans (misalnya c5 tidak meliputi c5.metal.) Anda juga dapat memilih `optimal` untuk memilih tipe instans (dari keluarga instans C4, M4, dan R4) yang sesuai dengan permintaan antrean tugas Anda.

 Note

Ketika Anda membuat lingkungan komputasi, tipe instans yang Anda pilih untuk lingkungan komputasi harus berbagi arsitektur yang sama. Misalnya, Anda tidak dapat mencampur instans x86 dan ARM di lingkungan komputasi yang sama.

 Note

Saat ini, `optimal` menggunakan tipe instans dari keluarga instans C4, M4, dan R4. Dalam Wilayah AWS hal itu tidak memiliki tipe instance dari keluarga instance tersebut, tipe instance dari keluarga instans C5, M5, dan R5 digunakan.


Tipe: Array string

Wajib: ya


`imageId`

Parameter ini tidak digunakan lagi.

ID Amazon Machine Image (AMI) digunakan untuk instans yang diluncurkan di lingkungan komputasi. Parameter ini diganti oleh anggota `imageIdOverride` dari struktur `Ec2Configuration`.

 Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

 Note

AMI yang Anda pilih untuk lingkungan komputasi harus cocok dengan arsitektur tipe instance yang ingin Anda gunakan untuk lingkungan komputasi tersebut. Misalnya,


jika lingkungan komputasi Anda menggunakan tipe A1 instans, AMI sumber daya komputasi yang Anda pilih harus mendukung Arm instans. Amazon ECS menjual keduanya x86 dan Arm versi Amazon ECS yang dioptimalkan Amazon Linux 2 AMI. Untuk informasi lebih lanjut, lihat [AMI Amazon Linux 2 yang dioptimalkan untuk Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

Tipe: String

Wajib: Tidak

subnets

Subnet VPC yang menjadi tujuan peluncuran sumber daya komputasi. Subnet ini harus berada dalam VPC yang sama. Sumber daya komputasi Fargate dapat berisi maksimum 16 subnet. Untuk informasi lebih lanjut, lihat [VPC dan Subnet](#) dalam Panduan Pengguna Amazon VPC.

 Note

AWS Batch di Amazon EC2 dan di AWS Batch Amazon EKS mendukung Local Zones. Untuk informasi selengkapnya, lihat [Local Zones](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux, Amazon EKS, dan [AWS Local Zones di Panduan Pengguna Amazon EKS](#) dan kluster Amazon [ECS di Local Zones, Wavelength Zones, dan di Panduan Pengembang Layanan Kontainer Elastis](#) Amazon. AWS Outposts AWS Batch di Fargate saat ini tidak mendukung Local Zones.

Saat memperbarui lingkungan komputasi, jika Anda memberikan daftar kosong subnet VPC, perilaku yang dihasilkan berbeda antara sumber daya komputasi Fargate dan EC2. Untuk sumber daya komputasi Fargate, menyediakan daftar kosong ditangani seolah-olah parameter ini tidak ditentukan dan tidak ada perubahan yang dibuat. Untuk sumber daya komputasi EC2, menyediakan daftar kosong akan menghapus subnet VPC dari sumber daya komputasi. Jika Anda mengubah subnet VPC, pembaruan infrastruktur lingkungan komputasi diperlukan. Ini adalah kasus untuk sumber daya komputasi Fargate dan EC2. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Tipe: Array string

Diperlukan: Ya

securityGroupIds

Grup keamanan Amazon EC2 yang terkait dengan instans yang diluncurkan di lingkungan komputasi. Satu atau beberapa grup keamanan harus ditentukan, baik di `securityGroupIds` atau menggunakan templat peluncuran yang direferensikan di `launchTemplate`. Parameter ini diperlukan untuk pekerjaan yang berjalan pada sumber daya Fargate dan harus berisi setidaknya satu grup keamanan. (Fargate tidak mendukung templat peluncuran.) Jika grup keamanan ditentukan menggunakan `securityGroupIds` dan `launchTemplate`, nilai di `securityGroupIds` akan digunakan.

Saat memperbarui lingkungan komputasi, jika Anda memberikan daftar kosong grup keamanan, perilaku yang dihasilkan berbeda antara sumber daya komputasi Fargate dan EC2. Untuk sumber daya komputasi Fargate, menyediakan daftar kosong ditangani seolah-olah parameter ini tidak ditentukan dan tidak ada perubahan yang dibuat. Untuk sumber daya komputasi EC2, menyediakan daftar kosong akan menghapus grup keamanan dari sumber daya komputasi. Jika Anda mengubah grup keamanan, pembaruan infrastruktur lingkungan komputasi diperlukan. Ini adalah kasus untuk sumber daya komputasi Fargate dan EC2. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Tipe: Array string

Diperlukan: Ya

ec2KeyPair

Pasangan kunci EC2 yang digunakan untuk instans yang diluncurkan di lingkungan komputasi. Anda dapat menggunakan pasangan kunci ini untuk masuk ke instans Anda dengan SSH. Saat memperbarui lingkungan komputasi, jika Anda mengubah key pair EC2, pembaruan infrastruktur lingkungan komputasi diperlukan. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Tipe: String

Wajib: Tidak

instanceRole

Profil instans Amazon ECS untuk dilampirkan ke instans Amazon EC2 dalam lingkungan komputasi. Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate. Jangan tentukan itu. Anda dapat menentukan nama pendek atau lengkap Amazon Resource Name (ARN) untuk profil instans. Misalnya, `ecsInstanceRole` atau `arn:aws:iam::aws_account_id:instance-profile/ecsInstanceRole`. Untuk informasi selengkapnya, lihat [Peran instans Amazon ECS](#).

Saat memperbarui lingkungan komputasi, jika Anda mengubah setelan ini, pembaruan infrastruktur lingkungan komputasi diperlukan. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Tipe: String

Wajib: Tidak

tags

Tanda pasangan kunci-nilai untuk diterapkan ke instans EC2 yang diluncurkan di lingkungan komputasi. Misalnya, Anda dapat menentukan `"Name": "AWS Batch Instance - C4OnDemand"` sebagai tanda agar setiap instans di lingkungan komputasi Anda memiliki nama tersebut. Hal ini sangat membantu untuk mengenali instans AWS Batch Anda di konsol Amazon EC2. Tag ini tidak terlihat saat menggunakan operasi AWS Batch [ListTagsForResourceAPI](#).

Saat memperbarui lingkungan komputasi, jika Anda mengubah tag EC2, pembaruan infrastruktur lingkungan komputasi diperlukan. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Tipe: Peta antar string

Diperlukan: Tidak

placementGroup

Grup penempatan Amazon EC2 yang akan dikaitkan dengan sumber daya komputasi Anda. Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate. Jangan tentukan itu. Jika Anda ingin mengirimkan pekerjaan paralel multi-node ke lingkungan komputasi Anda, pertimbangkan untuk membuat grup penempatan klaster dan mengaitkannya dengan sumber daya komputasi Anda. Hal ini membuat pekerjaan paralel multi-simpul Anda pada pengelompokan logis instans dalam Availability Zone tunggal dengan potensi aliran jaringan tinggi. Untuk informasi lebih lanjut, lihat [Grup Penempatan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Tipe: String

Wajib: Tidak

bidPercentage

Persentase maksimum yang dapat dicapai harga Instans Spot EC2 saat dibandingkan dengan harga Sesuai Permintaan untuk tipe instans tersebut sebelum instans diluncurkan. Misalnya, jika persentase maksimum Anda adalah 20%, harga Spot harus kurang dari 20% dari harga On-Demand saat ini untuk instans EC2 tersebut. Anda selalu membayar harga terendah (pasar) dan tidak pernah lebih dari persentase maksimum Anda. Jika Anda membiarkan bidang ini kosong, nilai default adalah 100% dari harga Sesuai Permintaan. Untuk sebagian besar kasus penggunaan, sebaiknya biarkan bidang ini kosong.

Saat memperbarui lingkungan komputasi, jika Anda mengubah persentase tawaran, pembaruan infrastruktur lingkungan komputasi diperlukan. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Diperlukan: Tidak

spotIamFleetRole

Amazon Resource Name (ARN) dari IAM role Armada Spot Amazon EC2 diterapkan ke lingkungan komputasi SPOT. Peran ini wajib jika strategi alokasi diatur ke BEST_FIT atau jika strategi alokasi tidak ditentukan. Untuk informasi selengkapnya, lihat [Peran armada spot Amazon EC2](#).

Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Important

Untuk menandai Instans Spot Anda saat pembuatan, peran IAM Armada Spot yang ditentukan di sini harus menggunakan kebijakan terkelola SpotFleetTaggingRoleAmazonEC2 yang lebih baru. Kebijakan SpotFleetRole terkelola AmazonEC2 yang direkomendasikan sebelumnya tidak memiliki izin yang diperlukan untuk menandai Instans Spot. Untuk informasi selengkapnya, lihat [Instans Spot tidak ditandai pada pembuatan](#).

Tipe: String

Wajib: Parameter ini wajib untuk lingkungan komputasi SPOT.

launchTemplate

Templat peluncuran opsional untuk dikaitkan dengan sumber daya komputasi Anda. Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate. Jangan tentukan itu. Parameter sumber daya komputasi lainnya yang Anda tentukan dalam operasi [CreateComputeEnvironment](#) atau [UpdateComputeEnvironment](#) API akan menimpa parameter yang sama dalam template peluncuran. Untuk menggunakan templat peluncuran, Anda harus menentukan ID templat peluncuran atau nama templat peluncuran dalam permintaan, tetapi tidak keduanya. Untuk informasi selengkapnya, lihat [Luncurkan dukungan templat](#).

Saat memperbarui lingkungan komputasi, untuk menghapus templat peluncuran kustom dan menggunakan templat peluncuran default, atur `launchTemplateId` atau `launchTemplateName` anggota spesifikasi templat peluncuran ke string kosong. Menghapus

template peluncuran dari lingkungan komputasi tidak akan menghapus AMI yang ditentukan dalam template peluncuran, jika itu yang digunakan. Untuk memperbarui AMI yang dipilih dari template peluncuran, `updateToLatestImageVersion` parameter harus disetel ke `true`. Saat memperbarui lingkungan komputasi, jika Anda mengubah template peluncuran, pembaruan infrastruktur lingkungan komputasi diperlukan. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Tipe: [LaunchTemplateSpecification](#)

objek

Diperlukan: Tidak

`launchTemplateId`

ID templat peluncuran.

Tipe: String

Wajib: Tidak

`launchTemplateName`

Nama templat peluncuran.

Tipe: String

Wajib: Tidak

`version`

Nomor versi templat peluncuran, `$Latest`, atau `$Default`.

Jika nilainya `$Latest`, versi terbaru dari templat peluncuran yang digunakan. Jika nilainya `$Default`, versi default dari templat peluncuran yang digunakan. Selama pembaruan infrastruktur, jika salah satu `$Latest` atau `$Default` ditentukan untuk lingkungan komputasi, AWS Batch evaluasi ulang versi template peluncuran dan mungkin menggunakan versi template peluncuran yang berbeda. Ini bahkan jika template peluncuran tidak ditentukan dalam pembaruan.

Bawaan: `$Default`.

Tipe: String

Wajib: Tidak

ec2Configuration

Memberikan informasi yang digunakan untuk memilih Amazon Machine Images (AMI) untuk instans di lingkungan komputasi EC2. Jika `Ec2Configuration` tidak ditentukan, defaultnya adalah [Amazon Linux 2](#) (ECS_AL2). Sebelum 31 Maret 2021, default ini adalah [Amazon Linux](#) (ECS_AL1) untuk instans non-GPU, non AWS Graviton.

Saat memperbarui lingkungan komputasi, jika Anda mengubah parameter ini, pembaruan infrastruktur lingkungan komputasi diperlukan. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Note

Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate.

Tipe: Array objek [Ec2Configuration](#)

Diperlukan: Tidak

imageIdOverride

ID AMI yang digunakan untuk instance yang diluncurkan di lingkungan komputasi yang cocok dengan jenis gambar. Pengaturan ini akan menggantikan `imageId` yang diatur dalam objek `computeResource`.

Tipe: String

Wajib: Tidak

imageKubernetesVersion

KubernetesVersi untuk lingkungan komputasi. Jika Anda tidak menentukan nilai, versi terbaru yang AWS Batch mendukung akan digunakan.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Diperlukan: Tidak

imageType

Jenis citra untuk dicocokkan dengan tipe instans untuk memilih AMI. Nilai yang didukung berbeda untuk ECS dan EKS sumber daya.

ECS

Jika `imageIdOverride` parameter tidak ditentukan, maka Amazon [Linux 2 AMI \(ECS_AL2\) yang dioptimalkan Amazon ECS](#) terbaru akan digunakan. Jika jenis gambar baru ditentukan dalam pembaruan, tetapi tidak ada parameter `imageId` atau `imageIdOverride` parameter yang ditentukan, maka Amazon ECS terbaru yang dioptimalkan AMI untuk jenis gambar yang didukung oleh AWS Batch digunakan.

ECS_AL2

[Amazon Linux 2](#): Default untuk semua keluarga instans non-GPU.

ECS_AL2_NVIDIA

[Amazon Linux 2 \(GPU\)](#): Default untuk semua keluarga instans GPU (misalnya P4 danG4) dan dapat digunakan untuk semua jenis instans berbasis AWS Graviton.

ECS_AL1

[Amazon Linux](#). Amazon Linux telah mencapai dukungan standar. end-of-life Untuk informasi lebih lanjut, lihat [Amazon Linux AMI](#).

EKS

Jika `imageIdOverride` parameter tidak ditentukan, maka Amazon Linux [AMI \(EKS_AL2\) yang dioptimalkan Amazon EKS](#) terbaru akan digunakan. Jika jenis gambar baru ditentukan dalam pembaruan, tetapi tidak ada parameter `imageId` atau `imageIdOverride` parameter yang ditentukan, maka Amazon EKS terbaru yang dioptimalkan AMI untuk jenis gambar yang AWS Batch mendukung digunakan.

EKS_AL2

[Amazon Linux 2](#): Default untuk semua keluarga instans non-GPU.

EKS_AL2_NVIDIA

[Amazon Linux 2 \(dipercepat\)](#): Default untuk semua keluarga instans GPU (misalnya, P4 danG4) dan dapat digunakan untuk semua jenis instans AWS berbasis Graviton.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Diperlukan: Ya

Konfigurasi Amazon EKS

Konfigurasi untuk kluster Amazon EKS yang mendukung lingkungan AWS Batch komputasi. Cluster harus ada sebelum lingkungan komputasi dapat dibuat.

`eksClusterArn`

Nama Sumber Daya Amazon (ARN) dari cluster Amazon EKS. Contohnya adalah `arn:aws:eks:us-east-1:123456789012:cluster/ClusterForBatch`.

Tipe: String

Diperlukan: Ya

`kubernetesNamespace`

Namespace dari cluster Amazon EKS. AWS Batch mengelola pod di namespace ini. Nilai tidak dapat dibiarkan kosong atau null. Panjangnya harus kurang dari 64 karakter, tidak dapat diatur ke default, tidak dapat dimulai dengan "kube-," dan harus cocok dengan ekspresi reguler ini: `^[a-z0-9]([-a-z0-9]*[a-z0-9])?$`. Untuk informasi selengkapnya, lihat [Ruang nama](#) dalam dokumentasi. Kubernetes

Tipe: String

Diperlukan: Ya

Jenis: [EksConfiguration](#) Objek

Diperlukan: Tidak

Peran layanan

`serviceRole`

Amazon Resource Name (ARN) lengkap dari IAM role yang mengizinkan AWS Batch untuk membuat panggilan ke layanan AWS lain atas nama Anda. Untuk informasi selengkapnya,


lihat [Menggunakan peran terkait layanan untuk AWS Batch](#). Kami menyarankan Anda untuk tidak menentukan peran layanan. Dengan cara ini, AWS Batch menggunakan peran `AWSServiceRoleForBatch` terkait layanan.

 Important

Jika akun Anda telah membuat peran AWS Batch terkait layanan (`AWSServiceRoleForBatch`), peran tersebut akan digunakan secara default untuk lingkungan komputasi kecuali Anda menentukan peran di sini. Jika peran AWS Batch yang terhubung dengan layanan tidak ada di akun Anda, dan tidak ada peran yang ditentukan di sini, layanan akan mencoba membuat peran AWS Batch yang terhubung dengan layanan di akun Anda. Untuk informasi selengkapnya tentang peran `AWSServiceRoleForBatch` terkait layanan, lihat [Izin peran terkait layanan untuk AWS Batch](#)

Jika lingkungan komputasi dibuat menggunakan peran `AWSServiceRoleForBatch` terkait layanan, itu tidak dapat diubah untuk menggunakan peran IAM biasa. Demikian juga, jika lingkungan komputasi dibuat dengan peran IAM biasa, itu tidak dapat diubah untuk menggunakan peran terkait `AWSServiceRoleForBatch` layanan. Untuk memperbarui parameter lingkungan komputasi yang memerlukan pembaruan infrastruktur untuk berubah, peran `AWSServiceRoleForBatch` terkait layanan harus digunakan. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Jika peran yang Anda tentukan memiliki jalur selain `/`, pastikan untuk menentukan ARN peran lengkap (disarankan) atau awalan nama peran dengan jalur tersebut.

 Note

Bergantung pada cara Anda membuat peran AWS Batch layanan, Nama Sumber Daya Amazon (ARN) mungkin berisi awalan `service-role` jalur. Jika Anda hanya menentukan nama peran layanan, AWS Batch mengasumsikan bahwa ARN Anda tidak menggunakan prefiks jalur `service-role`. Karena itu, kami sarankan Anda menentukan ARN lengkap dari peran layanan Anda ketika membuat lingkungan komputasi.

Tipe: String

Wajib: Tidak

Tag

tags

Tanda pasangan kunci-nilai untuk dikaitkan dengan lingkungan komputasi. Untuk informasi selengkapnya, lihat [Menandai sumber daya AWS Batch Anda](#).

Tipe: Peta antar string

Diperlukan: Tidak

Konfigurasi EC2

AWS Batch menggunakan AMI Amazon ECS yang dioptimalkan untuk lingkungan komputasi EC2 dan EC2 Spot. Defaultnya adalah [Amazon Linux 2](#) (ECS_AL2). Sebelum 31 Maret 2021, default ini adalah [Amazon Linux](#) (ECS_AL1) untuk instans non-GPU, non AWS Graviton.

Note

AWS Batch juga mendukung Amazon Linux 2023.

Amazon Linux AMI (juga disebut Amazon Linux 1) mencapai akhir hayatnya pada 31 Desember 2023. AWS Batch telah mengakhiri dukungan untuk Amazon Linux AMI karena tidak akan menerima pembaruan keamanan atau perbaikan bug mulai 1 Januari 2024. Untuk informasi selengkapnya tentang Amazon Linux end-of-life, lihat [FAQ AL](#).

Kami menyarankan Anda memperbarui lingkungan komputasi berbasis Amazon Linux yang ada ke Amazon Linux 2023 untuk mencegah gangguan beban kerja yang tidak terduga, dan terus menerima keamanan dan pembaruan lainnya.

Lingkungan komputasi Anda menggunakan Amazon Linux AMI dapat terus berfungsi setelah tanggal 31 Desember 2023 end-of-life . Namun, lingkungan komputasi ini tidak akan lagi menerima pembaruan perangkat lunak baru, patch keamanan, atau perbaikan bug dari. AWS Anda bertanggung jawab untuk memelihara lingkungan komputasi ini di Amazon Linux AMI setelahnya end-of-life. Sebaiknya migrasi lingkungan AWS Batch komputasi ke Amazon Linux 2023 atau Amazon Linux 2 untuk menjaga kinerja dan keamanan yang optimal.

Untuk bantuan migrasi AWS Batch dari Amazon Linux AMI ke Amazon Linux 2023 atau Amazon Linux 2, lihat [Memperbarui lingkungan komputasi](#) - AWS Batch

Strategi alokasi

Saat lingkungan komputasi terkelola dibuat, AWS Batch pilih jenis instance dari yang [instanceTypes](#) ditentukan yang paling sesuai dengan kebutuhan pekerjaan. Strategi alokasi mendefinisikan perilaku ketika AWS Batch membutuhkan kapasitas tambahan. Parameter ini tidak berlaku untuk pekerjaan yang berjalan di sumber daya Fargate. Jangan tentukan parameter ini.

BEST_FIT (default)

AWS Batch memilih jenis instance yang paling sesuai dengan kebutuhan pekerjaan dengan preferensi untuk jenis instans berbiaya terendah. Jika instance tambahan dari jenis instans yang dipilih tidak tersedia, AWS Batch tunggu instance tambahan tersedia. Jika tidak ada cukup instans yang tersedia, atau jika pengguna mencapai kuota [layanan Amazon EC2](#), maka pekerjaan tambahan tidak akan berjalan hingga pekerjaan yang sedang berjalan selesai. Strategi alokasi ini membuat biaya lebih rendah tetapi dapat membatasi penskalaan. Jika Anda menggunakan Armada SpotBEST_FIT, Peran IAM Armada Spot harus ditentukan. BEST_FIT tidak didukung saat memperbarui lingkungan komputasi. Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#).

Note

AWS Batch mengelola AWS sumber daya di akun Anda. Hitung lingkungan dengan strategi alokasi BEST_FIT yang awalnya menggunakan konfigurasi peluncuran secara default. Namun, penggunaan konfigurasi peluncuran dengan AWS akun baru akan dibatasi seiring waktu. Oleh karena itu, mulai akhir April 2024, lingkungan komputasi BEST_FIT yang baru dibuat akan default untuk meluncurkan templat. Jika peran layanan Anda tidak memiliki izin untuk mengelola template peluncuran, AWS Batch dapat terus menggunakan konfigurasi peluncuran. Lingkungan komputasi yang ada akan terus menggunakan konfigurasi peluncuran.

BEST_FIT_PROGRESSIVE

AWS Batch memilih jenis instance tambahan yang cukup besar untuk memenuhi persyaratan pekerjaan dalam antrian. Jenis instans dengan biaya lebih rendah untuk setiap unit vCPU lebih

disukai. Jika instans tambahan dari tipe instans yang dipilih sebelumnya tidak tersedia, AWS Batch akan memilih tipe instans baru.

SPOT_CAPACITY_OPTIMIZED

AWS Batch memilih satu atau lebih jenis contoh yang cukup besar untuk memenuhi persyaratan pekerjaan dalam antrian. Jenis instans yang cenderung tidak terputus lebih disukai. Strategi alokasi ini hanya tersedia untuk sumber daya komputasi Instans Spot.

SPOT_PRICE_CAPACITY_OPTIMIZED

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. Strategi alokasi ini hanya tersedia untuk sumber daya komputasi Instans Spot.

Note

Kami menyarankan Anda menggunakan SPOT_PRICE_CAPACITY_OPTIMIZED daripada SPOT_CAPACITY_OPTIMIZED dalam kebanyakan kasus.

BEST_FIT Strategi BEST_FIT_PROGRESSIVE dan strategi menggunakan Instans On-Demand atau Spot, dan SPOT_PRICE_CAPACITY_OPTIMIZED strategi SPOT_CAPACITY_OPTIMIZED dan menggunakan Instans Spot. Namun, AWS Batch mungkin perlu melebihi `maxvCpus` untuk memenuhi persyaratan kapasitas Anda. Dalam hal ini, AWS Batch tidak pernah melebihi `maxvCpus` lebih dari satu contoh.

Memperbarui lingkungan komputasi

Setelah membuat lingkungan komputasi yang menggunakan sumber daya EC2, Anda dapat memperbarui banyak pengaturan lingkungan komputasi secara langsung. Namun, mengubah beberapa pengaturan mengharuskan AWS Batch penggantian instance di lingkungan komputasi.

Untuk lingkungan komputasi yang menggunakan sumber daya Fargate, Anda dapat memperbarui yang berikut ini.

- `securityGroupIds`
- `subnets`

- `desiredvCpus`
- `maxvCpus`
- `minvCpus`

AWS Batch memiliki dua mekanisme pembaruan. Yang pertama adalah pembaruan penskalaan di mana instance ditambahkan atau dihapus dari lingkungan komputasi. Yang kedua adalah pembaruan infrastruktur di mana instance di lingkungan komputasi diganti. Pembaruan infrastruktur membutuhkan waktu lebih lama daripada pembaruan penskalaan.

Jika Anda memperbarui lingkungan komputasi dengan AWS Batch, mengubah hanya setelan ini akan menyebabkan pembaruan penskalaan: `vCPU ()` yang diinginkan, `vCPU maksimum ()` (`desiredvCpus`, `vCPU minimum (maxvCpus)`), peran layanan (`serviceRole`), dan status (`state`).

Note

Saat Anda memperbarui `desiredvCpus` pengaturan, nilainya harus berada di antara `maxvCpus` nilai `minvCpus` dan.

Selain itu, `desiredvCpus` nilai yang diperbarui harus lebih besar dari atau sama dengan `desiredvCpus` nilai saat ini. Untuk informasi selengkapnya, lihat [the section called “Pesan galat saat Anda memperbarui `desiredvCpus` pengaturan”](#).

Jika salah satu pengaturan berikut diubah dalam tindakan [UpdateComputeEnvironment](#) API, AWS Batch memulai pembaruan infrastruktur. Pembaruan infrastruktur mengharuskan peran layanan disetel ke `AWSServiceRoleForBatch`(default) dan strategi alokasi adalah `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED`, atau `SPOT_PRICE_CAPACITY_OPTIMIZED`. `BEST_FIT` tidak didukung. Kecuali untuk peran layanan, semua pengaturan yang dapat diubah untuk pembaruan penskalaan juga dapat diubah untuk pembaruan infrastruktur.

Note

Kami menyarankan Anda menggunakan `SPOT_PRICE_CAPACITY_OPTIMIZED` bukan `SPOT_CAPACITY_OPTIMIZED` dalam banyak kasus.

Selama pembaruan infrastruktur, status lingkungan komputasi berubah menjadi `UPDATING`. Instans baru diluncurkan menggunakan pengaturan yang diperbarui. Pekerjaan baru dijadwalkan pada contoh baru. Pekerjaan yang sedang berjalan dikirim sesuai dengan kebijakan pembaruan infrastruktur. Untuk informasi selengkapnya, lihat [UpdateComputeEnvironment](#) dan [UpdatePolicy](#) di Referensi AWS Batch API.

Dalam tipe `UpdatePolicy` data, pertimbangkan skenario berikut:

Note

Dalam skenario ini, berikut ini benar. Ketika sebuah instance dihentikan, pekerjaan yang sedang berjalan dihentikan. Secara default, pekerjaan ini tidak dicoba lagi. Untuk mencoba kembali salah satu pekerjaan ini setelah instance dihentikan, konfigurasi strategi coba ulang pekerjaan. Untuk informasi selengkapnya, lihat [the section called “Percobaan kembali tugas secara otomatis”](#) di AWS Batch Panduan Pengguna.

- Jika `terminateJobsOnUpdate` pengaturan disetel ke `true`, pekerjaan yang sedang berjalan dihentikan selama pembaruan infrastruktur. `jobExecutionTimeoutMinutes` pengaturan diabaikan.
- Jika `terminateJobsOnUpdate` pengaturan disetel ke `false`, pekerjaan dapat berjalan untuk waktu tambahan setelah pembaruan infrastruktur terjadi. Waktu tambahan ini dikonfigurasi dalam `jobExecutionTimeoutMinutes` pengaturan. Secara default, `jobExecutionTimeoutMinutes` pengaturannya adalah 30 menit.

Saat kapasitas tersedia di lingkungan komputasi, instans baru diluncurkan dengan pengaturan yang diperbarui dan pekerjaan dimulai pada instans baru. Karena semua pekerjaan selesai pada instance dengan pengaturan lama, instance lama dihentikan. Kapasitas apa yang tersedia berarti bahwa jumlah vCPU yang diinginkan berada di bawah jumlah maksimum vCPU dengan setidaknya sebanyak vCPU seperti yang dipersyaratkan oleh jenis instans terkecil.

Pembaruan infrastruktur

Pembaruan infrastruktur diperlukan untuk mengubah beberapa pengaturan untuk lingkungan komputasi. Jika salah satu pengaturan berikut diubah, pembaruan infrastruktur dimulai:

⚠ Important

Lingkungan komputasi harus menggunakan peran `AWSServiceRoleForBatch` terkait layanan untuk membuat perubahan yang memerlukan pembaruan infrastruktur.

Jika lingkungan komputasi menggunakan peran terkait layanan, itu tidak dapat diubah untuk menggunakan peran IAM biasa. Demikian juga, jika lingkungan komputasi memiliki peran IAM reguler, itu tidak dapat diubah untuk menggunakan peran terkait layanan. Oleh karena itu, Anda hanya dapat melakukan pembaruan infrastruktur pada lingkungan komputasi yang dibuat menggunakan peran terkait layanan.

- Strategi alokasi (`allocationStrategy`, harus salah satu `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED`, atau `SPOT_PRICE_CAPACITY_OPTIMIZED`. Jika strategi alokasi asli adalah `BEST_FIT`, pembaruan infrastruktur tidak didukung.)

ℹ Note

Kami menyarankan Anda menggunakan `SPOT_PRICE_CAPACITY_OPTIMIZED` bukan `SPOT_CAPACITY_OPTIMIZED` dalam banyak kasus.

- Persentase penawaran (`bidPercentage`)
- Konfigurasi EC2 () `ec2Configuration`
- Pasangan kunci (`ec2KeyPair`)
- ID Gambar (`imageId`)
- Peran contoh (`instanceRole`)
- Jenis contoh (`instanceTypes`)
- Luncurkan template (`launchTemplate`)
- Grup penempatan (`placementGroup`)
- Grup keamanan (`securityGroupIds`)
- Subnet VPC () `subnets`
- Tag EC2 () `tags`
- Jenis lingkungan komputasi (`type`, bisa salah satu dari EC2 atau SPOT)
- Apakah akan memperbarui ke AMI terbaru yang didukung oleh AWS Batch selama pembaruan infrastruktur `updateToLatestImageVersion`

Memperbarui ID AMI

Selama pembaruan infrastruktur, ID AMI lingkungan komputasi mungkin berubah, tergantung pada apakah AMI ditentukan dalam salah satu dari tiga pengaturan ini. AMI ditentukan dalam `imageId` (`incomputeResources`), `imageIdOverride` (`in ec2Configuration`), atau template peluncuran yang ditentukan dalam `launchTemplate`. Misalkan tidak ada ID AMI yang ditentukan dalam pengaturan tersebut dan `updateToLatestImageVersion` pengaturannya `true`. Kemudian, AMI Amazon ECS terbaru yang dioptimalkan didukung oleh AWS Batch digunakan untuk pembaruan infrastruktur apa pun.

Jika ID AMI ditentukan dalam setidaknya satu pengaturan ini, pembaruan bergantung pada pengaturan mana yang diberikan ID AMI yang digunakan sebelum pembaruan. Saat Anda membuat lingkungan komputasi, prioritas untuk memilih ID AMI adalah pertama template peluncuran, lalu `imageId` pengaturan, dan terakhir `imageIdOverride` pengaturan. Namun, jika ID AMI yang digunakan berasal dari template peluncuran, memperbarui `imageId` atau `imageIdOverride` pengaturan tidak akan memperbarui ID AMI. Satu-satunya cara untuk memperbarui ID AMI yang dipilih dari template peluncuran adalah dengan memperbarui template peluncuran. Jika parameter versi template peluncuran adalah `$Default` atau `$Latest`, versi default atau terbaru dari template peluncuran yang ditentukan dievaluasi. Jika ID AMI yang berbeda dipilih secara default atau versi terbaru dari template peluncuran dipilih, ID AMI tersebut digunakan dalam pembaruan.

Jika template peluncuran tidak digunakan untuk memilih ID AMI, ID AMI yang ditentukan dalam `imageIdOverride` parameter `imageId` atau akan digunakan. Jika keduanya ditentukan, ID AMI yang ditentukan dalam `imageIdOverride` parameter digunakan.

Misalkan lingkungan komputasi menggunakan ID AMI yang ditentukan oleh `imageId`, atau `launchTemplate` parameter `imageIdOverride`, dan Anda ingin menggunakan AMI Amazon ECS terbaru yang dioptimalkan didukung oleh AWS Batch. Kemudian, pembaruan harus menghapus pengaturan yang menyediakan ID AMI. Untuk `imageId`, ini membutuhkan menentukan string kosong untuk parameter itu. Untuk `imageIdOverride`, ini membutuhkan menentukan string kosong untuk `ec2Configuration` parameter.

Jika ID AMI berasal dari template peluncuran, Anda dapat mengubah ke AMI Amazon ECS terbaru yang dioptimalkan yang didukung AWS Batch oleh salah satu cara berikut:

- Hapus template peluncuran dengan menentukan string kosong untuk `launchTemplateName` parameter `launchTemplateId` or. Ini menghapus seluruh template peluncuran, bukan ID AMI saja.

- Jika versi terbaru dari template peluncuran tidak menentukan ID AMI, `updateToLatestImageVersion` parameter harus disetel ke `true`.

Lingkungan komputasi Amazon EKS

[Memulai dengan AWS Batch di Amazon EKS](#) menyediakan panduan singkat untuk membuat lingkungan komputasi EKS. Bagian ini memberikan detail lebih lanjut tentang lingkungan komputasi Amazon EKS.

Topik

- [Pilihan AMI default](#)
- [Versi Kubernetes yang didukung](#)
- [Memperbarui Kubernetes versi lingkungan komputasi](#)
- [Tanggung jawab bersama dari Kubernetes node](#)
- [Menjalankan DaemonSet pada node AWS Batch terkelola](#)
- [Menyesuaikan dengan template peluncuran](#)

Pilihan AMI default

Saat membuat lingkungan komputasi Amazon EKS, Anda tidak perlu menentukan Amazon Machine Image (AMI). AWS Batch memilih AMI Amazon EKS yang dioptimalkan berdasarkan Kubernetes versi dan jenis instans yang ditentukan dalam [CreateComputeEnvironment](#) permintaan Anda. Secara umum, kami menyarankan Anda menggunakan pilihan AMI default. Untuk informasi selengkapnya tentang AMI yang dioptimalkan [Amazon EKS, lihat AMI Amazon Linux Amazon EKS yang dioptimalkan](#) di Panduan Pengguna Amazon EKS.

Jalankan perintah berikut untuk melihat tipe AMI mana yang AWS Batch dipilih untuk lingkungan komputasi Amazon EKS Anda. Contoh berikut ini adalah jenis instance non-GPU.

```
# compute CE example: indicates Batch has chosen the AL2 x86 or ARM EKS 1.29 AMI,
depending on instance types
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1 \
  | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2",
```



```
    "imageKubernetesVersion": "1.29"
  }
]
```

Contoh berikut ini adalah jenis instance GPU.

```
# GPU CE example: indicates Batch has chosen the AL2 x86 EKS Accelerated 1.29 AMI
$ aws batch describe-compute-environments --compute-environments My-Eks-GPU-CE \
  | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2_NVIDIA",
    "imageKubernetesVersion": "1.29"
  }
]
```

Versi Kubernetes yang didukung

AWS Batch di Amazon EKS saat ini mendukung Kubernetes versi berikut:

- 1.29
- 1.28
- 1.27
- 1.26
- 1.25
- 1.24
- 1.23

Anda mungkin melihat pesan galat yang menyerupai berikut ini saat menggunakan operasi `CreateComputeEnvironment` API atau operasi `UpdateComputeEnvironment` API untuk membuat atau memperbarui lingkungan komputasi. Masalah ini terjadi jika Anda menentukan Kubernetes versi yang tidak didukung di `EC2Configuration`.

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Untuk mengatasi masalah ini, hapus lingkungan komputasi lalu buat ulang dengan versi yang didukung Kubernetes.

Anda dapat melakukan upgrade versi minor pada kluster Amazon EKS Anda. Misalnya, Anda dapat memutakhirkan cluster dari 1.xx ke 1.yy bahkan jika versi minor tidak didukung.

Namun, status lingkungan komputasi mungkin berubah menjadi INVALID setelah pembaruan versi utama. Misalnya, jika Anda melakukan upgrade versi utama dari 1.xx ke 2.yy. Jika versi mayor tidak didukung oleh AWS Batch, Anda akan melihat pesan galat yang menyerupai berikut ini.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Memperbarui Kubernetes versi lingkungan komputasi

Dengan AWS Batch, Anda dapat memperbarui Kubernetes versi lingkungan komputasi untuk mendukung peningkatan kluster Amazon EKS. KubernetesVersi lingkungan komputasi adalah versi Amazon EKS AMI untuk Kubernetes node yang AWS Batch diluncurkan untuk menjalankan pekerjaan. Anda dapat melakukan pemutakhiran Kubernetes versi pada node Amazon EKS sebelum atau setelah memperbarui versi bidang kontrol kluster Amazon EKS. Kami menyarankan Anda memperbarui node setelah memutakhirkan bidang kontrol. Untuk informasi selengkapnya, lihat [Memperbarui Kubernetes versi kluster Amazon EKS](#) di Panduan Pengguna Amazon EKS.

Untuk memutakhirkan Kubernetes versi lingkungan komputasi, gunakan operasi [UpdateComputeEnvironmentAPI](#).

```
$ aws batch update-compute-environment \  
  --compute-environment <compute-environment-name> \  
  --compute-resources \  
    'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.23}]'
```

Tanggung jawab bersama dari Kubernetes node

Pemeliharaan lingkungan komputasi adalah tanggung jawab bersama.

- Jangan mengubah atau menghapus AWS Batch node, label, taint, ruang nama, templat peluncuran, atau grup penskalaan otomatis. Jangan menambahkan taints ke node AWS Batch terkelola. Jika Anda membuat salah satu dari perubahan ini, lingkungan komputasi Anda tidak dapat didukung dan kegagalan termasuk instans idle terjadi.
- Jangan menargetkan pod Anda ke node AWS Batch terkelola. Jika Anda menargetkan pod Anda ke node terkelola, penskalaan yang rusak dan antrian pekerjaan macet akan terjadi. Jalankan beban kerja yang tidak digunakan AWS Batch pada node yang dikelola sendiri atau grup node

terkelola. Untuk informasi selengkapnya, lihat [Grup node terkelola](#) di Panduan Pengguna Amazon EKS.

- Anda dapat menargetkan DaemonSet untuk dijalankan pada node AWS Batch terkelola. Untuk informasi selengkapnya, lihat [Menjalankan DaemonSet pada node AWS Batch terkelola](#).

AWS Batch tidak secara otomatis memperbarui AMI lingkungan komputasi. Anda bertanggung jawab untuk memperbaruinya. Jalankan perintah berikut untuk memperbarui AMI Anda ke versi AMI terbaru.

```
$ aws batch update-compute-environment \  
  --compute-environment <compute-environment-name> \  
  --compute-resources 'updateToLatestImageVersion=true'
```

AWS Batch tidak secara otomatis memutakhirkan Kubernetes versi. Jalankan perintah berikut untuk memperbarui Kubernetes versi lingkungan komputer Anda ke **1.23**.

```
$ aws batch update-compute-environment \  
  --compute-environment <compute-environment-name> \  
  --compute-resources \  
    'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.23}]'
```

Saat memperbarui ke AMI atau Kubernetes versi yang lebih baru, Anda dapat menentukan apakah akan menghentikan pekerjaan saat diperbarui (`terminateJobsOnUpdate`) dan berapa lama menunggu sebelum instance diganti jika pekerjaan yang sedang berjalan tidak selesai (`jobExecutionTimeoutMinutes`.) Untuk informasi selengkapnya, lihat [Memperbarui lingkungan komputasi](#) dan kebijakan pembaruan infrastruktur ([UpdatePolicy](#)) yang disetel dalam operasi [UpdateComputeEnvironmentAPI](#).

Menjalankan DaemonSet pada node AWS Batch terkelola

AWS Batch menetapkan taints pada Kubernetes node AWS Batch terkelola. Anda dapat menargetkan DaemonSet untuk dijalankan pada node AWS Batch terkelola dengan yang berikut `initolerations`.

```
tolerations:  
- key: "batch.amazonaws.com/batch-node"  
  operator: "Exists"
```

Cara lain untuk melakukan ini adalah dengan yang berikut `initolerations`.

```
tolerations:  
- key: "batch.amazonaws.com/batch-node"  
  operator: "Exists"  
  effect: "NoSchedule"  
- key: "batch.amazonaws.com/batch-node"  
  operator: "Exists"  
  effect: "NoExecute"
```

Menyesuaikan dengan template peluncuran

AWS Batch di Amazon EKS mendukung template peluncuran. Ada kendala pada apa yang dapat dilakukan template peluncuran Anda.

Important

AWS Batch berjalan `/etc/eks/bootstrap.sh`. Jangan berjalan `/etc/eks/bootstrap.sh` di template peluncuran atau cloud-init user-data skrip Anda. Anda dapat menambahkan parameter tambahan selain `--kubelet-extra-args` parameter ke [bootstrap.sh](#). Untuk melakukan ini, atur `AWS_BATCH_KUBELET_EXTRA_ARGS` variabel dalam `/etc/aws-batch/batch.config` file. Lihat contoh berikut untuk detailnya.

Note

Jika template peluncuran diubah setelah [CreateComputeEnvironment](#) dipanggil, [UpdateComputeEnvironment](#) harus dipanggil untuk mengevaluasi versi template peluncuran untuk penggantian.

Topik

- [Menambahkan argumen kubelet tambahan](#)
- [Mengkonfigurasi runtime kontainer](#)
- [Memasang volume Amazon EFS](#)
- [Dukungan IPv6](#)

Menambahkan argumen `kubelet` tambahan

AWS Batch mendukung penambahan argumen tambahan ke `kubelet` perintah. Untuk daftar parameter yang didukung, lihat [kubelet](#) di Kubernetes dokumentasi. Dalam contoh berikut, `--node-labels mylabel=helloworld` ditambahkan ke baris `kubelet` perintah.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
mkdir -p /etc/aws-batch

echo AWS_BATCH_KUBELET_EXTRA_ARGS="\--node-labels mylabel=helloworld\" >> /etc/aws-batch/batch.config

--==MYBOUNDARY==--
```

Mengkonfigurasi runtime kontainer

Anda dapat menggunakan variabel AWS Batch `CONTAINER_RUNTIME` lingkungan untuk mengonfigurasi runtime kontainer pada node terkelola. Contoh berikut menetapkan runtime kontainer ke `containerd` saat `bootstrap.sh` berjalan. Untuk informasi lebih lanjut, lihat [containerd](#) di Kubernetes dokumentasi.

Note

Variabel `CONTAINER_RUNTIME` lingkungan setara dengan `--container-runtime opsibootstrap.sh`. Untuk informasi lebih lanjut, lihat [Options](#) di Kubernetes dokumentasi.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
mkdir -p /etc/aws-batch
```

```
echo CONTAINER_RUNTIME=containerd >> /etc/aws-batch/batch.config

--==MYBOUNDARY==--
```

Memasang volume Amazon EFS

Anda dapat menggunakan template peluncuran untuk memasang volume ke node. Dalam contoh berikut, `runcmd` pengaturan `cloud-config` packages dan digunakan. Untuk informasi selengkapnya, lihat [contoh konfigurasi Cloud](#) dalam `cloud-init` dokumentasi.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs _netdev,noresvport,tls,iam 0 0"
  >> /etc/fstab
- mount -t efs -o tls ${file_system_id_01}:/ ${efs_directory}

--==MYBOUNDARY==--
```

Untuk menggunakan volume ini dalam pekerjaan, itu harus ditambahkan dalam parameter [eksProperties](#) ke [RegisterJobDefinition](#). Contoh berikut adalah sebagian besar definisi pekerjaan.

```
{
  "jobDefinitionName": "MyJob0nEks_EFS",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
```

```
    "command": ["ls", "-la", "/efs"],
    "resources": {
      "limits": {
        "cpu": "1",
        "memory": "1024Mi"
      }
    },
    "volumeMounts": [
      {
        "name": "efs-volume",
        "mountPath": "/efs"
      }
    ]
  },
  "volumes": [
    {
      "name": "efs-volume",
      "hostPath": {
        "path": "/mnt/efs"
      }
    }
  ]
}
```

Di node, volume Amazon EFS dipasang di `/mnt/efs` direktori. Dalam wadah untuk pekerjaan Amazon EKS, volume dipasang di `/efs` direktori.

Dukungan IPv6

AWS Batch mendukung kluster Amazon EKS yang memiliki alamat IPv6. Tidak ada penyesuaian yang diperlukan untuk AWS Batch dukungan. Namun, sebelum memulai, sebaiknya Anda meninjau pertimbangan dan ketentuan yang diuraikan dalam [Menetapkan alamat IPv6 ke pod dan layanan di Panduan Pengguna Amazon EKS](#).

Manajemen Memori Sumber Daya Komputasi

Saat agen penampung Amazon ECS mendaftarkan sumber daya komputasi ke lingkungan komputasi, agen harus menentukan berapa banyak memori yang tersedia sumber daya komputasi untuk cadangan pekerjaan Anda. Karena overhead memori platform dan memori yang ditempati

oleh kernel sistem, jumlah ini berbeda dari jumlah memori yang diinstal untuk instans Amazon EC2. Contohnya, instans `m4.large` memiliki memori terinstal sebesar 8 GiB. Namun, ini tidak selalu diterjemahkan ke 8192 MiB memori yang tersedia untuk pekerjaan saat sumber daya komputasi mendaftar.

Misalkan Anda menentukan 8192 MiB untuk pekerjaan itu, dan tidak ada sumber daya komputasi Anda yang memiliki 8192 MiB atau lebih besar memori yang tersedia untuk memenuhi persyaratan ini. Kemudian, pekerjaan tidak dapat ditempatkan di lingkungan komputasi Anda. Jika Anda menggunakan lingkungan komputasi terkelola, AWS Batch harus meluncurkan jenis instans yang lebih besar untuk mengakomodasi permintaan.

AMI sumber daya komputasi AWS Batch default juga menyimpan 32 MiB memori untuk agen kontainer Amazon ECS dan proses sistem penting lainnya. Memori ini tidak tersedia untuk alokasi pekerjaan. Untuk informasi selengkapnya, lihat [Menyimpan Memori Sistem](#).

Agan kontainer Amazon ECS menggunakan fungsi `ReadMemInfo()` Docker untuk kueri total memori yang tersedia untuk sistem operasi. Linux menyediakan utilitas baris perintah untuk menentukan total memori.

Example - Menentukan memori total Linux

`free` Perintah mengembalikan memori total yang dikenali oleh sistem operasi.

```
$ free -b
```

Berikut ini adalah contoh keluaran untuk `m4.large` instance yang menjalankan Amazon Linux AMI yang dioptimalkan Amazon ECS.

```

                total        used        free      shared    buffers     cached
Mem:      8373026816  348180480  8024846336       90112    25534464    205418496
-/+ buffers/cache:  117227520  8255799296
```

Instance ini memiliki 8373026816 byte dari total memori. Ini berarti ada 7985 MiB yang tersedia untuk tugas.

Menyimpan Memori Sistem

Jika Anda menempati semua memori pada sumber daya komputasi dengan pekerjaan Anda, ada kemungkinan bahwa pekerjaan Anda bersaing dengan proses sistem penting untuk memori dan mungkin menyebabkan kegagalan sistem. Agen kontainer Amazon ECS menyediakan variabel

konfigurasi yang disebut `ECS_RESERVED_MEMORY`. Anda dapat menggunakan variabel konfigurasi ini untuk menghapus sejumlah MiB memori tertentu dari kumpulan yang dialokasikan untuk pekerjaan Anda. Ini secara efektif menyimpan memori untuk proses sistem yang penting.

AMI sumber daya komputasi AWS Batch default menyimpan 32 MiB memori untuk agen kontainer Amazon ECS dan proses sistem penting lainnya.

Melihat Memori Sumber Daya Komputasi

Anda dapat melihat berapa banyak memori yang didaftarkan oleh sumber daya komputasi di konsol Amazon ECS atau dengan operasi API. [DescribeContainerInstances](#) Jika Anda mencoba memaksimalkan pemanfaatan sumber daya Anda dengan menyediakan pekerjaan Anda sebanyak mungkin memori untuk jenis instance tertentu, Anda dapat mengamati memori yang tersedia untuk sumber daya komputasi itu dan kemudian menetapkan pekerjaan Anda sebanyak itu memori.

Untuk melihat memori sumber daya komputasi

1. Buka konsol di <https://console.aws.amazon.com/ecs/v2>.
2. Pilih Cluster, lalu pilih klaster yang menghosting sumber daya komputasi Anda untuk dilihat.

Nama klaster untuk lingkungan komputasi Anda dimulai dengan nama lingkungan komputasi Anda.

3. Pilih Infrastruktur.
4. Di bawah contoh Container, pilih instance container.
5. Bagian Sumber Daya dan jaringan menunjukkan memori terdaftar dan tersedia untuk sumber daya komputasi.

Nilai memori Terdaftar adalah sumber daya komputasi yang terdaftar di Amazon ECS saat pertama kali diluncurkan, dan nilai memori yang tersedia adalah apa yang belum dialokasikan untuk pekerjaan.

Pertimbangan memori dan vCPU untuk Amazon AWS Batch EKS

AWS Batch Di Amazon EKS, Anda dapat menentukan sumber daya yang tersedia untuk wadah. Misalnya, Anda dapat menentukan `requests` atau `limits` nilai untuk vCPU dan sumber daya memori.

Berikut ini adalah kendala untuk menentukan sumber daya vCPU:

- Setidaknya satu `vCPU requests` atau `limits` nilai harus ditentukan.
- Satu unit `vCPU` setara dengan satu inti fisik atau virtual.
- Nilai `vCPU` harus dimasukkan dalam bilangan bulat atau dengan penambahan 0,25.
- Nilai `vCPU` valid terkecil adalah 0,25.
- Jika keduanya ditentukan, `requests` nilainya harus kurang dari atau sama dengan `limits` nilainya. Dengan cara ini, Anda dapat mengonfigurasi konfigurasi `vCPU` lunak dan keras.
- Nilai `vCPU` tidak dapat ditentukan dalam bentuk `milliCPU`. Misalnya, `100m` bukan nilai yang valid.
- AWS Batch menggunakan `requests` nilai untuk keputusan penskalaan. Jika `requests` nilai tidak ditentukan, `limits` nilai disalin ke `requests` nilai.

Berikut ini adalah kendala untuk menentukan sumber daya memori:

- Setidaknya satu memori `requests` atau `limits` nilai harus ditentukan.
- Nilai memori harus dalam mebibytes (MiBs).
- Jika keduanya ditentukan, `requests` nilainya harus sama dengan `limits` nilainya.
- AWS Batch menggunakan `requests` nilai untuk keputusan penskalaan. Jika `requests` nilai tidak ditentukan, `limits` nilai disalin ke `requests` nilai.

Berikut ini adalah kendala untuk menentukan sumber daya GPU:

- Jika keduanya ditentukan, `requests` nilainya harus sama dengan `limits` nilainya.
- AWS Batch menggunakan `requests` nilai untuk keputusan penskalaan. Jika `requests` nilai tidak ditentukan, `limits` nilai disalin ke `requests` nilai.

Contoh ketentuan tugas

Berikut ini AWS Batch pada definisi pekerjaan Amazon EKS mengonfigurasi pembagian `vCPU` lunak. Ini memungkinkan AWS Batch di Amazon EKS menggunakan semua kapasitas `vCPU` untuk jenis instans. Namun, jika ada pekerjaan lain yang berjalan, pekerjaan tersebut dialokasikan maksimal 2 `vCPU`. Memori dibatasi hingga 2 GB.

```
{
  "jobDefinitionName": "MyJob0nEks_Sleep",
  "type": "container",
  "eksProperties": {
```

```

    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["sleep", "60"],
          "resources": {
            "requests": {
              "cpu": "2",
              "memory": "2048Mi"
            }
          }
        }
      ]
    }
  }
}

```

Berikut ini AWS Batch di Amazon EKS definisi pekerjaan memiliki request nilai 1 dan mengalokasikan maksimum 4 vCPU untuk pekerjaan.

```

{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["sleep", "60"],
          "resources": {
            "requests": {
              "cpu": "1"
            },
            "limits": {
              "cpu": "4",
              "memory": "2048Mi"
            }
          }
        }
      ]
    }
  }
}

```

Berikut ini AWS Batch di Amazon EKS definisi pekerjaan menetapkan `limits` nilai vCPU 1 dan `limits` nilai memori 1 GB.

```
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["sleep", "60"],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ]
    }
  }
}
```

Saat AWS Batch menerjemahkan pekerjaan AWS Batch di Amazon EKS ke dalam pod Amazon EKS, AWS Batch salin `limits` nilainya ke `requests` nilainya. Ini jika `requests` nilai tidak ditentukan. Saat Anda mengirimkan contoh definisi pekerjaan sebelumnya, pod spec adalah sebagai berikut.

```
apiVersion: v1
kind: Pod
...
spec:
  ...
  containers:
    - command:
      - sleep
      - 60
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      resources:
        limits:
          cpu: 1
          memory: 1024Mi
```

```

requests:
  cpu: 1
  memory: 1024Mi
  ...

```

Node CPU dan reservasi memori

AWS Batch bergantung pada logika default `bootstrap.sh` file untuk vCPU dan reservasi memori. Untuk informasi selengkapnya tentang `bootstrap.sh` file, lihat [bootstrap.sh](#). Saat Anda mengukur vCPU dan sumber daya memori Anda, pertimbangkan contoh-contoh berikut.

Note

Jika tidak ada instance yang berjalan, vCPU dan reservasi memori pada awalnya dapat AWS Batch memengaruhi logika penskalaan dan pengambilan keputusan. Setelah instance berjalan, AWS Batch sesuaikan alokasi awal.

Contoh reservasi CPU node

Nilai reservasi CPU dihitung dalam millicores menggunakan jumlah total vCPU yang tersedia untuk instance.

Nomor vCPU	Persentase dicadangkan
1	6%
2	1%
3-4	0,5%
4 dan di atas	0,25%

Menggunakan nilai-nilai sebelumnya, berikut ini benar:

- Nilai reservasi CPU untuk `c5.large` instance dengan 2 vCPU adalah 70 m. Ini dihitung dengan cara berikut: $(1 \cdot 60) + (1 \cdot 10) = 70$ m.
- Nilai reservasi CPU untuk `c5.24xlarge` instance dengan 96 vCPU adalah 310 m. Ini dihitung dengan cara berikut: $(1 \cdot 60) + (1 \cdot 10) + (2 \cdot 5) + (92 \cdot 2.5) = 310$ m.

Dalam contoh ini, ada 1930 (dihitung 2000-70) unit vCPU millicore yang tersedia untuk menjalankan pekerjaan pada sebuah instance. `c5.large` Misalkan pekerjaan Anda membutuhkan 2 (2* 1000 m) unit vCPU, pekerjaan tidak cocok pada satu instance. `c5.large` Namun, pekerjaan yang membutuhkan unit 1.75 vCPU cocok.

Contoh reservasi memori node

Nilai reservasi memori dihitung dalam mebibytes menggunakan yang berikut ini:

- Kapasitas instance dalam mebibytes. Misalnya, instance 8 GB adalah 7.748MiB.
- `kubeReservedNilainya`. `kubeReservedNilainya` adalah jumlah memori yang akan dicadangkan untuk daemon sistem. `kubeReservedNilainya` dihitung dengan cara berikut: $((11 * \text{jumlah maksimum pod yang didukung oleh tipe instance}) + 255)$. Untuk informasi tentang jumlah maksimum pod yang didukung oleh tipe instance, lihat [eni-max-pods.txt](#)
- `HardEvictionLimitNilainya`. Ketika memori yang tersedia jatuh di bawah `HardEvictionLimit` nilai, instance mencoba untuk mengusir pod.

*Rumus untuk menghitung memori yang dapat dialokasikan adalah sebagai berikut: $(Instance_capacity_in_mib) - (11 * (maximum_number_of_pods)) - 255 - (nilai. HardEvictionLimit)$.*

Sebuah `c5.large` instance mendukung hingga 29 pod. Untuk `c5.large` instance 8 GB dengan `HardEvictionLimit` nilai 100 MiB, memori yang dapat dialokasikan adalah 7074. MiB Ini dihitung dengan cara berikut: $(7748 - (11 * 29) - 255 - 100) = 7074$ MiB. Dalam contoh ini, 8.192 MiB pekerjaan tidak cocok dengan instance ini meskipun itu adalah instance 8 gibibyte (GiB).

DaemonSets

Saat Anda menggunakan `DaemonSets`, pertimbangkan hal berikut:

- Jika tidak ada instans AWS Batch Amazon EKS yang berjalan, awalnya `DaemonSets` dapat memengaruhi logika AWS Batch penskalaan dan pengambilan keputusan. AWS Batch awalnya mengalokasikan 0,5 unit vCPU dan 500 MiB untuk yang diharapkan. `DaemonSets` Setelah instance berjalan, AWS Batch sesuaikan alokasi awal.
- Jika a `DaemonSet` mendefinisikan vCPU atau batas memori, AWS Batch di Amazon EKS pekerjaan memiliki lebih sedikit sumber daya. Kami menyarankan Anda menjaga jumlah `DaemonSets` yang ditugaskan untuk AWS Batch pekerjaan serendah mungkin.

Kebijakan penjadwalan

Anda dapat menggunakan kebijakan penjadwalan untuk mengonfigurasi cara sumber daya komputasi dalam antrian pekerjaan dialokasikan antara pengguna atau beban kerja. Dengan menggunakan kebijakan penjadwalan, Anda dapat menetapkan pengenal berbagi wajar yang berbeda untuk beban kerja atau pengguna. AWS Batch memberikan setiap pengidentifikasi saham yang adil persentase dari total sumber daya yang tersedia selama periode waktu tertentu.

Persentase saham wajar dihitung menggunakan `shareDistribution` nilai `shareDecaySeconds` dan. Anda dapat menambahkan waktu ke analisis saham yang adil dengan menetapkan waktu pembusukan saham ke kebijakan. Menambahkan waktu memberi lebih banyak bobot pada waktu dan lebih sedikit pada bobot yang ditentukan. Anda dapat menyimpan sumber daya komputasi sebagai cadangan untuk pengidentifikasi saham wajar yang tidak aktif dengan menentukan reservasi komputasi. Untuk informasi selengkapnya, lihat [Parameter kebijakan penjadwalan](#).

Topik

- [Membuat kebijakan penjadwalan](#)
- [Parameter kebijakan penjadwalan](#)

Membuat kebijakan penjadwalan

Sebelum Anda dapat membuat antrian pekerjaan dengan kebijakan penjadwalan, Anda harus membuat kebijakan penjadwalan. Saat membuat kebijakan penjadwalan, Anda mengaitkan satu atau beberapa pengidentifikasi saham wajar atau awalan pengenal saham wajar dengan bobot untuk antrian dan secara opsional menetapkan periode peluruhan dan menghitung reservasi ke kebijakan tersebut.

Untuk membuat kebijakan penjadwalan

1. Buka konsol AWS Batch di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi, pilih Wilayah untuk digunakan.
3. Di panel navigasi, pilih Kebijakan penjadwalan, Buat.
4. Untuk Nama, masukkan nama unik untuk kebijakan penjadwalan Anda. Hingga 128 huruf (huruf besar dan huruf kecil), angka, tanda hubung, dan garis bawah diperbolehkan.
5. (Opsional) Untuk Share decay seconds, masukkan nilai integer untuk waktu peluruhan berbagi kebijakan penjadwalan. Waktu peluruhan berbagi yang lebih lama akan digunakan untuk

mempertimbangkan penggunaan sumber daya komputasi dalam waktu yang lebih lama saat menjadwalkan pekerjaan. Ini dapat memungkinkan pekerjaan yang menggunakan pengenal saham yang adil untuk sementara menggunakan lebih banyak sumber daya komputasi daripada bobot yang diizinkan oleh pengidentifikasi saham adil itu jika pengidentifikasi saham yang adil itu baru-baru ini tidak menggunakan sumber daya komputasi.

6. (Opsional) Untuk reservasi Compute, masukkan nilai integer untuk reservasi komputasi kebijakan penjadwalan. Reservasi komputasi akan menyimpan beberapa vCPU sebagai cadangan untuk digunakan untuk pengidentifikasi saham wajar yang saat ini tidak aktif.

Rasio cadangan ActiveFairShares adalah $(computeReservation/100)^{ActiveFairShares}$ di mana jumlah pengidentifikasi saham adil aktif.

Misalnya, `computeReservation` nilai 50 menunjukkan bahwa AWS Batch harus mencadangkan 50% dari VCPU maksimum yang tersedia jika hanya ada satu pengidentifikasi saham yang adil, 25% jika ada dua pengidentifikasi saham yang adil, dan 12,5% jika ada tiga pengidentifikasi saham yang adil. `computeReservation` Nilai 25 menunjukkan bahwa AWS Batch harus mencadangkan 25% dari VCPU maksimum yang tersedia jika hanya ada satu pengidentifikasi saham yang adil, 6,25% jika ada dua pengidentifikasi saham yang adil, dan 1,56% jika ada tiga pengidentifikasi saham yang adil.

7. Di bagian `Bagikan` atribut, Anda dapat menentukan pengenal saham wajar dan bobot untuk setiap pengidentifikasi saham wajar untuk dikaitkan dengan kebijakan penjadwalan.
 - a. Pilih `Tambahkan` pengenal berbagi.
 - b. Untuk pengenal `Berbagi`, tentukan pengenal saham wajar. Jika string diakhiri dengan `*`, ini menjadi awalan pengidentifikasi saham adil yang digunakan untuk mencocokkan pengidentifikasi saham yang adil untuk pekerjaan. Semua pengidentifikasi saham adil dan awalan pengenal saham adil dalam kebijakan penjadwalan harus unik dan tidak dapat tumpang tindih. Misalnya, Anda tidak dapat memiliki awalan pengenal saham yang adil `'userA*'` dan pengidentifikasi berbagi adil `'userA1'` dalam kebijakan penjadwalan yang sama.
 - c. Untuk faktor Berat, tentukan bobot relatif untuk pengidentifikasi saham wajar. Nilai defaultnya adalah 1.0. Nilai yang lebih rendah memiliki prioritas yang lebih tinggi untuk sumber daya komputasi. Jika awalan pengenal saham yang adil digunakan, pekerjaan dengan pengidentifikasi saham adil yang dimulai dengan awalan akan berbagi faktor bobot. Ini secara efektif meningkatkan faktor bobot untuk pekerjaan tersebut, menurunkan prioritas individu mereka tetapi mempertahankan faktor bobot yang sama untuk awalan pengidentifikasi saham yang adil.

8. (Opsional) Di tag bagian, Anda dapat menentukan kunci dan nilai untuk setiap tag untuk dikaitkan dengan kebijakan penjadwalan. Untuk informasi selengkapnya, lihat [Menandai sumber daya AWS Batch Anda](#).
9. Pilih Kirim untuk menyelesaikan dan membuat kebijakan penjadwalan Anda.

Templat kebijakan penjadwalan

Templat kebijakan penjadwalan kosong ditunjukkan di bawah ini. Anda dapat menggunakan template ini untuk membuat kebijakan penjadwalan Anda yang kemudian dapat disimpan ke file dan digunakan dengan AWS CLI `--cli-input-json` opsi. Untuk informasi selengkapnya tentang parameter ini, lihat [CreateSchedulingPolicy](#) di Referensi AWS Batch API.

```
{
  "name": "",
  "fairsharePolicy": {
    "shareDecaySeconds": 0,
    "computeReservation": 0,
    "shareDistribution": [
      {
        "shareIdentifier": "",
        "weightFactor": 0.0
      }
    ]
  },
  "tags": {
    "KeyName": ""
  }
}
```

Note

Anda dapat menghasilkan templat antrian tugas sebelumnya dengan perintah AWS CLI berikut.

```
$ aws batch create-scheduling-policy --generate-cli-skeleton
```

Parameter kebijakan penjadwalan

Kebijakan penjadwalan dibagi menjadi tiga komponen dasar: nama, kebijakan pembagian yang adil, dan tag kebijakan penjadwalan.

Topik

- [Nama kebijakan penjadwalan](#)
- [Kebijakan saham yang adil](#)
- [Tag](#)

Nama kebijakan penjadwalan

name

Nama untuk kebijakan penjadwalan Anda. Hingga 128 huruf (huruf besar dan huruf kecil), angka, tanda hubung, dan garis bawah diperbolehkan.

Tipe: String

Diperlukan: Ya

Kebijakan saham yang adil

fairsharePolicy

Kebijakan pembagian yang adil dari kebijakan penjadwalan.

```
"fairsharePolicy": {
  "computeReservation": number,
  "shareDecaySeconds": number,
  "shareDistribution": [
    {
      "shareIdentifier": "string",
      "weightFactor": number
    }
  ]
}
```

Tipe: Objek

Diperlukan: Tidak

`computeReservation`

Nilai yang digunakan untuk mencadangkan beberapa VCPU maksimum yang tersedia untuk pengidentifikasi saham wajar yang belum digunakan.

Rasio cadangan `ActiveFairShares` adalah

$(\text{computeReservation}/100)^{\text{ActiveFairShares}}$ di mana jumlah pengidentifikasi saham adil aktif.

Misalnya, `computeReservation` nilai 50 menunjukkan bahwa AWS Batch harus mencadangkan 50% dari VCPU maksimum yang tersedia jika hanya ada satu pengidentifikasi saham adil aktif, 25% jika ada dua pengidentifikasi saham adil aktif, dan 12,5% jika ada tiga pengidentifikasi saham adil aktif. `computeReservation` Nilai 25 menunjukkan bahwa AWS Batch harus mencadangkan 25% dari VCPU maksimum yang tersedia jika hanya ada satu pengidentifikasi saham adil aktif, 6,25% jika ada dua pengidentifikasi saham adil aktif, dan 1,56% jika ada tiga pengidentifikasi saham adil aktif.

Tipe: Bilangan Bulat

Rentang yang valid: Nilai minimum 0. Nilai maksimum 99.

Diperlukan: Tidak

`shareDecaySeconds`

Periode waktu yang digunakan untuk menghitung persentase saham yang adil untuk setiap pengidentifikasi saham wajar yang digunakan. Nilai nol (0) menunjukkan bahwa hanya penggunaan saat ini yang harus diukur. Peluruhan memungkinkan pekerjaan yang dijalankan baru-baru ini memiliki bobot lebih dari pekerjaan yang berjalan lebih awal.

Tipe: Bilangan Bulat

Rentang yang valid: Nilai minimum 0. Nilai maksimum 604800 (1 minggu).

Diperlukan: Tidak

`shareDistribution`

Larik objek yang berisi bobot untuk pengidentifikasi saham wajar untuk kebijakan pembagian yang adil. Pengidentifikasi saham adil yang tidak disertakan memiliki bobot default sebesar.

1.0

```
"shareDistribution": [  
  {  
    "shareIdentifier": "string",  
    "weightFactor": number  
  }  
]
```

Jenis: Array

Diperlukan: Tidak

`shareIdentifier`

Pengidentifikasi saham yang adil atau awalan pengenalan saham yang adil. Jika string diakhiri dengan '*' maka string ini menentukan awalan pengidentifikasi saham yang adil untuk pengidentifikasi saham adil yang dimulai dengan awalan itu. Misalnya jika nilainya adalah `UserA*` dan 1 dan ada dua pengidentifikasi saham wajar yang dimulai dengan `UserA`, maka masing-masing pengidentifikasi saham wajar tersebut akan memiliki bobot 2; jika ada lima pengidentifikasi saham wajar tersebut, maka masing-masing akan memiliki bobot 5. `weightFactor`

Daftar pengidentifikasi saham yang adil dan awalan pengenalan saham adil dalam kebijakan saham yang adil tidak dapat tumpang tindih. Misalnya, Anda tidak dapat memiliki awalan pengenalan saham yang adil `UserA*` dan pengenalan saham yang adil `UserA-1` dalam kebijakan saham adil yang sama.

Tipe: String

Diperlukan: Ya

`weightFactor`

Faktor bobot untuk pengidentifikasi saham yang adil. Nilai defaultnya adalah 1.0. Nilai yang lebih rendah memiliki prioritas yang lebih tinggi untuk sumber daya komputasi. Misalnya, pekerjaan yang menggunakan pengidentifikasi saham dengan faktor bobot 0,125 (1/8) mendapatkan 8 kali sumber daya komputasi pekerjaan yang menggunakan pengidentifikasi saham dengan faktor bobot 1.

Nilai terkecil yang didukung adalah 0,0001 dan nilai dukungan terbesar adalah 999,9999.

Tipe: Float

Diperlukan: Tidak

Tag

tags

Tag pasangan nilai-kunci untuk dikaitkan dengan kebijakan penjadwalan. Untuk informasi selengkapnya, lihat [Menandai sumber daya AWS Batch Anda](#).

Tipe: Peta antar string

Diperlukan: Tidak

Mengorkestrasi AWS Batch pekerjaan dengan mesin status Step Functions di konsol AWS Batch

Anda dapat menggunakan AWS Batch konsol untuk melihat detail tentang mesin status Step Functions dan fungsi yang mereka gunakan.

Bagian-bagian

- [Melihat detail mesin status](#)
- [Mengedit mesin status](#)
- [Menjalankan mesin keadaan](#)

Melihat detail mesin status

AWS BatchKonsol menampilkan daftar mesin status Anda saat ini Wilayah AWS yang berisi setidaknya satu langkah alur kerja yang AWS Batch mengirimkan pekerjaan.

Pilih mesin keadaan untuk melihat representasi grafis alur kerja. Langkah-langkah yang disorot dengan warna biru mewakili AWS Batch pekerjaan. Gunakan kontrol grafik untuk memperbesar, memperkecil, dan memusatkan grafik.

Note

Saat AWS Batch pekerjaan [direferensikan secara dinamis JsonPath](#) dalam definisi mesin status, detail fungsi tidak dapat ditampilkan di AWS Batch konsol. Sebagai gantinya, nama pekerjaan terdaftar sebagai referensi Dinamis, dan langkah-langkah yang sesuai dalam grafik berwarna abu-abu.

Untuk melihat detail mesin keadaan

1. Buka AWS Batch konsol [Workflow orkestrasi didukung oleh halaman Step Functions](#).
2. Pilih mesin keadaan.

<result>

AWS BatchKonsol membuka halaman Detail.

</result>

Untuk informasi selengkapnya, lihat [Step Functions](#) dalam Panduan Developer AWS Step Functions.

Mengedit mesin status

Saat Anda ingin mengedit mesin status, AWS Batch buka halaman Edit definisi konsol Step Functions.

Untuk mengedit mesin keadaan

1. Buka AWS Batch konsol [Workflow orkestrasi didukung oleh halaman Step Functions](#).
2. Pilih mesin keadaan.
3. Pilih Edit.

Konsol Step Functions membuka halaman Edit definisi yang baru.

4. Edit mesin keadaan dan pilih Simpan.

Untuk informasi selengkapnya tentang mengedit mesin status, lihat [bahasa mesin status Step Functions](#) dalam Panduan Developer AWS Step Functions.

Menjalankan mesin keadaan

Saat Anda ingin menjalankan mesin status, AWS Batch buka halaman eksekusi baru pada konsol Step Functions.

Untuk menjalankan mesin keadaan

1. Buka AWS Batch konsol [Workflow orkestrasi didukung oleh halaman Step Functions](#).
2. Pilih mesin keadaan.
3. Pilih Jalankan.

Konsol Step Functions membuka halaman Eksekusi baru.

4. (Opsional) Edit mesin keadaan dan pilih Mulai eksekusi.

Untuk informasi selengkapnya tentang menjalankan mesin status, lihat [konsep eksekusi mesin status Step Functions](#) dalam Panduan Developer AWS Step Functions.

AWS Batch di AWS Fargate

AWS Fargate adalah teknologi yang dapat Anda gunakan dengan AWS Batch untuk menjalankan [kontainer](#) tanpa harus mengelola server atau klaster instans Amazon EC2. Dengan AWS Fargate, Anda tidak perlu menyediakan, mengonfigurasi, atau menskalakan klaster mesin virtual untuk menjalankan kontainer. Anda tidak perlu memilih jenis server, memutuskan kapan akan menskalakan klaster Anda, atau mengoptimalkan paket klaster.

Ketika Anda menjalankan tugas dengan sumber daya Fargate, Anda membuat paket aplikasi Anda dalam kontainer, menentukan persyaratan CPU dan memori, menentukan jaringan dan kebijakan IAM, dan meluncurkan aplikasi. Setiap tugas Fargate memiliki batas isolasi tersendiri dan tidak berbagi inti kernel, sumber daya CPU, sumber daya memori, atau antarmuka jaringan elastis dengan tugas lain.

Daftar Isi

- [Kapan saatnya menggunakan Fargate](#)
- [Ketentuan tugas di Fargate](#)
- [Antrean tugas di Fargate](#)
- [Lingkungan komputasi di Fargate](#)

Kapan saatnya menggunakan Fargate

Kami merekomendasikan menggunakan Fargate di sebagian besar skenario. Fargate meluncurkan dan menskalakan komputasi agar sesuai dengan persyaratan sumber daya yang Anda tentukan untuk penampung. Dengan Fargate, Anda tidak perlu menyediakan terlalu banyak atau membayar server tambahan. Anda juga tidak perlu khawatir tentang spesifikasi parameter terkait infrastruktur seperti tipe instance. Ketika lingkungan komputasi perlu ditingkatkan, pekerjaan yang berjalan pada sumber daya Fargate dapat dimulai lebih cepat. Biasanya, dibutuhkan beberapa menit untuk memutar instans Amazon EC2 baru. Namun, pekerjaan yang berjalan di Fargate dapat disediakan dalam waktu sekitar 30 detik. Waktu yang tepat yang dibutuhkan tergantung pada beberapa faktor, termasuk ukuran gambar kontainer dan jumlah pekerjaan.

Namun, kami menyarankan Anda menggunakan Amazon EC2 jika pekerjaan Anda memerlukan salah satu dari yang berikut:

- Lebih dari 16 vCPU

- Lebih dari 120 gibibyte (GiB) memori
- SEBUAH GPU
- Gambar Mesin Amazon kustom (AMI)
- Salah satu parameter [LinuxParameters](#)

Jika Anda memiliki banyak pekerjaan, kami sarankan Anda menggunakan infrastruktur Amazon EC2. Misalnya, jika jumlah pekerjaan yang berjalan secara bersamaan melebihi batas pelambatan Fargate. Ini karena, dengan EC2, pekerjaan dapat dikirim pada tingkat yang lebih tinggi ke sumber daya EC2 daripada ke sumber daya Fargate. Selain itu, lebih banyak pekerjaan dapat berjalan secara bersamaan saat Anda menggunakan EC2. Untuk informasi lebih lanjut, lihat [Kuota layanan AWS Fargate](#) dalam Panduan Developer Amazon Elastic Container Service.

Ketentuan tugas di Fargate

Tugas AWS Batch di Fargate tidak mendukung semua parameter ketentuan tugas yang tersedia. Beberapa parameter tidak didukung sama sekali, dan parameter lainnya memiliki perilaku yang berbeda untuk tugas Fargate.

Daftar berikut ini menjelaskan parameter ketentuan tugas yang tidak valid atau dibatasi dalam tugas Fargate.

`platformCapabilities`

Harus ditentukan sebagai FARGATE.

```
"platformCapabilities": [ "FARGATE" ]
```

`type`

Harus ditentukan sebagai `container`.

```
"type": "container"
```

Parameter dalam `containerProperties`

`executionRoleArn`

Harus ditentukan untuk tugas yang berjalan di sumber daya Fargate. Untuk informasi lebih lanjut, lihat [IAM Role untuk Tugas](#) dalam Panduan Developer Amazon Elastic Container Service.

```
"executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole"
```

`fargatePlatformConfiguration`

(Opsional, hanya untuk ketentuan tugas Fargate). Menentukan versi platform Fargate, atau LATEST untuk versi platform terbaru. Nilai yang mungkin untuk `platformVersion` adalah 1.3.0, 1.4.0, dan LATEST (default).

```
"fargatePlatformConfiguration": { "platformVersion": "1.4.0" }
```

`instanceType`, `ulimits`

Tidak berlaku untuk tugas yang berjalan di sumber daya Fargate.

`memory`, `vcpus`

Pengaturan ini harus ditentukan dalam `resourceRequirements`

`privileged`

Jangan tentukan parameter ini, atau tentukan sebagai `false`.

```
"privileged": false
```

`resourceRequirements`

Persyaratan memori dan vCPU harus ditentukan menggunakan nilai yang [didukung](#). Sumber daya GPU tidak didukung untuk pekerjaan yang berjalan di sumber daya Fargate.

Jika Anda menggunakan GuardDuty Runtime Monitoring, ada sedikit overhead memori untuk agen GuardDuty keamanan. Oleh karena itu batas memori harus mencakup ukuran agen GuardDuty keamanan. Untuk informasi tentang batas memori agen GuardDuty keamanan, lihat [batas CPU dan memori](#) di Panduan GuardDuty Pengguna. Untuk informasi tentang praktik

terbaik, lihat [Bagaimana cara memperbaiki kesalahan memori pada tugas Fargate setelah mengaktifkan Runtime Monitoring](#) di Panduan Pengembang Amazon ECS.

```
"resourceRequirements": [  
  {"type": "MEMORY", "value": "512"},  
  {"type": "VCPU", "value": "0.25"}  
]
```

Parameter dalam `linuxParameters`

`devices`, `maxSwap`, `sharedMemorySize`, `swappiness`, `tmpfs`

Tidak berlaku untuk pekerjaan yang berjalan pada sumber daya Fargate.

Parameter dalam `logConfiguration`

`logDriver`

Hanya `awslogs` dan `splunk` yang didukung. Untuk informasi selengkapnya, lihat [Menggunakan driver log awslogs](#).

Anggota di `networkConfiguration`

`assignPublicIp`

Jika subnet pribadi tidak memiliki gateway NAT yang terpasang untuk mengirim lalu lintas ke Internet, [assignPublicIp](#) harus `"ENABLED"`. Untuk informasi selengkapnya, lihat [AWS Batch eksekusi peran IAM](#).

Antrean tugas di Fargate

Antrean tugas AWS Batch di Fargate pada dasarnya tidak berubah. Satu-satunya batasan adalah bahwa lingkungan komputasi yang terdaftar di semua `computeEnvironmentOrder` harus lingkungan komputasi Fargate (atau). FARGATE FARGATE_SPOT Lingkungan komputasi EC2 dan Fargate tidak dapat dicampur.

Lingkungan komputasi di Fargate

Lingkungan komputasi AWS Batch di Fargate tidak mendukung semua parameter lingkungan komputasi yang tersedia. Beberapa parameter tidak didukung sama sekali. Lainnya memiliki persyaratan khusus untuk Fargate.

Daftar berikut menjelaskan parameter lingkungan komputasi yang tidak valid atau dibatasi dalam pekerjaan Fargate.

type

Parameter ini harus MANAGED.

```
"type": "MANAGED"
```

Parameter di objek computeResources

`allocationStrategy`, `bidPercentage`, `desiredvCpus`, `imageId`, `instanceTypes`, `ec2Configuration`, `ec2KeyPair`, `instanceRole`, `launchTemplate`, `minvCpus`, `placementGroup`, `spotIamFleetRole`

Ini tidak berlaku untuk lingkungan komputasi Fargate dan tidak dapat disediakan.

subnets

Jika subnet yang tercantum dalam parameter ini tidak memiliki gateway NAT yang terlampir, parameter `assignPublicIp` dalam ketentuan tugas harus diatur ke `ENABLED`.

tags

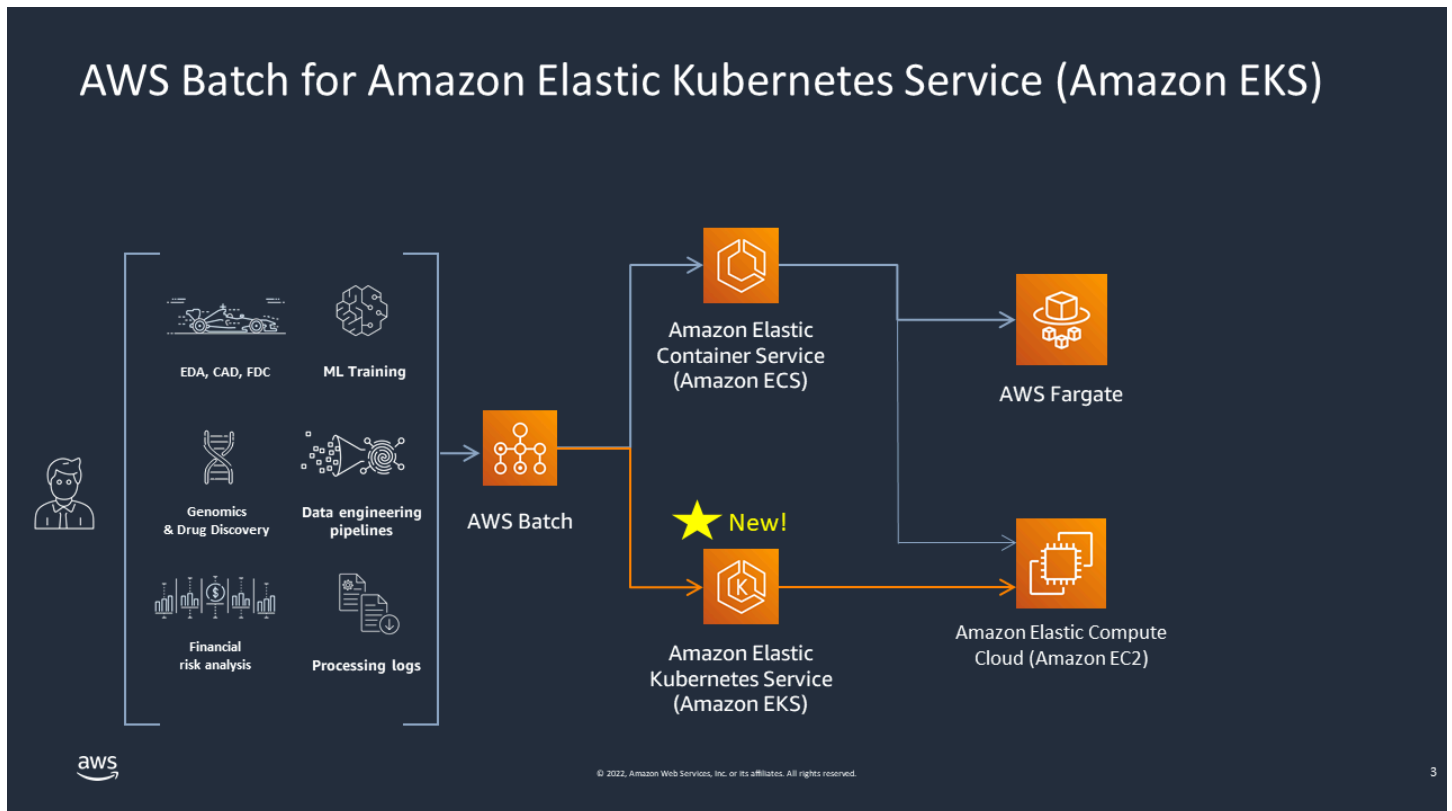
Ini tidak berlaku untuk lingkungan komputasi Fargate dan tidak dapat disediakan. Untuk menentukan tanda untuk lingkungan komputasi Fargate, gunakan parameter `tags` yang tidak ada di objek `computeResources`.

type

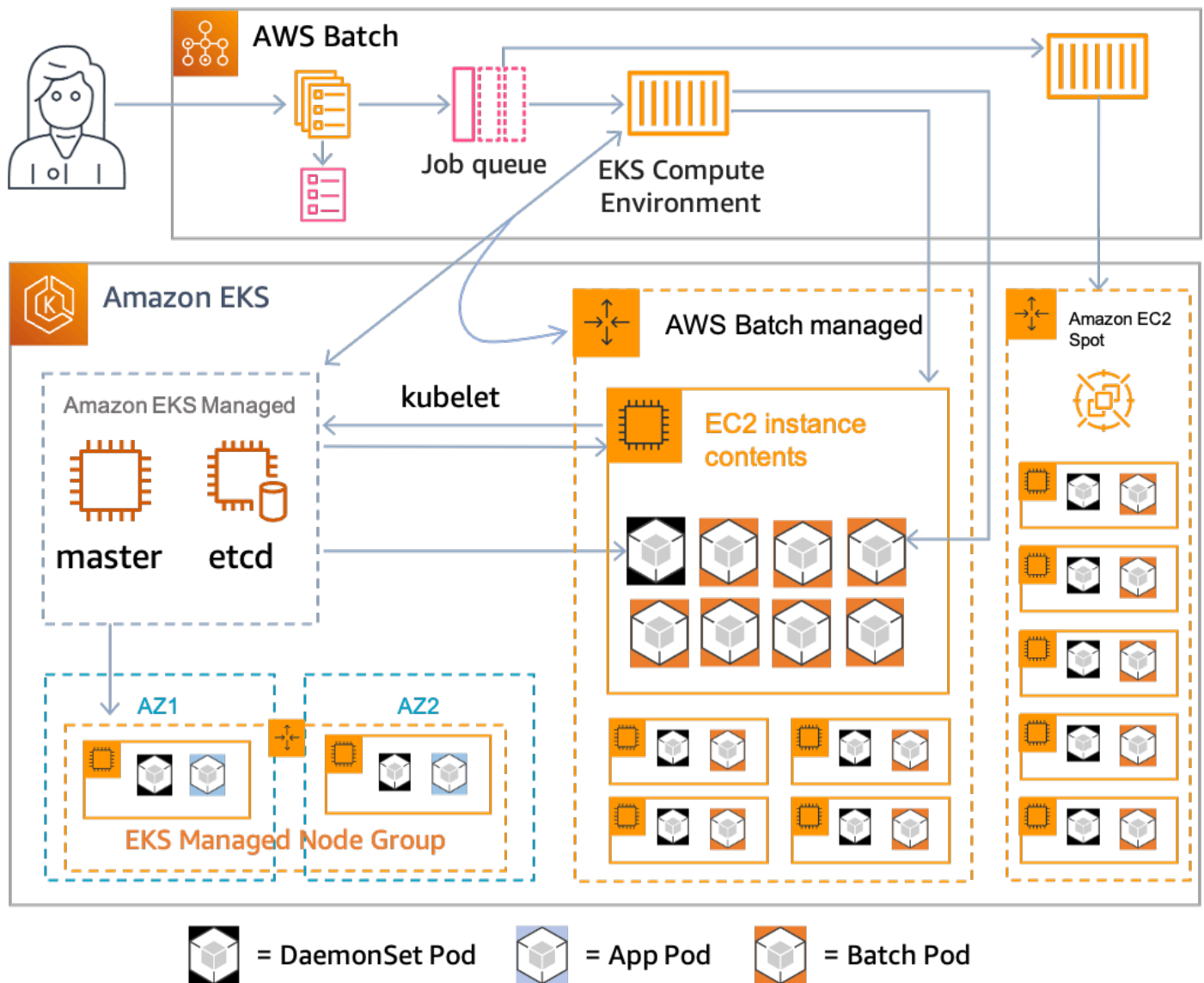
Ini harus `FARGATE` atau `FARGATE_SPOT`.

```
"type": "FARGATE_SPOT"
```

AWS Batch di Amazon EKS



AWS Batch menyederhanakan beban kerja batch Anda di kluster Amazon EKS dengan menyediakan kemampuan batch terkelola. Ini termasuk antrian, pelacakan ketergantungan, percobaan ulang dan prioritas pekerjaan terkelola, manajemen pod, dan penskalaan node. AWS Batch dapat menangani beberapa Availability Zone dan beberapa jenis dan ukuran instans Amazon EC2. AWS Batch mengintegrasikan beberapa praktik terbaik Amazon EC2 Spot untuk menjalankan beban kerja Anda dengan cara yang toleran terhadap kesalahan, sehingga mengurangi interupsi. Anda dapat menggunakan AWS Batch untuk menjalankan beberapa pekerjaan semalam atau jutaan pekerjaan mission-critical dengan percaya diri.



AWS Batch adalah layanan terkelola yang mengatur beban kerja batch di Kubernetes cluster Anda yang dikelola oleh Amazon Elastic Kubernetes Service (Amazon EKS). AWS Batch melakukan orkestrasi ini di luar cluster Anda menggunakan model “overlay”. Karena AWS Batch merupakan layanan terkelola, tidak ada Kubernetes komponen (misalnya, Operator atau Sumber Daya Kustom) untuk menginstal atau mengelola di cluster Anda. AWS Batch hanya perlu klaster Anda dikonfigurasi dengan Role-Based Access Controls (RBAC) yang memungkinkan AWS Batch untuk berkomunikasi dengan server API. Kubernetes AWS Batch memanggil Kubernetes API untuk membuat, memantau, dan menghapus Kubernetes pod dan node.

AWS Batch memiliki logika penskalaan bawaan untuk menskalakan Kubernetes node berdasarkan beban antrian pekerjaan dengan pengoptimalan dalam hal alokasi kapasitas pekerjaan. Saat antrian pekerjaan kosong, AWS Batch kurangi node ke kapasitas minimum yang Anda tetapkan, yang secara

default adalah nol. AWS Batch mengelola siklus hidup penuh dari node ini, dan menghias node dengan label dan taints. Dengan cara ini, Kubernetes beban kerja lain tidak ditempatkan pada node yang dikelola oleh AWS Batch. Pengecualian untuk ini adalah `DaemonSets`, yang dapat menargetkan AWS Batch node untuk menyediakan pemantauan dan fungsionalitas lain yang diperlukan untuk pelaksanaan pekerjaan yang tepat. Selain itu, AWS Batch tidak menjalankan pekerjaan, khususnya pod, pada node di cluster Anda yang tidak dikelola. Dengan cara ini, Anda dapat menggunakan logika dan layanan penskalaan terpisah untuk aplikasi lain di cluster.

Untuk mengirimkan lowongan kerja AWS Batch, Anda berinteraksi langsung dengan AWS Batch API. AWS Batch menerjemahkan pekerjaan ke dalam podspecs dan kemudian membuat permintaan untuk menempatkan pod pada node yang dikelola oleh AWS Batch di kluster Amazon EKS Anda. Anda dapat menggunakan alat seperti `kubectl` untuk melihat pod dan node yang sedang berjalan. Ketika sebuah pod telah menyelesaikan eksekusinya, AWS Batch menghapus pod yang dibuatnya untuk mempertahankan beban yang lebih rendah pada Kubernetes sistem.

Anda dapat memulai dengan menghubungkan kluster Amazon EKS yang valid dengan AWS Batch. Kemudian lampirkan antrian AWS Batch pekerjaan ke sana, dan daftarkan definisi pekerjaan Amazon EKS menggunakan atribut yang podspec setara. Terakhir, kirimkan pekerjaan menggunakan operasi [SubmitJob](#) API yang mengacu pada definisi pekerjaan. Untuk informasi selengkapnya, lihat [Memulai dengan AWS Batch di Amazon EKS](#).

Elastic Fabric Adapter

Elastic Fabric Adapter (EFA) adalah perangkat jaringan untuk mempercepat aplikasi Komputasi Kinerja Tinggi (HPC). AWS Batch mendukung aplikasi yang menggunakan EFA jika syarat berikut terpenuhi.

- Untuk daftar jenis instans yang mendukung EFA, lihat [Jenis instans yang didukung](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Tip

Untuk melihat daftar jenis instance yang mendukung EFA dalam sebuah Wilayah AWS, jalankan perintah berikut. Kemudian, referensi silang daftar yang dikembalikan dengan daftar jenis instance yang tersedia di AWS Batch konsol.

```
$ aws ec2 describe-instance-types --region us-east-1 --filters Name=network-info.efa-supported,Values=true --query "InstanceTypes[*].[InstanceType]" --output text | sort
```

- Untuk daftar sistem operasi yang mendukung EFA, lihat [Sistem operasi yang didukung](#).
- AMI memiliki driver EFA yang dimuat.
- Grup keamanan untuk EFA harus mengizinkan semua lalu lintas masuk dan keluar ke dan dari grup keamanan itu sendiri.
- Semua instance yang menggunakan EFA harus berada dalam grup penempatan cluster yang sama.
- Ketentuan tugas harus mencakup anggota devices dengan hostPath yang diatur ke /dev/infiniband/uverbs0 untuk mengizinkan perangkat EFA diteruskan ke kontainer. Jika containerPath ditentukan, itu juga harus diatur ke /dev/infiniband/uverbs0. Jika permissions diatur, itu harus diatur ke READ | WRITE | MKNOD.

Lokasi [LinuxParameters](#) anggota berbeda untuk pekerjaan paralel multi-node dan pekerjaan kontainer simpul tunggal. Contoh berikut menunjukkan perbedaan, tetapi tidak ada nilai yang diperlukan.

Example Contoh untuk tugas paralel multisimpul

```
{
```



```

"jobDefinitionName": "EFA-MNP-JobDef",
"type": "multinode",
"nodeProperties": {
  ...
  "nodeRangeProperties": [
    {
      ...
      "container": {
        ...
        "linuxParameters": {
          "devices": [
            {
              "hostPath": "/dev/infiniband/uverbs0",
              "containerPath": "/dev/infiniband/uverbs0",
              "permissions": [
                "READ", "WRITE", "MKNOD"
              ]
            },
          ],
        },
      },
    },
  ],
},
},
},
},
},
},
},
},
},
},
}

```

Example Contoh untuk tugas kontainer simpel tunggal

```

{
  "jobDefinitionName": "EFA-Container-JobDef",
  "type": "container",
  ...
  "containerProperties": {
    ...
    "linuxParameters": {
      "devices": [
        {
          "hostPath": "/dev/infiniband/uverbs0",
        },
      ],
    },
  },
},
}

```

Untuk informasi lebih lanjut tentang EFA, lihat [Elastic Fabric Adapter](#) dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Kebijakan, izin, dan IAM role AWS Batch

Secara default, pengguna tidak memiliki izin untuk membuat atau memodifikasi AWS Batch sumber daya atau untuk melakukan tugas menggunakan AWS Batch API, AWS Batch konsol, atau AWS CLI. Untuk memungkinkan pengguna melakukan tindakan ini, buat kebijakan IAM yang memberikan izin kepada pengguna untuk sumber daya dan operasi API tertentu. Kemudian, lampirkan kebijakan ke pengguna atau grup yang memerlukan izin tersebut.

Saat Anda melampirkan kebijakan ke pengguna atau grup pengguna, kebijakan tersebut mengizinkan atau menolak izin untuk melakukan tugas tertentu pada sumber daya tertentu. Untuk informasi selengkapnya, lihat [Izin dan Kebijakan](#) dalam Panduan Pengguna IAM. Untuk informasi selengkapnya tentang cara mengelola dan membuat kebijakan IAM, lihat [Mengelola Kebijakan IAM](#).

AWS Batch membuat panggilan ke orang lain Layanan AWS atas nama Anda. Akibatnya, AWS Batch harus mengautentikasi menggunakan kredensi Anda. Lebih khusus lagi, AWS Batch mengautentikasi dengan membuat peran dan kebijakan IAM yang menyediakan izin ini. Kemudian, ini mengaitkan peran dengan lingkungan komputasi Anda saat Anda membuatnya. Untuk informasi lebih lanjut, lihat [Peran instans Amazon ECS](#), [IAM Roles](#), [Menggunakan Peran yang Terhubung dengan Layanan](#), dan [Membuat Peran untuk Mendelegasikan Izin ke Layanan AWS](#) dalam Panduan Pengguna IAM.

Memulai

Kebijakan IAM harus memberikan atau menolak izin untuk menggunakan satu atau beberapa tindakan AWS Batch.

Topik

- [Struktur kebijakan](#)
- [Izin tingkat sumber daya yang didukung untuk tindakan API AWS Batch](#)
- [Contoh kebijakan](#)
- [Kebijakan terkelola AWS Batch](#)
- [Membuat kebijakan IAM AWS Batch](#)
- [Peran instans Amazon ECS](#)
- [Peran armada spot Amazon EC2](#)
- [EventBridge Peran IAM](#)

Struktur kebijakan

Topik-topik berikut ini menjelaskan struktur dari kebijakan IAM.

Topik

- [Sintaksis kebijakan](#)
- [Tindakan untuk AWS Batch](#)
- [Amazon Resource Names untuk AWS Batch](#)
- [Pemeriksaan jika pengguna memiliki izin yang diperlukan](#)

Sintaksis kebijakan

kebijakan IAM adalah dokumen JSON yang terdiri dari satu atau beberapa pernyataan. Masing-masing pernyataan memiliki struktur sebagai berikut.

```
{
  "Statement": [{
    "Effect": "effect",
    "Action": "action",
    "Resource": "arn",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  }
]
```

Ada berbagai elemen yang membentuk pernyataan:

- **Efek:** Efek bisa berupa Allow atau Deny. Secara default, pengguna tidak memiliki izin untuk menggunakan sumber daya dan tindakan API. Jadi, semua permintaan ditolak. izin eksplisit akan menggantikan izin default. penolakan eksplisit akan menggantikan izin apa pun.
- **Action:** Tindakan adalah tindakan API tertentu yang Anda berikan atau tolak izinnya. Untuk petunjuk tentang cara menentukan tindakan, lihat [Tindakan untuk AWS Batch](#).
- **Resource:** Sumber daya yang dipengaruhi oleh tindakan. Beberapa tindakan API AWS Batch memungkinkan Anda untuk menyertakan sumber daya tertentu dalam kebijakan Anda yang

dapat dibuat atau dimodifikasi oleh tindakan tersebut. Untuk menentukan sumber daya dalam pernyataan, gunakan Amazon Resource Name (ARN). Lihat informasi yang lebih lengkap di [Izin tingkat sumber daya yang didukung untuk tindakan API AWS Batch](#) dan [Amazon Resource Names untuk AWS Batch](#). Jika operasi AWS Batch API saat ini tidak mendukung izin tingkat sumber daya, sertakan wildcard (*) untuk menentukan bahwa semua sumber daya dapat terpengaruh oleh tindakan tersebut.

- Syarat: Syarat-syarat bersifat opsional. Syarat-syarat ini dapat digunakan untuk mengendalikan kapan kebijakan Anda berlaku.

Untuk informasi lebih lanjut tentang contoh pernyataan kebijakan IAM untuk AWS Batch, lihat [Membuat kebijakan IAM AWS Batch](#).

Tindakan untuk AWS Batch

Dalam pernyataan kebijakan IAM, Anda dapat menentukan tindakan API apa pun dari layanan apa pun yang mendukung IAM. Untuk AWS Batch, gunakan awalan berikut dengan nama tindakan API: `batch:` (misalnya, `batch:SubmitJob` dan `batch>CreateComputeEnvironment`).

Untuk menentukan beberapa tindakan dalam satu pernyataan, pisahkan setiap tindakan dengan koma.

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

Anda juga dapat menentukan beberapa tindakan dengan menyertakan wildcard (*). Misalnya, Anda dapat menentukan semua tindakan dengan nama yang dimulai dengan kata "Jelaskan."

```
"Action": "batch:Describe*"
```

Untuk menentukan semua tindakan AWS Batch API, sertakan wildcard (*).

```
"Action": "batch:*"
```

Untuk melihat daftar tindakan AWS Batch, lihat [Actions](#) dalam Referensi API AWS Batch.

Amazon Resource Names untuk AWS Batch

Setiap pernyataan kebijakan IAM berlaku untuk sumber daya yang Anda tentukan menggunakan Nama Sumber Daya Amazon (ARN) mereka.

Nama Sumber Daya Amazon (ARN) memiliki sintaks umum berikut:

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

layanan

Layanan (contohnya, batch).

wilayah

Wilayah AWS Untuk sumber daya (misalnya, us-east-2).

akun

Akun AWSID, tanpa tanda hubung (misalnya, 123456789012).

resourceType

Jenis dari sumber daya (contohnya, compute-environment).

resourcePath

jalur yang mengidentifikasi sumber daya. Anda dapat menggunakan wildcard (*) di jalur Anda.

AWS Batch Operasi API saat ini mendukung izin tingkat sumber daya pada beberapa operasi API. Untuk informasi selengkapnya, lihat [Izin tingkat sumber daya yang didukung untuk tindakan API AWS Batch](#). Untuk menentukan semua sumber daya, atau jika tindakan API tertentu tidak mendukung ARN, sertakan wildcard (*) dalam elemen Resource

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

Pemeriksaan jika pengguna memiliki izin yang diperlukan

Sebelum Anda memasukkan kebijakan IAM ke dalam produksi, pastikan kebijakan tersebut memberi pengguna izin untuk menggunakan tindakan dan sumber daya API tertentu yang mereka butuhkan.

Untuk melakukan ini, pertama-tama buat pengguna untuk tujuan pengujian dan lampirkan kebijakan IAM ke pengguna uji. Kemudian, buatlah permintaan sebagai pengguna uji. Anda dapat membuat permintaan tes di konsol atau dengan AWS CLI.

Note

Anda juga dapat menguji kebijakan Anda dengan menggunakan [IAM Policy Simulator](#). Untuk informasi selengkapnya tentang simulator kebijakan, lihat [Bekerja dengan Simulator Kebijakan IAM](#) dalam Panduan Pengguna IAM.

Jika kebijakan tidak memberikan izin kepada pengguna seperti yang Anda harapkan, atau terlalu longgar, Anda dapat menyesuaikan kebijakan sesuai kebutuhan. Lakukan pengujian ulang sampai Anda mendapatkan hasil yang diinginkan.

Important

Pengujian ini dapat memakan waktu beberapa menit sebelum perubahan terjadi pada kebijakan untuk ditransmisikan sebelum diberlakukan. Oleh karena itu, kami menyarankan agar Anda mengizinkan setidaknya lima menit berlalu sebelum Anda menguji pembaruan kebijakan Anda.

Jika pemeriksaan otorisasi gagal, maka permintaan akan menampilkan informasi berencode yang memuat informasi diagnostik. Anda dapat melakukan decode pada pesan tersebut menggunakan tindakan `DecodeAuthorizationMessage`. Untuk informasi selengkapnya, lihat [DecodeAuthorizationMessage](#) di Referensi AWS Security Token Service API, dan [decode-authorization-message](#) di Referensi AWS CLI Perintah.

Izin tingkat sumber daya yang didukung untuk tindakan API AWS Batch

Istilah izin tingkat sumber daya mengacu kepada kemampuan untuk menentukan sumber daya yang boleh digunakan pengguna untuk melakukan tindakan. AWS Batch memiliki dukungan parsial untuk izin tingkat sumber daya. Untuk beberapa AWS Batch tindakan, Anda dapat mengontrol kapan pengguna diizinkan untuk menggunakan tindakan tersebut berdasarkan kondisi yang harus dipenuhi. Anda juga dapat mengontrol berdasarkan sumber daya spesifik yang diizinkan untuk digunakan pengguna. Misalnya, Anda dapat memberikan izin kepada pengguna untuk mengirimkan tugas, tetapi hanya ke antrian tugas tertentu dan hanya dengan ketentuan tugas tertentu.

Daftar berikut menjelaskan tindakan AWS Batch API yang saat ini mendukung izin tingkat sumber daya. Daftar ini juga menjelaskan sumber daya yang didukung, ARN sumber daya, dan kunci kondisi untuk setiap tindakan.

Important

Jika suatu tindakan API AWS Batch tidak tercantum dalam daftar ini, itu berarti tindakan tersebut tidak mendukung izin tingkat sumber daya. Jika tindakan AWS Batch API tidak mendukung izin tingkat sumber daya, Anda dapat memberikan izin kepada pengguna untuk menggunakan tindakan tersebut. Namun, Anda harus menyertakan wildcard (*) untuk elemen sumber daya pernyataan kebijakan Anda.

Tindakan

[CancelJob](#), [CreateComputeEnvironment](#), [CreateJobQueue](#), [CreateSchedulingPolicy](#), [DeleteComputeEnvironment](#), [DeleteJobQueue](#), [DeleteSchedulingPolicy](#), [DeregisterJobDefinition](#), [ListTagsForResource](#), [RegisterJobDefinition](#), [SubmitJob](#), [TagResource](#), [TerminateJob](#), [UntagResource](#), [UpdateComputeEnvironment](#), [UpdateSchedulingPolicy](#), [UpdateJobQueue](#)

[CancelJob](#)

Membatalkan tugas di antrean AWS Batch.

Sumber

Job

arn:aws:batch:*region*:*account*:job/*jobId*

Kunci kondisi

aws:ResourceTag/\${TagKey} (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

[CreateComputeEnvironment](#)

Membuat lingkungan komputasi AWS Batch.

Sumber

Lingkungan Komputasi

arn:aws:batch: wilayah: akun: komputasi-lingkungan/ compute-environment-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Kunci kondisi

`aws:RequestTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang diberikan dalam permintaan.

`aws:TagKeys` (String)

Memfilter tindakan berdasarkan kunci tanda yang diberikan dalam permintaan.

[CreateJobQueue](#)

Membuat antrian tugas AWS Batch.

Sumber

Lingkungan Komputasi

arn:aws:batch: wilayah: akun: komputasi-lingkungan/ compute-environment-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Antrian Job

arn:aws:batch:region:account:job-queue/queue-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Kebijakan Penjadwalan

arn:aws:batch: wilayah: akun: penjadwalan-kebijakan/scheduling-policy-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Kunci kondisi

`aws:RequestTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang diberikan dalam permintaan.

`aws:TagKeys` (String)

Memfilter tindakan berdasarkan kunci tanda yang diberikan dalam permintaan.

DeleteComputeEnvironment

Menghapus lingkungan komputasi AWS Batch.

Sumber

Lingkungan Komputasi

arn:aws:batch: wilayah: akun: komputasi-lingkungan/ compute-environment-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

CreateSchedulingPolicy

Membuat kebijakan AWS Batch penjadwalan.

Sumber

Kebijakan Penjadwalan

arn:aws:batch: wilayah: akun: penjadwalan-kebijakan/scheduling-policy-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Kunci kondisi

`aws:RequestTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang diberikan dalam permintaan.

`aws:TagKeys` (String)

Memfilter tindakan berdasarkan kunci tanda yang diberikan dalam permintaan.

[DeleteJobQueue](#)

Menghapus antrian tugas tertentu. Penghapusan antrian tugas pada akhirnya akan menghapus semua tugas dalam antrian. Tugas dihapus dengan kecepatan sekitar 16 tugas setiap detik.

Sumber

Antrian Job

`arn:aws:batch:region:account:job-queue/queue-name`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

[DeleteSchedulingPolicy](#)

Menghapus kebijakan penjadwalan yang ditentukan.

Sumber

Kebijakan Penjadwalan

`arn:aws:batch: wilayah: akun: penjadwalan-kebijakan/scheduling-policy-name`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

DeregisterJobDefinition

Membatalkan pendaftaran ketentuan tugas AWS Batch.

Sumber

Definisi Job

arn:aws:batch:*region:account*:job-definition/*definition-name:revision*

Kunci kondisi

aws:ResourceTag/\${TagKey} (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

ListTagsForResource

Mendaftar tanda untuk sumber daya tertentu.

Sumber

Lingkungan Komputasi

arn:aws:batch: wilayah: akun: komputasi-lingkungan/ compute-environment-name

Kunci kondisi

aws:ResourceTag/\${TagKey} (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Job

arn:aws:batch:*region:account*:job/*jobId*

Kunci kondisi

aws:ResourceTag/\${TagKey} (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Definisi Job

arn:aws:batch:*region:account*:job-definition/*definition-name:revision*

Kunci kondisi

aws:ResourceTag/\${TagKey} (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Antrian Job

arn:aws:batch:region:account:job-queue/queue-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Kebijakan Penjadwalan

arn:aws:batch: wilayah: akun: penjadwalan-kebijakan/scheduling-policy-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

RegisterJobDefinition

Mendaftarkan ketentuan AWS Batch.

Sumber

Definisi Job

arn:aws:batch: region: account:job-definition/definition-name

Kunci kondisi

`batch:AWSLogsCreateGroup` (Boolean)

Jika parameter ini betul, `awslogs-group` akan dibuat untuk log.

`batch:AWSLogsGroup` (String)

Grup `awslogs` tempat log berada.

`batch:AWSLogsRegion` (String)

Wilayah tujuan pengiriman log.

`batch:AWSLogsStreamPrefix` (String)

Prefiks pengaliran log `awslogs`.

`batch:Image` (String)

Gambar Docker yang digunakan untuk memulai tugas.

`batch:LogDriver` (String)

Driver log yang digunakan untuk tugas.

`batch:Privileged` (Boolean)

Ketika parameter ini benar, penampung untuk pekerjaan tersebut diberikan izin yang ditinggikan pada instance wadah host.

`batch:User` (String)

Nama pengguna atau uid numerik untuk digunakan dalam kontainer untuk tugas.

`aws:RequestTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang diberikan dalam permintaan.

`aws:TagKeys` (String)

Memfilter tindakan berdasarkan kunci tanda yang diberikan dalam permintaan.

[SubmitJob](#)

Mengirimkan tugas AWS Batch dari ketentuan tugas.

Sumber

Job

`arn:aws:batch:region:account:job/jobId`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Definisi Job

`arn:aws:batch: region: account:job-definition/definition-name [: revisi]`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Note

Kunci ini hanya dapat digunakan ketika definisi pekerjaan Amazon Resource Name (ARN) dalam format `arn:aws:batch:region:account_number:job-definition/definition-name:revision`.

Antrian Job

`arn:aws:batch:region:account:job-queue/queue-name`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

TagResource

Menandai sumber daya yang ditentukan.

Sumber

Lingkungan Komputasi

`arn:aws:batch: wilayah: akun: komputasi-lingkungan/ compute-environment-name`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Job

`arn:aws:batch:region:account:job/jobId`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Definisi Job

`arn:aws:batch:region:account:job-definition/definition-name:revision`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Antrian Job

`arn:aws:batch:region:account:job-queue/queue-name`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Kebijakan Penjadwalan

`arn:aws:batch: wilayah: akun: penjadwalan-kebijakan/scheduling-policy-name`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Kunci kondisi

`aws:RequestTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang diberikan dalam permintaan.

`aws:TagKeys` (String)

Memfilter tindakan berdasarkan kunci tanda yang diberikan dalam permintaan.

TerminateJob

Mengakhiri tugas di antrean tugas AWS Batch.

Sumber

Job

`arn:aws:batch:region:account:job/jobId`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

UntagResource

Membatalkan tag sumber daya yang ditentukan.

Sumber

Lingkungan Komputasi

arn:aws:batch: wilayah: akun: komputasi-lingkungan/ compute-environment-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Job

arn:aws:batch:region:account:job/jobId

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Definisi Job

arn:aws:batch:region:account:job-definition/definition-name:revision

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Antrian Job

arn:aws:batch:region:account:job-queue/queue-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Kebijakan Penjadwalan

arn:aws:batch: wilayah: akun: penjadwalan-kebijakan/scheduling-policy-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Kunci kondisi

`aws:TagKeys` (String)

Memfilter tindakan berdasarkan kunci tanda yang diberikan dalam permintaan.

UpdateComputeEnvironment

Memperbarui lingkungan komputasi AWS Batch.

Sumber

Lingkungan Komputasi

arn:aws:batch: wilayah: akun: komputasi-lingkungan/ compute-environment-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

UpdateJobQueue

Memperbarui antrean tugas.

Sumber

Antrian Job

arn:aws:batch:region:account:job-queue/queue-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Kebijakan Penjadwalan

arn:aws:batch: wilayah: akun: penjadwalan-kebijakan/scheduling-policy-name

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

[UpdateSchedulingPolicy](#)

Memperbarui kebijakan penjadwalan.

Sumber

Kebijakan Penjadwalan

`arn:aws:batch: wilayah: akun: penjadwalan-kebijakan/scheduling-policy-name`

Kunci kondisi

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

Kunci syarat untuk tindakan API AWS Batch

AWS Batch mendefinisikan kunci kondisi berikut yang digunakan dalam `Condition` elemen kebijakan IAM. Anda dapat menggunakan kunci ini untuk menyempurnakan kondisi yang berlaku untuk pernyataan kebijakan. Untuk melihat kunci syarat global yang tersedia untuk semua layanan, lihat [kunci syarat global yang tersedia](#) dalam Panduan Pengguna IAM.

`batch:AWSLogsCreateGroup` (Boolean)

Jika parameter ini betul, `awslogs-group` akan dibuat untuk log.

`batch:AWSLogsGroup` (String)

Grup `awslogs` tempat log berada.

`batch:AWSLogsRegion` (String)

Wilayah AWS tempat log dikirim ke.

`batch:AWSLogsStreamPrefix` (String)

Prefiks pengaliran log `awslogs`.

`batch:Image` (String)

Gambar Docker yang digunakan untuk memulai tugas.

`batch:LogDriver` (String)

Driver log yang digunakan untuk tugas.

`batch:Privileged` (Boolean)

Jika parameter ini betul, kontainer untuk tugas akan diberi izin yang lebih tinggi di instans kontainer host (mirip dengan pengguna asal).

`aws:ResourceTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang dikaitkan dengan sumber daya.

`aws:RequestTag/${TagKey}` (String)

Memfilter tindakan berdasarkan tanda yang diberikan dalam permintaan.

`batch:ShareIdentifier` (String)

Menyaring tindakan berdasarkan `shareIdentifier` parameter yang dikirim ke [SubmitJob](#).

`aws:TagKeys` (String)

Memfilter tindakan berdasarkan kunci tanda yang diberikan dalam permintaan.

`batch:User` (String)

Nama pengguna atau ID pengguna numerik (uid) untuk digunakan di dalam wadah untuk pekerjaan itu.

Contoh kebijakan

Contoh berikut menunjukkan pernyataan kebijakan yang dapat Anda gunakan untuk mengontrol izin yang dimiliki pengguna. AWS Batch

Contoh-contoh

- [Akses hanya-baca](#)
- [Membatasi ke pengguna POSIX, gambar Docker, tingkat hak istimewa, dan peran pada pengiriman pekerjaan](#)
- [Batasi awalan definisi pekerjaan pada pengajuan pekerjaan](#)
- [Batasi antrian pekerjaan](#)
- [Tolak tindakan saat mengkondisikan semua tombol cocok dengan string](#)
- [Tolak tindakan ketika tombol kondisi apa pun cocok dengan string](#)

- [Gunakan tombol batch:ShareIdentifier kondisi](#)

Akses hanya-baca

Kebijakan berikut memberikan izin kepada pengguna untuk menggunakan semua tindakan AWS Batch API dengan nama yang dimulai dengan `Describe` dan `List`

Kecuali pernyataan lain memberi mereka izin untuk melakukannya, pengguna tidak memiliki izin untuk melakukan tindakan apa pun pada sumber daya. Secara default, mereka ditolak izin untuk menggunakan tindakan API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:Describe*",
        "batch:List*"
      ],
      "Resource": "*"
    }
  ]
}
```

Membatasi ke pengguna POSIX, gambar Docker, tingkat hak istimewa, dan peran pada pengiriman pekerjaan

Kebijakan berikut memungkinkan pengguna POSIX untuk mengelola kumpulan definisi pekerjaan terbatas mereka sendiri.

Gunakan pernyataan pertama dan kedua untuk mendaftar dan membatalkan pendaftaran nama definisi pekerjaan apa pun yang namanya diawali dengan `A_`. `JobDef`

Pernyataan pertama juga menggunakan kunci konteks bersyarat untuk membatasi pengguna POSIX, status istimewa, dan nilai-nilai citra kontainer dalam ketentuan tugas `containerProperties`. Untuk informasi selengkapnya, lihat [RegisterJobDefinition](#) di dalam Referensi API AWS Batch. Dalam contoh ini, definisi pekerjaan hanya dapat didaftarkan ketika pengguna POSIX disetel `kenobody`.

Bendera istimewa diatur ke `false`. Terakhir, gambar diatur ke `myImage` dalam repositori Amazon ECR.

⚠ Important

Docker menyelesaikan `user` parameter ke pengguna itu `uid` dari dalam gambar kontainer. Dalam kebanyakan kasus, ini ditemukan di file `/etc/passwd` dalam citra kontainer. Resolusi nama ini dapat dihindari dengan menggunakan nilai `uid` langsung dalam ketentuan tugas dan kebijakan IAM apa pun yang terkait. Baik operasi API AWS Batch maupun kunci bersyarat IAM `batch:User` mendukung nilai numerik.

Gunakan pernyataan ketiga untuk membatasi hanya peran tertentu pada definisi pekerjaan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:User": [
            "nobody"
          ],
          "batch:Image": [
            "<aws_account_id>.dkr.ecr.<aws_region>.amazonaws.com/myImage"
          ]
        },
        "Bool": {
          "batch:Privileged": "false"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
```

```

        "batch:DeregisterJobDefinition"
    ],
    "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam:<aws_account_id>:role/MyBatchJobRole"
    ]
}
]
}

```

Batasi awalan definisi pekerjaan pada pengajuan pekerjaan

Gunakan kebijakan berikut untuk mengirimkan pekerjaan ke antrian pekerjaan apa pun dengan nama definisi pekerjaan apa pun yang dimulai dengan *JobDefA*.

Important

Ketika membuat cakupan akses tingkat sumber daya untuk pengiriman tugas, Anda harus menyediakan antrian tugas dan tipe sumber daya ketentuan tugas.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*",
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-queue/*"
      ]
    }
  ]
}

```

```
]
}
```

Batasi antrian pekerjaan

Gunakan kebijakan berikut untuk mengirimkan pekerjaan ke antrian pekerjaan tertentu yang diberi nama `queue1` dengan nama definisi pekerjaan apa pun.

Important

Ketika membuat cakupan akses tingkat sumber daya untuk pengiriman tugas, Anda harus menyediakan antrean tugas dan tipe sumber daya ketentuan tugas.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/*",
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-queue/queue1"
      ]
    }
  ]
}
```

Tolak tindakan saat mengkondisikan semua tombol cocok dengan string

Kebijakan berikut menolak akses ke operasi [RegisterJobDefinitionAPI](#) ketika kunci kondisi `batch:Image` (ID gambar kontainer) adalah `"string1"` dan kunci kondisi `batch:LogDriver` (driver log kontainer) adalah `"string2."` AWS Batch mengevaluasi kunci kondisi pada setiap kontainer. Ketika sebuah pekerjaan mencakup beberapa kontainer seperti pekerjaan paralel multi-node, kontainer mungkin memiliki konfigurasi yang berbeda. Jika beberapa kunci kondisi dievaluasi dalam satu pernyataan, mereka digabungkan menggunakan AND logika. Jadi, jika salah satu dari beberapa kunci kondisi tidak cocok untuk wadah, Deny efeknya

tidak diterapkan untuk wadah itu. Sebaliknya, wadah yang berbeda dalam pekerjaan yang sama mungkin ditolak.

Untuk daftar kunci kondisi AWS Batch, lihat [Kunci kondisi untuk AWS Batch](#) dalam Referensi Otorisasi Layanan. Kecuali `batch:ShareIdentifier`, semua tombol batch kondisi dapat digunakan dengan cara ini. Kunci `batch:ShareIdentifier` kondisi didefinisikan untuk pekerjaan, bukan definisi pekerjaan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": "batch:RegisterJobDefinition",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "batch:Image": "string1",
          "batch:LogDriver": "string2"
        }
      }
    }
  ]
}
```

Tolak tindakan ketika tombol kondisi apa pun cocok dengan string

*Kebijakan berikut menolak akses ke operasi [RegisterJobDefinition](#) API ketika kunci kondisi `batch:Image` (ID gambar kontainer) adalah "*string1*" atau kunci kondisi `batch:LogDriver` (driver log kontainer) adalah "*string2*." Ketika sebuah pekerjaan mencakup beberapa kontainer seperti pekerjaan paralel multi-node, kontainer mungkin memiliki konfigurasi yang berbeda. Jika beberapa kunci kondisi dievaluasi dalam satu*

pernyataan, mereka digabungkan menggunakan AND logika. Jadi, jika salah satu dari beberapa kunci kondisi tidak cocok untuk wadah, Deny efeknya tidak diterapkan untuk wadah itu. Sebaliknya, wadah yang berbeda dalam pekerjaan yang sama mungkin ditolak.

Untuk daftar kunci kondisi AWS Batch, lihat [Kunci kondisi untuk AWS Batch](#) dalam Referensi Otorisasi Layanan. Kecuali `batch:ShareIdentifier`, semua tombol batch kondisi dapat digunakan dengan cara ini. (Kunci `batch:ShareIdentifier` kondisi didefinisikan untuk pekerjaan, bukan definisi pekerjaan.)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:Image": [
            "string1"
          ]
        }
      }
    }
  ],
  {
    "Effect": "Deny",
    "Action": [
      "batch:RegisterJobDefinition"
    ],
    "Resource": [
```

```

        "*"
    ],
    "Condition": {
        "StringEquals": {
            "batch:LogDriver": [
                "string2"
            ]
        }
    }
}
]
}

```

Gunakan tombol **batch:ShareIdentifier** kondisi

Gunakan kebijakan berikut untuk mengirimkan lowongan yang menggunakan definisi jobDefA pekerjaan ke antrian jobqueue1 pekerjaan dengan pengenal lowCpu berbagi.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws::batch:<aws_region>:<aws_account_id>:job-definition/JobDefA",
        "arn:aws::batch:<aws_region>:<aws_account_id>:job-queue/jobqueue1"
      ],
      "Condition": {
        "StringEquals": {
          "batch:ShareIdentifier": [
            "lowCpu"
          ]
        }
      }
    }
  ]
}

```

Kebijakan terkelola AWS Batch

AWS Batch menyediakan kebijakan terkelola yang dapat Anda lampirkan ke pengguna yang memberikan izin untuk menggunakan AWS Batch sumber daya dan operasi API. Anda dapat menerapkan kebijakan ini secara langsung, atau Anda dapat menggunakannya sebagai titik awal untuk membuat kebijakan Anda sendiri. Untuk informasi lebih lanjut tentang setiap operasi API yang disebutkan dalam kebijakan ini, lihat [Tindakan](#) di Referensi API AWS Batch.

AWSBatchFullAccess

Kebijakan ini memungkinkan akses administrator penuh ke AWS Batch.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:*",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeImages",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ecs:DescribeClusters",
        "ecs:Describe*",
        "ecs:List*",
        "eks:DescribeCluster",
        "eks:ListClusters",
        "logs:Describe*",
        "logs:Get*",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "iam:ListInstanceProfiles",
        "iam:ListRoles"
      ],
      "Resource": "*"
    }
  ]
}
```

```
"Effect": "Allow",
"Action": [
  "iam:PassRole"
],
"Resource": [
  "arn:aws:iam::*:role/AWSBatchServiceRole",
  "arn:aws:iam::*:role/service-role/AWSBatchServiceRole",
  "arn:aws:iam::*:role/ecsInstanceRole",
  "arn:aws:iam::*:instance-profile/ecsInstanceRole",
  "arn:aws:iam::*:role/iaws-ec2-spot-fleet-role",
  "arn:aws:iam::*:role/aws-ec2-spot-fleet-role",
  "arn:aws:iam::*:role/AWSBatchJobRole*"
]
},
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole"
  ],
  "Resource": "arn:aws:iam::*:role/*Batch*",
  "Condition": {
    "StringEquals": {
      "iam:AWSServiceName": "batch.amazonaws.com"
    }
  }
}
]
```

Membuat kebijakan IAM AWS Batch

Anda dapat membuat kebijakan IAM tertentu untuk membatasi panggilan dan sumber daya yang dapat diakses oleh pengguna di akun Anda. Kemudian, Anda dapat melampirkan kebijakan tersebut ke pengguna.

Saat Anda melampirkan kebijakan ke pengguna atau grup pengguna, kebijakan tersebut mengizinkan atau menolak izin pengguna untuk tugas tertentu pada sumber daya tertentu. Untuk informasi selengkapnya, lihat [Izin dan Kebijakan](#) dalam Panduan Pengguna IAM. Untuk petunjuk tentang cara mengelola dan membuat kebijakan IAM kustom, lihat [Mengelola Kebijakan IAM](#).

Peran instans Amazon ECS

AWS Batch lingkungan komputasi diisi dengan instans kontainer Amazon ECS. Mereka menjalankan agen kontainer Amazon ECS secara lokal. Agen kontainer Amazon ECS membuat panggilan ke berbagai operasi API AWS atas nama Anda. Oleh karena itu, instans kontainer yang menjalankan agen memerlukan kebijakan dan IAM role untuk layanan ini untuk mengenali bahwa agen tersebut merupakan milik Anda. Anda harus membuat peran IAM dan profil instance untuk instance container yang akan digunakan saat diluncurkan. Jika tidak, Anda tidak dapat membuat lingkungan komputasi dan meluncurkan instans kontainer ke dalamnya. Persyaratan ini berlaku untuk instans kontainer yang diluncurkan dengan atau tanpa AMI yang dioptimalkan untuk Amazon ECS yang disediakan oleh Amazon. Untuk informasi lebih lanjut, lihat [IAM role instans kontainer Amazon ECS](#) dalam Panduan Developer Amazon Elastic Container Service.

Peran instans dan profil instans Amazon ECS secara otomatis dibuat untuk Anda di konsol pengalaman perjalanan pertama kali. Namun, Anda dapat mengikuti langkah-langkah ini untuk memeriksa apakah akun Anda sudah memiliki peran instans Amazon ECS dan profil instans. Langkah-langkah berikut juga mencakup cara melampirkan kebijakan IAM terkelola.

Untuk memeriksa **ecsInstanceRole** di dalam konsol IAM

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Peran.
3. Cari daftar peran untuk `ecsInstanceRole`. Jika peran tidak ada, gunakan langkah-langkah berikut untuk membuat peran.
 - a. Pilih Buat Peran.
 - b. Untuk jenis entitas Tepercaya, pilih Layanan AWS.
 - c. Untuk kasus penggunaan umum, pilih EC2.
 - d. Pilih Berikutnya.
 - e. Untuk kebijakan Izin, cari `ContainerServiceforAmazonec2 EC2role`.
 - f. Pilih kotak centang di sebelah `Amazonec2 ContainerServicefor EC2role`, lalu pilih Berikutnya.
 - g. Untuk Role Name (Nama Peran), ketik `ecsInstanceRole` dan pilih Create Role (Buat Peran).

Note

Jika Anda menggunakan AWS Management Console untuk membuat peran Amazon EC2, konsol akan membuat profil instans dengan nama yang sama dengan peran.

Atau, Anda dapat menggunakan AWS CLI untuk membuat peran `ecsInstanceRole` IAM. Contoh berikut membuat peran IAM dengan kebijakan kepercayaan dan kebijakan AWS terkelola.

Untuk membuat IAM role dan profil instans (AWS CLI)

1. Buat kebijakan kepercayaan berikut dan simpan dalam file teks yang diberi nama `ecsInstanceRole-role-trust-policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Gunakan perintah [create-role](#) untuk membuat peran. `ecsInstanceRole` Tentukan lokasi file kebijakan kepercayaan dalam `assume-role-policy-document` parameter.

```
$ aws iam create-role \
  --role-name ecsInstanceRole \
  --assume-role-policy-document file://ecsInstanceRole-role-trust-policy.json
```

Berikut ini adalah contoh respons.

```
{
  "Role": {
    "Path": "/",
    "RoleName": "ecsInstanceRole",
    "RoleId": "AR0AT46P5RDIY4EXAMPLE",
```

```

    "Arn": "arn:aws:iam::123456789012:role/ecsInstanceRole".
    "CreateDate": "2022-12-12T23:46:37.247Z",
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "ec2.amazonaws.com"
          }
          "Action": "sts:AssumeRole",
        }
      ]
    }
  }
}

```

- Gunakan [create-instance-profile](#) perintah untuk membuat profil instance yang diberi nama `ecsInstanceRole`.

Note

Anda perlu membuat peran dan profil instance sebagai tindakan terpisah di AWS API AWS CLI dan API.

```
$ aws iam create-instance-profile --instance-profile-name ecsInstanceRole
```

Berikut ini adalah contoh respons.

```

{
  "InstanceProfile": {
    "Path": "/",
    "InstanceProfileName": "ecsInstanceRole",
    "InstanceProfileId": "AIPAT46P5RDITREXAMPLE",
    "Arn": "arn:aws:iam::123456789012:instance-profile/ecsInstanceRole",
    "CreateDate": "2022-06-30T23:53:34.093Z",
    "Roles": [],
  }
}

```

- Gunakan perintah [add-role-to-instance-profile](#) untuk menambahkan `ecsInstanceRole` peran ke profil `ecsInstanceRole` instance.


```
aws iam add-role-to-instance-profile \  
  --role-name ecsInstanceRole --instance-profile-name ecsInstanceRole
```

5. Gunakan [attach-role-policy](#) perintah untuk melampirkan kebijakan AmazonEC2ContainerServiceforEC2Role AWS terkelola ke ecsInstanceRole peran.

```
$ aws iam attach-role-policy \  
  --policy-arn arn:aws:iam::aws:policy/service-role/  
AmazonEC2ContainerServiceforEC2Role \  
  --role-name ecsInstanceRole
```

Peran armada spot Amazon EC2

Jika Anda membuat lingkungan komputasi terkelola yang menggunakan Instans Armada Spot Amazon EC2, Anda harus membuat kebijakan. AmazonEC2SpotFleetTaggingRole Kebijakan ini memberikan izin kepada Armada Spot untuk meluncurkan, menandai, dan menghentikan instans atas nama Anda. Tentukan peran dalam permintaan Armada Spot Anda. Anda juga harus memiliki peran AWSServiceRoleForEC2Spot dan AWSServiceRoleForEC2SpotFleet terkait layanan untuk Armada Spot dan Spot Amazon EC2. Gunakan instruksi berikut untuk membuat semua peran. Untuk informasi lebih lanjut, lihat [Menggunakan Peran yang Terhubung dengan Layanan](#) dan [Membuat Peran untuk Mendelegasikan Izin ke Layanan AWS](#) dalam Panduan Pengguna IAM.

Topik

- [Buat peran armada spot Amazon EC2 di AWS Management Console](#)
- [Buat Peran Armada Spot Amazon EC2 dengan AWS CLI](#)

Buat peran armada spot Amazon EC2 di AWS Management Console

Untuk membuat IAM role yang terhubung dengan layanan **AmazonEC2SpotFleetTaggingRole** untuk Armada Spot Amazon EC2

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Untuk Manajemen Akses, pilih Peran,
3. Untuk Peran, pilih Buat peran.
4. Dari Pilih entitas tepercaya untuk jenis entitas Tepercaya, pilih Layanan AWS.

5. Untuk kasus Penggunaan untuk kasus lainnya Layanan AWS, pilih EC2 dan kemudian pilih EC2 - Spot Fleet Tagging.
6. Pilih Berikutnya.
7. Dari kebijakan Izin untuk nama Kebijakan, verifikasi `AmazonEC2SpotFleetTaggingRole`.
8. Pilih Berikutnya.
9. Untuk Nama, tinjau, dan buat:
 - a. Untuk nama Peran, masukkan nama untuk mengidentifikasi peran.
 - b. Untuk Deskripsi, masukkan penjelasan singkat untuk kebijakan tersebut.
 - c. (Opsional) Untuk Langkah 1: Pilih entitas tepercaya, pilih Edit untuk mengubah kode.
 - d. (Opsional) Untuk Langkah 2: Tambahkan izin, pilih Edit untuk mengubah kode.
 - e. (Opsional) Untuk Tambahkan tag, pilih Tambahkan tag untuk menambahkan tag ke sumber daya.
 - f. Pilih Buat peran.

Note

Di masa lalu, ada dua kebijakan terkelola untuk peran Armada Spot Amazon EC2.

- `AmazonEC2 SpotFleetRole`: Ini adalah kebijakan terkelola asli untuk peran Armada Spot. Namun, kami tidak lagi menyarankan Anda menggunakannya di AWS Batch. Kebijakan ini tidak mendukung penandaan Instance Spot di lingkungan komputasi, yang diperlukan untuk menggunakan peran terkait `AWSBatchServiceRoleForBatch` layanan. Jika sebelumnya Anda membuat peran Armada Spot dengan kebijakan ini, terapkan kebijakan baru yang direkomendasikan untuk peran tersebut. Untuk informasi selengkapnya, lihat [Instans Spot tidak ditandai pada pembuatan](#).
- `AmazonEC2 SpotFleetTaggingRole`: Peran ini menyediakan semua izin yang diperlukan untuk menandai Instans Spot Amazon EC2. Gunakan peran ini untuk mengizinkan penandaan Spot Instans di lingkungan komputasi AWS Batch.

Buat Peran Armada Spot Amazon EC2 dengan AWS CLI

Untuk membuat peran AmazonEC2 SpotFleetTaggingRole IAM untuk lingkungan komputasi Armada Spot Anda

1. Jalankan perintah berikut dengan AWS CLI.

```
$ aws iam create-role --role-name AmazonEC2SpotFleetTaggingRole \
  --assume-role-policy-document '{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "spotfleet.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

2. Untuk melampirkan kebijakan IAM SpotFleetTaggingRole terkelola AmazonEC2 ke SpotFleetTaggingRole peran AmazonEC2 Anda, jalankan perintah berikut dengan AWS CLI

```
$ aws iam attach-role-policy \
  --policy-arn \
  arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \
  --role-name \
  AmazonEC2SpotFleetTaggingRole
```

Untuk membuat IAM role yang terhubung dengan layanan **AWSServiceRoleForEC2Spot** untuk Spot Amazon EC2

Note

Jika peran terkait layanan **AWSServiceRoleForEC2Spot** IAM sudah ada, Anda akan melihat pesan galat yang menyerupai berikut ini.

```
An error occurred (InvalidInput) when calling the CreateServiceLinkedRole operation:
Service role name AWSServiceRoleForEC2Spot has been taken in this account,
please try a different suffix.
```

- Jalankan perintah berikut dengan AWS CLI.

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Untuk membuat IAM role yang terhubung dengan layanan **AWSServiceRoleForEC2SpotFleet** untuk Armada Spot Amazon EC2

Note

Jika peran terkait layanan **AWSServiceRoleForEC2SpotFleet** IAM sudah ada, Anda akan melihat pesan galat yang menyerupai berikut ini.

```
An error occurred (InvalidInput) when calling the CreateServiceLinkedRole operation:
Service role name AWSServiceRoleForEC2SpotFleet has been taken in this account,
please try a different suffix.
```

- Jalankan perintah berikut dengan AWS CLI.

```
$ aws iam create-service-linked-role --aws-service-name spotfleet.amazonaws.com
```

EventBridge Peran IAM

Amazon EventBridge memberikan aliran peristiwa sistem yang mendekati waktu nyata yang menggambarkan perubahan sumber daya. AWS Batch pekerjaan tersedia sebagai EventBridge target. Menggunakan aturan sederhana yang dapat Anda siapkan dengan cepat, Anda dapat mencocokkan peristiwa dan mengirimkan tugas AWS Batch untuk meresponsnya. Sebelum Anda

dapat mengirimkan AWS Batch pekerjaan dengan EventBridge aturan dan target, EventBridge harus memiliki izin untuk menjalankan AWS Batch pekerjaan atas nama Anda.

Note

Saat membuat aturan di EventBridge konsol yang menentukan AWS Batch antrian sebagai target, Anda dapat membuat peran ini. Untuk panduan contoh, lihat [AWS Batch pekerjaan sebagai EventBridge target](#). Anda dapat membuat EventBridge peran secara manual menggunakan konsol IAM. Untuk petunjuknya, lihat [Membuat peran menggunakan kebijakan kepercayaan khusus \(konsol\)](#) di Panduan Pengguna IAM.

Hubungan kepercayaan untuk peran EventBridge IAM Anda harus memberikan kepala `events.amazonaws.com` layanan kemampuan untuk mengambil peran tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Pastikan kebijakan yang dilampirkan pada peran EventBridge IAM memungkinkan `batch:SubmitJob` izin pada sumber daya Anda. Dalam contoh berikut, AWS Batch berikan kebijakan `AWSBatchServiceEventTargetRole` terkelola untuk memberikan izin ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
    }
  ]
}
```

```
    "Resource": "*"
  }
]
}
```

AWS Batch Aliran Acara untuk Amazon EventBridge

Anda dapat menggunakan streaming AWS Batch acara untuk Amazon EventBridge untuk menerima pemberitahuan mendekati waktu nyata mengenai status pekerjaan saat ini dalam antrian pekerjaan Anda.

Anda dapat menggunakan EventBridge untuk mendapatkan wawasan lebih lanjut tentang AWS Batch layanan Anda. Lebih khusus lagi, Anda dapat menggunakannya untuk memeriksa kemajuan pekerjaan, membuat alur kerja AWS Batch khusus, membuat laporan atau metrik penggunaan, atau membuat dasbor Anda sendiri. Dengan AWS Batch dan EventBridge, Anda tidak perlu penjadwalan dan kode pemantauan yang terus-menerus melakukan polling AWS Batch untuk perubahan status pekerjaan. Sebagai gantinya, Anda dapat menangani perubahan status AWS Batch pekerjaan secara asinkron menggunakan berbagai target Amazon. EventBridge ini termasuk AWS Lambda, Layanan Antrian Sederhana Amazon, Layanan Pemberitahuan Sederhana Amazon, atau Amazon Kinesis Data Streams.

Acara dari aliran AWS Batch acara dipastikan akan dikirimkan setidaknya satu kali. Jika duplikat peristiwa dikirim, peristiwa menyediakan informasi yang cukup untuk mengidentifikasi duplikat. Dengan begitu, Anda dapat membandingkan stempel waktu peristiwa dan status tugas.

AWS Batch pekerjaan tersedia sebagai EventBridge target. Dengan menggunakan aturan sederhana, Anda dapat mencocokkan acara dan mengirimkan AWS Batch pekerjaan sebagai tanggapan terhadapnya. Untuk informasi lebih lanjut, lihat [Apa itu EventBridge?](#) di Panduan EventBridge Pengguna Amazon. Anda juga dapat menggunakan EventBridge untuk menjadwalkan tindakan otomatis yang memicu sendiri pada waktu-waktu tertentu menggunakan cron atau menilai ekspresi. Untuk informasi selengkapnya, lihat [Membuat EventBridge aturan Amazon yang berjalan sesuai jadwal](#) di Panduan EventBridge Pengguna Amazon. Untuk panduan contoh, lihat [AWS Batch pekerjaan sebagai EventBridge target](#). Untuk informasi tentang menggunakan EventBridge Penjadwal, lihat [Menyiapkan EventBridge Penjadwal Amazon](#) di EventBridge Panduan Pengguna Amazon.

Topik

- [AWS Batch Event](#)
- [Menggunakan Notifikasi AWS Pengguna dengan AWS Batch](#)
- [AWS Batch pekerjaan sebagai EventBridge target](#)
- [Tutorial: Mendengarkan AWS Batch EventBridge](#)

- [Tutorial: Mengirim Pemberitahuan Amazon Simple Notification Service untuk Peristiwa Tugas Gagal](#)

AWS Batch Event

AWS Batch mengirimkan acara perubahan status pekerjaan ke EventBridge. AWS Batch melacak keadaan pekerjaan Anda. Jika status tugas yang dikirim sebelumnya berubah, peristiwa akan dipanggil. Misalnya, jika tugas di status RUNNING berpindah ke status FAILED. Peristiwa ini diklasifikasikan sebagai peristiwa perubahan status tugas.

Note

AWS Batch mungkin menambahkan jenis acara, sumber, dan detail lainnya di masa mendatang. Jika Anda secara terprogram deserialisasi data peristiwa JSON, pastikan aplikasi Anda siap untuk menangani properti yang tidak dikenal. Ini untuk menghindari masalah jika dan ketika properti tambahan ini ditambahkan.

Peristiwa Perubahan Status Tugas

Kapan saja pekerjaan yang ada (yang dikirimkan sebelumnya) berubah menyatakan, sebuah acara dibuat. Untuk informasi lebih lanjut tentang status AWS Batch pekerjaan, lihat [Status Job](#).

Note

Peristiwa tidak dibuat untuk pengiriman tugas awal.

Example Peristiwa Perubahan Status Tugas

Peristiwa perubahan status pekerjaan disampaikan dalam format berikut. detailBagian ini menyerupai [JobDetail](#) objek yang dikembalikan dari operasi [DescribeJobs](#) API di Referensi AWS Batch API. Untuk informasi selengkapnya tentang EventBridge parameter, lihat [Peristiwa dan Pola Peristiwa](#) di Panduan EventBridge Pengguna Amazon.

```
{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Job State Change",
```



```

"source": "aws.batch",
"account": "123456789012",
"time": "2022-01-11T23:36:40Z",
"region": "us-east-1",
"resources": [
  "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
],
"detail": {
  "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
  "jobName": "event-test",
  "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
  "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/
PexjEHappyPathCanary2JobQueue",
  "status": "RUNNABLE",
  "attempts": [],
  "createdAt": 1641944200058,
  "retryStrategy": {
    "attempts": 2,
    "evaluateOnExit": []
  },
  "dependsOn": [],
  "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-
run-job-definition:1",
  "parameters": {},
  "container": {
    "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
    "command": [
      "sleep",
      "600"
    ],
    "volumes": [],
    "environment": [],
    "mountPoints": [],
    "ulimits": [],
    "networkInterfaces": [],
    "resourceRequirements": [
      {
        "value": "2",
        "type": "VCPU"
      }, {
        "value": "256",
        "type": "MEMORY"
      }
    ]
  }
}

```

```

    ],
    "secrets": [],
  },
  "tags": {
    "resourceArn": "arn:aws:batch:us-
east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
  },
  "propagateTags": false,
  "platformCapabilities": []
}
}

```

Acara yang diblokir antrean pekerjaan

Kapan saja yang AWS Batch mendeteksi pekerjaan di RUNNABLE negara bagian dan dengan demikian memblokir antrian, sebuah peristiwa dibuat di Amazon CloudWatch Events. Untuk informasi selengkapnya tentang penyebab antrian diblokir yang didukung, lihat [contoh pesan antrean pekerjaan yang diblokir](#). Alasan yang sama juga tersedia di statusReason bidang dalam tindakan [DescribeJobs](#) API.

Example Peristiwa Perubahan Status Tugas

Peristiwa perubahan status pekerjaan disampaikan dalam format berikut. detailBagian ini menyerupai [JobDetail](#) objek yang dikembalikan dari operasi [DescribeJobs](#) API di Referensi AWS Batch API. Untuk informasi selengkapnya tentang EventBridge parameter, lihat [Peristiwa dan Pola Peristiwa](#) di Panduan EventBridge Pengguna Amazon.

```

{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Job Queue Blocked",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
    "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue"
  ],
  "detail": {

```

```

    "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
    "jobName": "event-test",
    "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/
PexjEHappyPathCanary2JobQueue",
    "status": "RUNNABLE",
    "statusReason": "blocked-reason"
    "attempts": [],
    "createdAt": 1641944200058,
    "retryStrategy": {
      "attempts": 2,
      "evaluateOnExit": []
    },
    "dependsOn": [],
    "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-
run-job-definition:1",
    "parameters": {},
    "container": {
      "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
      "command": [
        "sleep",
        "600"
      ],
      "volumes": [],
      "environment": [],
      "mountPoints": [],
      "ulimits": [],
      "networkInterfaces": [],
      "resourceRequirements": [
        {
          "value": "2",
          "type": "VCPU"
        }, {
          "value": "256",
          "type": "MEMORY"
        }
      ],
      "secrets": []
    },
    "tags": {
      "resourceArn": "arn:aws:batch:us-
east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
    },
  },

```

```
    "propagateTags": false,  
    "platformCapabilities": []  
  }  
}
```

Menggunakan Notifikasi AWS Pengguna dengan AWS Batch

Anda dapat menggunakan [Pemberitahuan AWS Pengguna](#) untuk mengatur saluran pengiriman agar mendapat pemberitahuan tentang AWS Batch peristiwa. Anda akan menerima notifikasi saat ada sebuah peristiwa yang cocok dengan sebuah aturan yang Anda tentukan. Anda dapat menerima notifikasi untuk peristiwa melalui beberapa saluran, termasuk email, notifikasi obrolan [AWS Chatbot](#), atau notifikasi push [AWS Console Mobile Application](#). Anda juga dapat melihat notifikasi di [Pusat Notifikasi Konsol](#). Notifikasi Pengguna mendukung agregasi, hal ini akan dapat mengurangi jumlah notifikasi yang Anda terima selama peristiwa-peristiwa tertentu.

Untuk mengkonfigurasi Pemberitahuan Pengguna di AWS Batch:

1. Buka [konsol AWS Batch](#).
2. Pilih Dasbor.
3. Pilih Konfigurasi Pemberitahuan.
4. Di Pemberitahuan AWS Pengguna, pilih Buat konfigurasi notifikasi.

Untuk informasi selengkapnya tentang cara mengonfigurasi dan melihat notifikasi pengguna, lihat [Memulai Pemberitahuan AWS Pengguna](#).

AWS Batch pekerjaan sebagai EventBridge target

Amazon EventBridge memberikan aliran peristiwa sistem yang mendekati real-time yang menjelaskan perubahan dalam sumber daya Amazon Web Services. Biasanya, AWS Batch di Amazon Elastic Container Service, Amazon Elastic Kubernetes Service, AWS dan pekerjaan Fargate tersedia sebagai target. EventBridge Dengan menggunakan aturan sederhana, Anda dapat mencocokkan acara dan mengirimkan AWS Batch pekerjaan sebagai tanggapan terhadapnya. Untuk informasi lebih lanjut, lihat [Apa itu EventBridge?](#) di Panduan EventBridge Pengguna Amazon.

Anda juga dapat menggunakan EventBridge untuk menjadwalkan tindakan otomatis yang dipanggil pada waktu-waktu tertentu menggunakan cron atau menilai ekspresi. Untuk informasi selengkapnya,

lihat [Membuat EventBridge aturan Amazon yang berjalan sesuai jadwal](#) di Panduan EventBridge Pengguna Amazon.

Untuk informasi tentang cara membuat aturan yang berjalan saat peristiwa cocok dengan pola peristiwa, lihat [Membuat EventBridge aturan Amazon yang bereaksi terhadap peristiwa](#) di Panduan EventBridge Pengguna Amazon.

Kasus penggunaan umum untuk AWS Batch pekerjaan sebagai EventBridge target meliputi kasus penggunaan berikut:

- Pekerjaan terjadwal terjadi pada interval waktu yang teratur. Contohnya, tugas cron terjadi hanya selama jam penggunaan rendah ketika Instans Spot Amazon EC2 lebih murah.
- AWS Batch Pekerjaan berjalan sebagai respons terhadap operasi API yang masuk CloudTrail. Misalnya, pekerjaan dikirimkan setiap kali objek diunggah ke bucket Amazon S3 yang ditentukan. Setiap kali ini terjadi, transformator EventBridge input melewati ember dan nama kunci objek ke AWS Batch parameter.

Note

Dalam skenario ini, semua sumber AWS daya terkait harus berada di Wilayah yang sama. Ini termasuk sumber daya seperti bucket Amazon S3, EventBridge aturan, dan CloudTrail log.

Sebelum Anda dapat mengirimkan AWS Batch pekerjaan dengan EventBridge aturan dan target, EventBridge layanan memerlukan beberapa izin untuk menjalankan AWS Batch pekerjaan. Saat membuat aturan di EventBridge konsol yang menentukan AWS Batch pekerjaan sebagai target, Anda juga dapat membuat peran ini. Untuk informasi lebih lanjut tentang pelaku utama layanan dan izin IAM yang diperlukan untuk peran ini, lihat [EventBridge Peran IAM](#).

Membuat AWS Batch pekerjaan terjadwal

Prosedur berikut mencakup cara membuat AWS Batch pekerjaan terjadwal dan peran EventBridge IAM yang diperlukan.

Untuk membuat AWS Batch pekerjaan terjadwal dengan EventBridge

Note

Prosedur ini berfungsi untuk semua AWS Batch di Amazon ECS, Amazon EKS, dan pekerjaan AWS Fargate.

1. Buka EventBridge konsol Amazon di <https://console.aws.amazon.com/events/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Aturan.
4. Pilih Buat aturan.
5. Untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat berisi hingga 64 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).

Note

Aturan tidak boleh memiliki nama yang sama dengan aturan lain di Wilayah yang sama dan di bus peristiwa yang sama.

6. (Opsional) Untuk Deskripsi, masukkan deskripsi untuk aturan.
7. Untuk bus acara, pilih bus acara yang ingin Anda kaitkan dengan aturan ini. Jika Anda ingin aturan ini cocok dengan peristiwa yang berasal dari akun Anda, pilih default. Ketika Layanan AWS di akun Anda memancarkan acara, itu selalu masuk ke bus acara default akun Anda.
8. (Opsional) Matikan aturan pada bus yang dipilih jika Anda tidak ingin segera menjalankan aturan.
9. Untuk jenis Aturan, pilih Jadwal.
10. Pilih Lanjutkan untuk membuat aturan atau Berikutnya.
11. Untuk pola Jadwal, lakukan salah satu hal berikut:
 - Pilih Jadwal berbutir halus yang berjalan pada waktu tertentu, seperti pukul 8:00 pagi. PST pada hari Senin pertama setiap bulan dan kemudian masukkan ekspresi cron. Untuk informasi selengkapnya, lihat [Ekspresi Cron di Panduan EventBridge Pengguna Amazon](#).
 - Pilih jadwal yang berjalan dengan tarif reguler, seperti setiap 10 menit. dan kemudian masukkan ekspresi tingkat.
12. Pilih Berikutnya.

13. Untuk Tipe target, pilih Layanan AWS.
14. Untuk Pilih target, pilih Antrian pekerjaan Batch. Kemudian, konfigurasi yang berikut ini:
 - Job queue (Antrean tugas): Masukkan Amazon Resource Name (ARN) dari antrean tugas untuk menjadwalkan tugas Anda.
 - Job definition (Ketentuan Tugas): Masukkan nama dan revisi atau ARN lengkap ketentuan tugas yang akan digunakan untuk tugas Anda.
 - Job name (Nama tugas): Masukkan nama untuk tugas Anda.
 - Array size (Ukuran array): (Opsional) Masukkan ukuran array untuk tugas Anda guna menjalankan lebih dari satu salinan. Untuk informasi selengkapnya, lihat [Tugas array](#).
 - Job attempts (Upaya tugas): (Opsional) Masukkan jumlah untuk mencoba kembali tugas Anda jika gagal. Untuk informasi selengkapnya, lihat [Percobaan kembali tugas secara otomatis](#).
15. Untuk jenis target antrian pekerjaan Batch, EventBridge memerlukan izin untuk mengirim peristiwa ke target. EventBridge dapat membuat peran IAM yang diperlukan agar aturan Anda berjalan. Lakukan salah satu dari langkah berikut ini:
 - Untuk membuat IAM role secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
 - Untuk menggunakan peran IAM yang telah Anda buat, pilih Gunakan peran yang ada.
16. (Opsional) Perluas Pengaturan tambahan.
 - a. Untuk Mengonfigurasi input target, pilih bagaimana teks dari suatu peristiwa diproses sebelum diteruskan ke target.
 - b. Untuk Usia maksimum acara, tentukan interval waktu untuk berapa lama peristiwa yang belum diproses disimpan.
 - c. Untuk mencoba lagi, masukkan berapa kali suatu peristiwa dicoba ulang.
 - d. Untuk antrian Dead-letter, pilih opsi bagaimana peristiwa yang belum diproses ditangani. Jika perlu, tentukan antrian Amazon SQS yang akan digunakan sebagai antrian huruf mati.
17. (Opsional) Pilih Tambahkan target lain untuk menambahkan target lain untuk aturan ini.
18. Pilih Berikutnya.
19. (Opsional) Untuk Tag, pilih Tambahkan tag baru untuk menambahkan label sumber daya untuk aturan. Untuk informasi selengkapnya, lihat [EventBridge Tag Amazon](#).
20. Pilih Berikutnya.
21. Untuk Meninjau dan membuat, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat aturan.

Untuk informasi selengkapnya tentang membuat aturan, lihat [Membuat EventBridge aturan Amazon yang berjalan sesuai jadwal](#) di Panduan EventBridge Pengguna Amazon.

Membuat aturan dengan pola acara

Prosedur berikut mencakup cara membuat aturan dengan pola acara.

Untuk membuat aturan yang mengirimkan acara ke target saat acara cocok dengan pola yang ditentukan

Note

Prosedur ini berfungsi untuk semua AWS Batch di Amazon ECS, Amazon EKS, dan pekerjaan AWS Fargate.

1. Buka EventBridge konsol Amazon di <https://console.aws.amazon.com/events/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Aturan.
4. Pilih Buat aturan.
5. Untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat berisi hingga 64 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).

Note

Aturan tidak boleh memiliki nama yang sama dengan aturan lain di Wilayah yang sama dan di bus peristiwa yang sama.

6. (Opsional) Untuk Deskripsi, masukkan deskripsi untuk aturan.
7. Untuk bus acara, pilih bus acara yang ingin Anda kaitkan dengan aturan ini. Jika Anda ingin aturan ini cocok dengan peristiwa yang berasal dari akun Anda, pilih default. Ketika Layanan AWS di akun Anda memancarkan acara, itu selalu masuk ke bus acara default akun Anda.
8. (Opsional) Matikan aturan pada bus yang dipilih jika Anda tidak ingin segera menjalankan aturan.
9. Untuk Tipe aturan, pilih Aturan dengan pola peristiwa.
10. Pilih Selanjutnya.

11. Untuk Sumber Acara, pilih AWS acara atau acara EventBridge mitra.
12. (Opsional) Untuk acara Sampel:
 - a. Untuk jenis acara Contoh, pilih AWS acara.
 - b. Untuk contoh peristiwa, pilih Batch Job State Change.
13. Untuk metode Creation, pilih Gunakan formulir pola.
14. Untuk pola Acara:
 - a. Untuk Sumber peristiwa, pilih Layanan AWS.
 - b. Untuk Layanan AWS, pilih Batch.
 - c. Untuk jenis Event, pilih Batch Job State Change.
15. Pilih Berikutnya.
16. Untuk Tipe target, pilih Layanan AWS.
17. Untuk Pilih target, pilih jenis target. Misalnya, pilih Antrian pekerjaan Batch. Kemudian tentukan yang berikut ini:
 - Job queue (Antrean tugas): Masukkan Amazon Resource Name (ARN) dari antrean tugas untuk menjadwalkan tugas Anda.
 - Job definition (Ketentuan Tugas): Masukkan nama dan revisi atau ARN lengkap ketentuan tugas yang akan digunakan untuk tugas Anda.
 - Job name (Nama tugas): Masukkan nama untuk tugas Anda.
 - Array size (Ukuran array): (Opsional) Masukkan ukuran array untuk tugas Anda guna menjalankan lebih dari satu salinan. Untuk informasi selengkapnya, lihat [Tugas array](#).
 - Job attempts (Upaya tugas): (Opsional) Masukkan jumlah untuk mencoba kembali tugas Anda jika gagal. Untuk informasi selengkapnya, lihat [Percobaan kembali tugas secara otomatis](#).
18. Untuk jenis target antrian pekerjaan Batch, EventBridge memerlukan izin untuk mengirim peristiwa ke target. EventBridge dapat membuat peran IAM yang diperlukan agar aturan Anda berjalan. Lakukan salah satu dari langkah berikut ini:
 - Untuk membuat IAM role secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
 - Untuk menggunakan peran IAM yang Anda buat sebelumnya, pilih Gunakan peran yang sudah ada.
19. (Opsional) Perluas Pengaturan tambahan.
 - a. Untuk Mengonfigurasi input target, pilih bagaimana teks dari suatu peristiwa diproses.

- b. Untuk Usia maksimum acara, tentukan interval waktu untuk berapa lama peristiwa yang belum diproses disimpan.
 - c. Untuk mencoba lagi, masukkan berapa kali suatu peristiwa dicoba ulang.
 - d. Untuk antrian Dead-letter, pilih opsi bagaimana peristiwa yang belum diproses ditangani. Jika perlu, tentukan antrian Amazon SQS yang akan digunakan sebagai antrian huruf mati.
20. (Opsional) Pilih Tambahkan target lain untuk menambahkan target tambahan.
 21. Pilih Berikutnya.
 22. (Opsional) Untuk Tag, pilih Tambahkan tag baru untuk menambahkan label sumber daya. Untuk informasi selengkapnya, lihat [EventBridge tag Amazon](#) di Panduan EventBridge Pengguna Amazon.
 23. Pilih Berikutnya.
 24. Untuk Meninjau dan membuat, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat aturan.

Untuk informasi selengkapnya tentang membuat aturan, lihat [Membuat EventBridge aturan Amazon yang bereaksi terhadap peristiwa](#) di Panduan EventBridge Pengguna Amazon.

Meneruskan informasi acara ke AWS Batch Target sesuai jadwal menggunakan transformator EventBridge input

Anda dapat menggunakan transformator EventBridge input untuk meneruskan informasi acara ke AWS Batch dalam pengiriman pekerjaan. Ini bisa sangat berharga jika Anda meminta pekerjaan sebagai hasil dari informasi AWS acara lainnya. Salah satu contohnya adalah upload objek ke bucket Amazon S3. Anda juga dapat menggunakan definisi pekerjaan dengan nilai substitusi parameter dalam perintah container. Transformator EventBridge input dapat memberikan nilai parameter berdasarkan data peristiwa.


Kemudian, setelah itu, Anda membuat target AWS Batch acara yang mem-parsing informasi dari acara yang memulainya dan mengubahnya menjadi `parameters` objek. Ketika tugas berjalan, parameter dari peristiwa pemicu disampaikan ke perintah kontainer tugas.

Note

Dalam skenario ini, semua sumber AWS daya (seperti bucket Amazon S3, EventBridge aturan, dan CloudTrail log) harus berada di Wilayah yang sama.

Untuk membuat AWS Batch target yang menggunakan transformator input

1. Buka EventBridge konsol Amazon di <https://console.aws.amazon.com/events/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Aturan.
4. Pilih Buat aturan.
5. Untuk Nama, tentukan nama unik untuk lingkungan komputasi Anda. Nama dapat berisi hingga 64 karakter. Nama dapat memuat huruf besar dan huruf kecil, angka, tanda hubung (-), dan garis bawah (_).

 Note

Aturan tidak dapat memiliki nama yang sama dengan aturan lain di bus acara yang sama Wilayah AWS dan di bus acara yang sama.

6. (Opsional) Untuk Deskripsi, masukkan deskripsi untuk aturan.
7. Untuk bus acara, pilih bus acara yang ingin Anda kaitkan dengan aturan ini. Jika Anda ingin aturan ini cocok dengan peristiwa yang berasal dari akun Anda, pilih default. Ketika Layanan AWS di akun Anda memancarkan acara, itu selalu masuk ke bus acara default akun Anda.
8. (Opsional) Matikan aturan pada bus yang dipilih jika Anda tidak ingin segera menjalankan aturan.
9. Untuk jenis Aturan, pilih Jadwal.
10. Pilih Lanjutkan untuk membuat aturan atau Berikutnya.
11. Untuk pola Jadwal, lakukan salah satu hal berikut:
 - Pilih Jadwal berbutir halus yang berjalan pada waktu tertentu, seperti pukul 8:00 pagi. PST pada hari Senin pertama setiap bulan dan kemudian masukkan ekspresi cron. Untuk informasi selengkapnya, lihat [Ekspresi Cron di](#) Panduan EventBridge Pengguna Amazon.
 - Pilih jadwal yang berjalan dengan tarif reguler, seperti setiap 10 menit. dan kemudian masukkan ekspresi tingkat.
12. Pilih Berikutnya.
13. Untuk Tipe target, pilih Layanan AWS.
14. Untuk Pilih target, pilih Antrian pekerjaan Batch. Kemudian, konfigurasi yang berikut ini:
 - Job queue (Antrean tugas): Masukkan Amazon Resource Name (ARN) dari antrean tugas untuk menjadwalkan tugas Anda.

- Job definition (Ketentuan Tugas): Masukkan nama dan revisi atau ARN lengkap ketentuan tugas yang akan digunakan untuk tugas Anda.
 - Job name (Nama tugas): Masukkan nama untuk tugas Anda.
 - Array size (Ukuran array): (Opsional) Masukkan ukuran array untuk tugas Anda guna menjalankan lebih dari satu salinan. Untuk informasi selengkapnya, lihat [Tugas array](#).
 - Job attempts (Upaya tugas): (Opsional) Masukkan jumlah untuk mencoba kembali tugas Anda jika gagal. Untuk informasi selengkapnya, lihat [Percobaan kembali tugas secara otomatis](#).
15. Untuk jenis target antrian pekerjaan Batch, EventBridge memerlukan izin untuk mengirim peristiwa ke target. EventBridge dapat membuat peran IAM yang diperlukan agar aturan Anda berjalan. Lakukan salah satu dari langkah berikut ini:
- Untuk membuat IAM role secara otomatis, pilih Buat peran baru untuk sumber daya khusus ini.
 - Untuk menggunakan peran IAM yang telah Anda buat, pilih Gunakan peran yang ada.
16. (Opsional) Perluas Pengaturan tambahan.
17. Di bagian Pengaturan tambahan, untuk Konfigurasi input target, pilih Input Transformer.
18. Pilih Konfigurasi transformator input.
19. (Opsional) Untuk acara Sampel:
- a. Untuk jenis acara Contoh, pilih AWS acara.
 - b. Untuk contoh peristiwa, pilih Batch Job State Change.
20. Di bagian Transformator input target, untuk Jalur input, tentukan nilai yang akan diuraikan dari peristiwa pemicu. Misalnya, untuk mengurai peristiwa Perubahan Status Pekerjaan Batch, gunakan format JSON berikut.

```
{
  "instance": "$.detail.jobId",
  "state": "$.detail.status"
}
```

21. Untuk Template, masukkan yang berikut ini.

```
{
  "instance": <jobId> ,
  "status": <status>
}
```

22. Pilih Konfirmasi.

23. Untuk Usia maksimum acara, tentukan interval waktu untuk berapa lama peristiwa yang belum diproses disimpan.
24. Untuk mencoba lagi, masukkan berapa kali suatu peristiwa dicoba ulang.
25. Untuk antrian Dead-letter, pilih opsi bagaimana peristiwa yang belum diproses ditangani. Jika perlu, tentukan antrian Amazon SQS yang akan digunakan sebagai antrian huruf mati.
26. (Opsional) Pilih Tambahkan target lain untuk menambahkan target tambahan.
27. Pilih Berikutnya.
28. (Opsional) Untuk Tag, pilih Tambahkan tag baru untuk menambahkan label sumber daya. Untuk informasi selengkapnya, lihat [EventBridge tag Amazon](#) di Panduan EventBridge Pengguna Amazon.
29. Pilih Berikutnya.
30. Untuk Meninjau dan membuat, tinjau langkah-langkah konfigurasi. Jika Anda perlu melakukan perubahan, pilih Edit. Setelah selesai, pilih Buat aturan.

Tutorial: Mendengarkan AWS Batch EventBridge

Dalam tutorial ini, Anda menyiapkan AWS Lambda fungsi sederhana yang mendengarkan acara AWS Batch pekerjaan dan menuliskannya ke aliran CloudWatch log Log.

Prasyarat

Tutorial ini mengasumsikan bahwa Anda memiliki lingkungan komputasi yang berfungsi dan antrian tugas yang siap untuk menerima tugas. Jika Anda tidak memiliki lingkungan komputasi yang berjalan dan antrian tugas untuk menangkap peristiwa, ikuti langkah-langkah di [Memulai dengan AWS Batch](#) untuk membuatnya. Pada akhir tutorial ini, secara opsional Anda dapat mengirim tugas ke antrian tugas ini untuk menguji bahwa Anda telah mengonfigurasi fungsi Lambda Anda dengan benar.

Langkah 1: Buat Fungsi Lambda

Dalam prosedur ini, Anda membuat fungsi Lambda sederhana yang berfungsi sebagai target untuk pesan alur kejadian AWS Batch.

Untuk membuat fungsi Lambda target

1. Buka konsol AWS Lambda di <https://console.aws.amazon.com/lambda/>.
2. Pilih Create function, Author dari awal.

3. Untuk Nama fungsi, masukkan `batch-event-stream-handler`.
4. Untuk Waktu pengoperasian, pilih Python 3.8.
5. Pilih Buat fungsi.
6. Di bagian Sumber kode, edit kode sampel agar sesuai dengan contoh berikut:

```
import json

def lambda_handler(event, _context):
    # _context is not used
    del _context
    if event["source"] != "aws.batch":
        raise ValueError("Function only supports input from events with a source
type of: aws.batch")

    print(json.dumps(event))
```

Ini adalah fungsi Python 3.8 sederhana yang mencetak peristiwa yang dikirim oleh AWS Batch. Jika semuanya dikonfigurasi dengan benar, di akhir tutorial ini, detail peristiwa muncul di aliran CloudWatch log Log yang terkait dengan fungsi Lambda ini.

7. Pilih Deploy.

Langkah 2: Daftarkan Aturan Peristiwa

Di bagian ini, Anda membuat aturan EventBridge acara yang menangkap peristiwa pekerjaan yang berasal dari AWS Batch sumber daya Anda. Aturan ini menangkap semua peristiwa yang berasal dari AWS Batch dalam akun tempat aturan ditentukan. Pesan tugas itu sendiri berisi informasi tentang sumber peristiwa, termasuk antrean tugas tempat itu dikirimkan. Anda dapat menggunakan informasi ini untuk mem-filter dan mengurutkan peristiwa secara terprogram.

Note

Jika Anda menggunakan aturan AWS Management Console untuk membuat acara, konsol secara otomatis menambahkan izin IAM EventBridge untuk memanggil fungsi Lambda Anda. Namun, jika Anda membuat aturan peristiwa menggunakan AWS CLI, Anda harus memberikan izin secara eksplisit. Untuk informasi selengkapnya, lihat [Peristiwa dan Pola Peristiwa](#) di Panduan EventBridge Pengguna Amazon.

Untuk membuat EventBridge aturan Anda

1. Buka EventBridge konsol Amazon di <https://console.aws.amazon.com/events/>.
2. Di panel navigasi, pilih Aturan.
3. Pilih Buat aturan.
4. Masukkan nama dan deskripsi untuk aturan.

Aturan tidak boleh memiliki nama yang sama dengan aturan lain di Wilayah yang sama dan di bus kejadian yang sama.

5. Untuk bus acara, pilih bus acara yang ingin Anda kaitkan dengan aturan ini. Jika Anda ingin aturan ini cocok dengan acara yang berasal dari akun Anda, pilih bus acara AWS default. Saat layanan AWS di akun Anda menghasilkan peristiwa, layanan tersebut akan selalu masuk ke bus peristiwa default akun Anda.
6. Untuk Tipe aturan, pilih Aturan dengan pola peristiwa.
7. Pilih Selanjutnya.
8. Untuk sumber acara, pilih Lainnya.
9. Untuk pola Acara, pilih Pola kustom (editor JSON).
10. Tempel pola peristiwa berikut ke area teks.

```
{
  "source": [
    "aws.batch"
  ]
}
```

Aturan ini berlaku di semua grup AWS Batch Anda dan untuk setiap peristiwa AWS Batch. Sebagai alternatif, Anda dapat membuat pola yang lebih spesifik untuk memfilter beberapa hasil.

11. Pilih Berikutnya.
12. Untuk Jenis Target, pilih Layanan AWS.
13. Untuk Pilih target, pilih fungsi Lambda, dan pilih fungsi Lambda Anda.
14. (Opsional) Untuk pengaturan tambahan, lakukan hal berikut:
 - a. Untuk Masa peristiwa maksimal, masukkan nilai antara satu menit (00:01) dan 24 jam (24:00).
 - b. Untuk Upaya coba lagi, masukkan angka antara 0 dan 185.

- c. Untuk antrian Dead-letter, pilih apakah akan menggunakan antrian Amazon SQS standar sebagai antrian huruf mati. EventBridge mengirimkan peristiwa yang cocok dengan aturan ini ke antrian huruf mati jika tidak berhasil dikirim ke target. Lakukan salah satu dari cara berikut:
 - Pilih Tidak ada untuk tidak menggunakan antrian surat mati.
 - Pilih Pilih antrian Amazon SQS di AWS akun saat ini untuk digunakan sebagai antrian huruf mati dan kemudian pilih antrian yang akan digunakan dari tarik-turun.
 - Pilih Pilihan antrian Amazon SQS di akun AWS lainnya sebagai antrian surat mati dan kemudian masukkan ARN antrian untuk menggunakannya. Anda harus melampirkan kebijakan berbasis sumber daya ke antrian yang memberikan EventBridge izin untuk mengirim pesan ke antrian tersebut. Untuk informasi selengkapnya, lihat [Memberikan izin untuk antrian huruf mati di Panduan Pengguna](#) Amazon. EventBridge
15. Pilih Berikutnya.
16. (Opsional) Masukkan satu atau lebih tanda untuk aturan. Untuk informasi selengkapnya, lihat [EventBridge tag Amazon](#) di Panduan EventBridge Pengguna Amazon.
17. Pilih Berikutnya.
18. Tinjau detail aturan dan pilih Buat aturan.

Langkah 3: Uji Konfigurasi Anda

Anda sekarang dapat menguji EventBridge konfigurasi Anda dengan mengirimkan pekerjaan ke antrian pekerjaan Anda. Jika semuanya dikonfigurasi dengan benar, fungsi Lambda Anda dipicu dan menulis data peristiwa ke aliran CloudWatch log Log untuk fungsi tersebut.

Untuk menguji konfigurasi EFA dan NCCL Anda

1. Buka konsol AWS Batch di <https://console.aws.amazon.com/batch/>.
2. Kirim tugas AWS Batch baru. Untuk informasi selengkapnya, lihat [Mengirimkan pekerjaan](#).
3. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
4. Di panel navigasi, pilih Logs (Log) dan pilih grup log untuk fungsi Lambda Anda (misalnya, `/aws/lambda/my-function`).
5. Pilih stream log untuk melihat data peristiwa.

Tutorial: Mengirim Pemberitahuan Amazon Simple Notification Service untuk Peristiwa Tugas Gagal

Dalam tutorial ini, Anda mengonfigurasi aturan EventBridge acara yang hanya menangkap peristiwa pekerjaan di mana pekerjaan telah pindah ke FAILED status. Pada akhir tutorial ini, Anda juga dapat secara opsional mengirim tugas ke antrean tugas ini. Ini untuk menguji bahwa Anda telah mengonfigurasi pemberitahuan Amazon SNS Anda dengan benar.

Prasyarat

Tutorial ini mengasumsikan bahwa Anda memiliki lingkungan komputasi yang berfungsi dan antrean tugas yang siap untuk menerima tugas. Jika Anda tidak memiliki lingkungan komputasi yang berjalan dan antrean tugas untuk menangkap peristiwa, ikuti langkah-langkah di [Memulai dengan AWS Batch](#) untuk membuatnya.

Langkah 1: Buat dan Berlangganan Topik Amazon SNS

Untuk tutorial ini, Anda mengonfigurasi topik Amazon SNS yang berfungsi sebagai target peristiwa untuk aturan peristiwa baru Anda.

Untuk membuat topik Amazon SNS

1. Buka konsol Amazon SNS di <https://console.aws.amazon.com/sns/v3/home>.
2. Pilih Topics (Topik), Create topic (Buat topik).
3. Untuk Tipe, pilih Standar.
4. Untuk Nama, masukkan **JobFailedAlert** dan pilih Buat topik.
5. Di JobFailedAlertlayar, pilih Buat langganan.
6. Untuk Protokol, pilih Email.
7. Untuk Titik akhir, masukkan alamat email yang Anda dapat mengaksesnya dan pilih Buat langganan.
8. Periksa akun email Anda dan tunggu sampai Anda menerima pesan email konfirmasi langganan. Saat Anda menerimanya, pilih Konfirmasi langganan.

Langkah 2: Daftarkan Aturan Peristiwa

Selanjutnya, daftarkan aturan peristiwa yang hanya menangkap peristiwa tugas gagal.

Untuk mendaftarkan EventBridge aturan Anda

1. Buka EventBridge konsol Amazon di <https://console.aws.amazon.com/events/>.
2. Di panel navigasi, pilih Aturan.
3. Pilih Buat aturan.
4. Masukkan nama dan deskripsi untuk aturan.

Aturan tidak boleh memiliki nama yang sama dengan aturan lain di Wilayah yang sama dan di bus kejadian yang sama.

5. Untuk bus acara, pilih bus acara yang ingin Anda kaitkan dengan aturan ini. Jika Anda ingin aturan ini cocok dengan acara yang berasal dari akun Anda, pilih bus acara AWS default. Ketika AWS layanan di akun Anda memancarkan suatu acara, itu selalu masuk ke bus acara default akun Anda.
6. Untuk Tipe aturan, pilih Aturan dengan pola peristiwa.
7. Pilih Selanjutnya.
8. Untuk sumber acara, pilih Lainnya.
9. Untuk pola Acara, pilih Pola kustom (editor JSON).
10. Tempel pola peristiwa berikut ke area teks.

```
{
  "detail-type": [
    "Batch Job State Change"
  ],
  "source": [
    "aws.batch"
  ],
  "detail": {
    "status": [
      "FAILED"
    ]
  }
}
```

Kode ini mendefinisikan EventBridge aturan yang cocok dengan peristiwa apa pun di mana status pekerjaan berada FAILED. Untuk informasi selengkapnya tentang pola peristiwa, lihat [Peristiwa dan Pola Peristiwa](#) di Panduan EventBridge Pengguna Amazon.

11. Pilih Berikutnya.

12. Untuk Jenis target, pilih Layanan AWS .
13. Untuk Pilih target, pilih topik SNS, dan untuk Topik, pilih JobFailedAlert.
14. (Opsional) Untuk pengaturan tambahan, lakukan hal berikut:
 - a. Untuk Masa peristiwa maksimal, masukkan nilai antara satu menit (00:01) dan 24 jam (24:00).
 - b. Untuk Upaya coba lagi, masukkan angka antara 0 dan 185.
 - c. Untuk antrian Dead-letter, pilih apakah akan menggunakan antrean Amazon SQS standar sebagai antrian huruf mati. EventBridge mengirimkan peristiwa yang cocok dengan aturan ini ke antrian huruf mati jika tidak berhasil dikirim ke target. Lakukan salah satu hal berikut:
 - Pilih Tidak ada untuk tidak menggunakan antrean surat mati.
 - Pilih Pilih antrean Amazon SQS di AWS akun saat ini untuk digunakan sebagai antrian huruf mati dan kemudian pilih antrian yang akan digunakan dari tarik-turun.
 - Pilih Pilih antrean Amazon SQS di AWS akun lain sebagai antrian huruf mati dan kemudian masukkan ARN antrian yang akan digunakan. Anda harus melampirkan kebijakan berbasis sumber daya ke antrian yang memberikan EventBridge izin untuk mengirim pesan ke antrean tersebut. Untuk informasi selengkapnya, lihat [Memberikan izin untuk antrean huruf mati di Panduan Pengguna](#) Amazon. EventBridge
15. Pilih Berikutnya.
16. (Opsional) Masukkan satu atau lebih tanda untuk aturan. Untuk informasi selengkapnya, lihat [EventBridge tag Amazon](#) di Panduan EventBridge Pengguna Amazon.
17. Pilih Berikutnya.
18. Tinjau detail aturan dan pilih Buat aturan.

Langkah 3: Uji Aturan Anda

Untuk menguji aturan Anda, kirim tugas yang keluar segera setelah dimulai dengan kode keluar bukan nol. Jika aturan acara Anda dikonfigurasi dengan benar, Anda akan menerima pesan email dalam beberapa menit dengan teks peristiwa.

Untuk menguji aturan

1. Buka AWS Batch konsol di <https://console.aws.amazon.com/batch/>.
2. Kirim AWS Batch pekerjaan baru. Untuk informasi selengkapnya, lihat [Mengirimkan pekerjaan](#). Untuk perintah tugas, ganti perintah ini untuk keluar dari kontainer dengan kode keluar 1.

```
/bin/sh, -c, 'exit 1'
```

3. Periksa email Anda untuk mengonfirmasi bahwa Anda menerima pemberitahuan email untuk notifikasi tugas yang gagal.

Aturan alternatif: Antrian Pekerjaan Batch Diblokir

Untuk membuat aturan acara yang memantau Antrian Pekerjaan Batch Diblokir, ulangi langkah-langkah dalam tutorial ini dengan perubahan berikut:

1. Pada Langkah 1, gunakan *BlockedJobQueue* sebagai nama topik.
2. Pada Langkah 2, gunakan pola berikut di editor JSON:

```
{
  "detail-type": [
    "Batch Job Queue Blocked"
  ],
  "source": [
    "aws.batch"
  ]
}
```

Menggunakan CloudWatch Log dengan AWS Batch

Anda dapat mengonfigurasi AWS Batch pekerjaan Anda pada sumber daya EC2 untuk mengirim informasi log terperinci dan metrik ke CloudWatch Log. Dengan melakukan ini, Anda dapat melihat log yang berbeda dari pekerjaan Anda di satu lokasi yang nyaman. Untuk informasi selengkapnya tentang CloudWatch Log, lihat [Apa itu CloudWatch Log Amazon?](#) di Panduan CloudWatch Pengguna Amazon.

Note

Secara default, CloudWatch Log dihidupkan untuk AWS kontainer Fargate.

Untuk mengaktifkan dan menyesuaikan logging CloudWatch Log, tinjau tugas konfigurasi satu kali berikut:

- Untuk lingkungan AWS Batch komputasi yang didasarkan pada sumber daya EC2, tambahkan kebijakan IAM ke peran. `ecsInstanceRole` Untuk informasi selengkapnya, lihat [the section called “Menambahkan kebijakan IAM CloudWatch Log”](#).
- Buat template peluncuran Amazon EC2 yang menyertakan CloudWatch pemantauan terperinci, lalu tentukan templat saat Anda membuat lingkungan AWS Batch komputasi. Anda juga dapat menginstal CloudWatch agen pada gambar yang ada dan kemudian menentukan gambar di wizard yang AWS Batch dijalankan pertama.
- (Opsional) Konfigurasi driver `awslogs`. Anda dapat menambahkan parameter yang mengubah perilaku default pada sumber daya EC2 dan Fargate. Untuk informasi selengkapnya, lihat [the section called “Menggunakan driver log `awslogs`”](#).

Menambahkan kebijakan IAM CloudWatch Log

Sebelum pekerjaan Anda dapat mengirim data log dan metrik terperinci ke CloudWatch Log, Anda harus membuat kebijakan IAM yang menggunakan API CloudWatch Log. Setelah Anda membuat kebijakan IAM, lampirkan ke `ecsInstanceRole` peran.

Note

Jika `ECS-CloudWatchLogs` kebijakan tidak dilampirkan ke `ecsInstanceRole` peran, metrik dasar masih dapat dikirim ke CloudWatch Log. Namun, metrik dasar tidak menyertakan data log atau metrik terperinci seperti ruang disk kosong.

AWS Batch lingkungan komputasi menggunakan sumber daya Amazon EC2. Saat Anda membuat lingkungan komputasi menggunakan wizard yang AWS Batch dijalankan pertama, AWS Batch buat `ecsInstanceRole` peran dan konfigurasi lingkungan dengannya.

Jika Anda tidak menggunakan wizard yang dijalankan pertama, Anda dapat menentukan `ecsInstanceRole` peran saat membuat lingkungan komputasi di AWS Command Line Interface atau AWS Batch API. Untuk informasi selengkapnya, lihat Referensi [AWS CLI Perintah](#) atau Referensi [AWS Batch API](#).

Untuk membuat kebijakan IAM **ECS-CloudWatchLogs**

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Kebijakan.
3. Pilih Buat kebijakan.
4. Pilih JSON, lalu masukkan kebijakan berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "arn:aws:logs:*:*:*"
      ]
    }
  ]
}
```

```
}
```

5. Pilih Berikutnya: Tanda.
6. (Opsional) Untuk Menambahkan tag, pilih Tambahkan tag untuk menambahkan tag ke kebijakan.
7. Pilih Berikutnya: Peninjauan.
8. Pada halaman Kebijakan tinjau, untuk Nama, masukkan **ECS-CloudWatchLogs**, lalu masukkan Deskripsi opsional.
9. Pilih Buat kebijakan.

Untuk melampirkan kebijakan **ECS-CloudWatchLogs** ke **ecsInstanceRole**

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Peran.
3. Pilih `ecsInstanceRole`. Jika peran tidak ada, ikuti prosedur [Peran instans Amazon ECS](#) untuk membuat peran.
4. Pilih Tambahkan Izin, lalu pilih Lampirkan kebijakan.
5. Pilih CloudWatchLogs kebijakan ECS dan kemudian pilih Lampirkan kebijakan.

Instal dan konfigurasi CloudWatch agen

Anda dapat membuat template peluncuran Amazon EC2 yang mencakup CloudWatch pemantauan. Untuk informasi selengkapnya, lihat [Meluncurkan instance dari template peluncuran](#) dan [Detail lanjutan](#) di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Anda juga dapat menginstal CloudWatch agen pada Amazon EC2 AMI yang ada dan kemudian menentukan gambar di wizard yang dijalankan AWS Batch pertama. Untuk informasi selengkapnya, lihat [Menginstal CloudWatch agen](#) dan [Memulai AWS Batch](#).

Note

Template peluncuran tidak didukung pada AWS Fargate sumber daya.

Lihat CloudWatch Log

Anda dapat melihat dan mencari CloudWatch log Log di fileAWS Management Console.

Note

Mungkin perlu beberapa menit agar data ditampilkan di CloudWatch Log.

Untuk melihat data CloudWatch Log

1. Buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Di panel navigasi kiri, pilih Log, lalu pilih Grup log.

<input type="checkbox"/>	Log group	Retention	Metric filters
<input type="checkbox"/>	/aws/batch/job	Never expire	-



3. Pilih grup log yang akan dilihat.



<input type="checkbox"/>	Log stream	Last event time
<input type="checkbox"/>	Test-jd/default/6622fe43-b2a3-4805-a0a6-3828329cc32b	2020-08-18T19:50:19.311Z
<input type="checkbox"/>	first-run-job-definition/default/86ed75ac-4f3f-4044-8fb0-dfd9c85ae6b2	2020-08-18T02:07:42.738Z
<input type="checkbox"/>	Test-jd/default/48f4a9dd-be07-4b43-8696-f0995eefe28b	2020-08-14T00:18:19.395Z
<input type="checkbox"/>	first-run-job-definition/default/d7d5ccf4-a0a0-44f1-bf36-35f2b3632912	2020-08-13T22:39:06.936Z
<input type="checkbox"/>	gpuJD/default/6ecf8ffb-ee03-4041-aa18-ab5e7a6dff0d	2019-03-26T08:48:39.637Z

4. Pilih pengaliran log yang akan dilihat. Secara default, pengaliran diidentifikasi oleh 200 karakter pertama dari nama tugas dan ID tugas Amazon ECS.

Tip

Untuk mengunduh data aliran log, pilih Tindakan.

Log events  **Actions** 

▶	Timestamp	Message
		There are older events to load. Load more.
▶	2020-08-17T19:07:42.738-07:00...	'hello world'
		No newer events at this moment. <i>Auto retry paused.</i> Resume

Gunakan CloudWatch Log untuk memantau pekerjaan AWS Batch Amazon EKS

Anda dapat menggunakan Amazon CloudWatch Logs untuk memantau, menyimpan, dan melihat semua file log Anda di satu lokasi. Menggunakan CloudWatch Log, Anda dapat mencari, memfilter, dan menganalisis data log dari berbagai sumber.

Anda dapat mengunduh Fluent Bit gambar AWS untuk yang menyertakan plugin untuk memantau pekerjaan AWS Batch Amazon EKS di CloudWatch Log. Fluent Bit adalah prosesor log open-source dan forwarder yang keduanya Docker dan kompatibel. Kubernetes Kami menyarankan Anda menggunakan Fluent Bit sebagai router log Anda karena sumber daya kurang intensif daripada Fluentd. Untuk informasi selengkapnya, lihat [Menggunakan gambar AWS for Fluent Bit](#).

Prasyarat

Lampirkan `CloudWatchAgentServerPolicy` kebijakan ke AWS Identity and Access Management kebijakan node pekerja Anda. Untuk informasi selengkapnya, lihat [Memverifikasi prasyarat](#).

Instal AWS untuk Fluent Bit

Untuk petunjuk tentang cara menginstal AWS Fluent Bit dan membuat CloudWatch grup, lihat [Menyiapkan Fluent Bit](#) atau [Mulai Cepat dengan CloudWatch agen dan Fluent Bit](#).

Tip

Ingat bahwa Fluent Bit menggunakan CPU .5 dan 100 MB memori pada AWS Batch node. Ini mengurangi total kapasitas yang tersedia untuk AWS Batch pekerjaan. Pertimbangkan ini ketika Anda mengukur pekerjaan Anda.

Aktifkan Fluent Bit untuk node AWS Batch

Untuk memastikan Fluent Bit logging DaemonSet berjalan pada node AWS Batch terkelola, ubah Fluent Bit DaemonSet toleransi:

```
tolerations:
```

```
- key: "batch.amazonaws.com/batch-node"  
  operator: "Exists"
```

AWS Batch CloudWatch Wawasan Kontainer

CloudWatch Container Insights mengumpulkan, mengorganisir, dan merangkum metrik dan log dari lingkungan komputasi dan pekerjaan Anda AWS Batch. Metrik termasuk CPU, memori, disk, dan pemanfaatan jaringan. Anda dapat menambahkan metrik ini ke CloudWatch dasbor.

Data operasional dikumpulkan sebagai log acara performa. Berikut adalah entry yang menggunakan skema JSON terstruktur yang memungkinkan data dengan kardinalitas tinggi diserap dan disimpan sesuai skala. Dari data ini, CloudWatch buat metrik agregat tingkat tinggi di lingkungan komputasi dan tingkat pekerjaan sebagai metrik. CloudWatch Untuk informasi selengkapnya, lihat [Log Terstruktur Wawasan Penampung untuk Amazon ECS](#) di CloudWatch Panduan Pengguna Amazon.

Important

CloudWatch Wawasan Kontainer dibebankan sebagai metrik khusus oleh CloudWatch Untuk informasi selengkapnya, lihat [harga CloudWatch Acara Amazon](#)

Aktifkan Wawasan Kontainer

Anda dapat mengaktifkan Wawasan Kontainer untuk lingkungan AWS Batch komputasi.

1. Buka [konsol AWS Batch](#).
2. Pilih Lingkungan Komputasi.
3. Pilih lingkungan komputasi yang Anda inginkan.
4. Untuk Wawasan Kontainer, aktifkan Wawasan Kontainer untuk komputasi lingkungan.

Tip

Anda dapat memilih interval default untuk menggabungkan metrik atau membuat kustom interval.

Secara default, metrik berikut ditampilkan. Untuk daftar lengkap metrik Amazon ECS Container Insights, lihat Metrik [Wawasan Kontainer Amazon ECS di Panduan](#) Pengguna Amazon. CloudWatch

- **JobCount**— Jumlah pekerjaan yang berjalan di lingkungan komputasi.

- **ContainerInstanceCount**— Jumlah instans Amazon Elastic Compute Cloud yang menjalankan agen Amazon ECS dan terdaftar di lingkungan komputasi.
- **MemoryReserved**— Memori yang disediakan oleh pekerjaan lingkungan komputasi. Metrik ini dikumpulkan hanya untuk pekerjaan yang memiliki reservasi memori yang ditentukan dalam definisi pekerjaan mereka.
- **MemoryUtilized**— Memori yang digunakan oleh pekerjaan lingkungan komputasi. Metrik ini dikumpulkan hanya untuk pekerjaan yang memiliki reservasi memori yang ditentukan dalam definisi pekerjaan mereka.
- **CpuReserved**— Unit CPU yang dicadangkan oleh pekerjaan lingkungan komputasi. Metrik ini dikumpulkan hanya untuk pekerjaan yang memiliki reservasi CPU yang ditentukan dalam definisi pekerjaan mereka.
- **CpuUtilized**— Unit CPU yang digunakan oleh pekerjaan di lingkungan komputasi. Metrik ini dikumpulkan hanya untuk pekerjaan yang memiliki reservasi CPU yang ditentukan dalam definisi pekerjaan mereka.
- **NetworkRxBytes**- Jumlah byte yang diterima. Metrik ini hanya tersedia untuk kontainer dalam pekerjaan yang menggunakan mode jaringan awsvpc atau jembatan.
- **NetworkTxBytes**— Jumlah byte yang ditransmisikan. Metrik ini hanya tersedia untuk kontainer dalam pekerjaan yang menggunakan mode jaringan awsvpc atau jembatan.
- **StorageReadBytes**— Jumlah byte yang dibaca dari penyimpanan.
- **StorageWriteBytes**— Jumlah byte yang ditulis ke penyimpanan.

Mencatat AWS Batch panggilan API dengan AWS CloudTrail

AWS Batch terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di AWS Batch. CloudTrail menangkap semua panggilan API untuk AWS Batch sebagai peristiwa. Panggilan yang direkam mencakup panggilan dari AWS Batch konsol dan panggilan kode ke operasi API AWS Batch ini. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk AWS Batch. 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 AWS Batch, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

AWS Batch Informasi di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di AWS Batch, aktivitas tersebut dicatat dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS. Untuk informasi selengkapnya, lihat [Melihat Acara dengan Riwayat CloudTrail Acara](#).

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

- [Gambaran umum untuk Membuat Jejak](#)
- [CloudTrail Layanan dan Integrasi yang Didukung](#)
- [Mengkonfigurasi Notifikasi Amazon SNS untuk CloudTrail](#)
- [Menerima File CloudTrail Log dari Beberapa Wilayah](#) dan [Menerima File CloudTrail Log dari Beberapa Akun](#)

Semua AWS Batch tindakan dicatat oleh CloudTrail dan didokumentasikan di <https://docs.aws.amazon.com/batch/latest/APIReference/>. Misalnya, panggilan ke [SubmitJob](#), [ListJobs](#), dan [DescribeJobs](#) menghasilkan entri dalam berkas log CloudTrail .

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

- Apakah permintaan dibuat dengan kredensial akar atau pengguna IAM.
- Apakah permintaan dibuat dengan kredensial keamanan sementara untuk suatu peran atau pengguna gabungan.
- Apakah permintaan dibuat oleh layanan AWS lain.

Untuk informasi lain, lihat [Elemen userIdentity CloudTrail](#) .

Memahami Entri File Berkas Log AWS Batch

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. Sebuah kejadian mewakili permintaan tunggal dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. Berkas log CloudTrail bukan merupakan jejak tumpukan terurut dari panggilan API publik, sehingga tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan [CreateComputeEnvironment](#) tindakan.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:admin",
    "arn": "arn:aws:sts::012345678910:assumed-role/Admin/admin",
    "accountId": "012345678910",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-12-20T00:48:46Z"
      },
      "sessionIssuer": {
```

```
    "type": "Role",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::012345678910:role/Admin",
    "accountId": "012345678910",
    "userName": "Admin"
  }
}
},
"eventTime": "2017-12-20T00:48:46Z",
"eventSource": "batch.amazonaws.com",
"eventName": "CreateComputeEnvironment",
"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.1",
"userAgent": "aws-cli/1.11.167 Python/2.7.10 Darwin/16.7.0 botocore/1.7.25",
"requestParameters": {
  "computeResources": {
    "subnets": [
      "subnet-5eda8e04"
    ],
    "tags": {
      "testBatchTags": "CLI testing CE"
    },
    "desiredvCpus": 0,
    "minvCpus": 0,
    "instanceTypes": [
      "optimal"
    ],
    "securityGroupIds": [
      "sg-aba9e8db"
    ],
    "instanceRole": "ecsInstanceRole",
    "maxvCpus": 128,
    "type": "EC2"
  },
  "state": "ENABLED",
  "type": "MANAGED",
  "computeEnvironmentName": "Test"
},
"responseElements": {
  "computeEnvironmentName": "Test",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-environment/
Test"
},
"requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
```



```
"eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",  
"readOnly": false,  
"eventType": "AwsApiCall",  
"recipientAccountId": "012345678910"  
}
```

Membuat cloud pribadi virtual

Sumber daya komputasi di lingkungan komputasi Anda memerlukan akses jaringan eksternal untuk berkomunikasi dengan dan titik akhir AWS Batch layanan Amazon ECS. Namun, Anda mungkin memiliki pekerjaan yang ingin Anda jalankan di subnet pribadi. Untuk memiliki fleksibilitas untuk menjalankan pekerjaan baik di subnet publik atau swasta, buat VPC yang memiliki subnet publik dan swasta.

Anda dapat menggunakan Amazon Virtual Private Cloud (Amazon VPC) untuk meluncurkan AWS sumber daya ke jaringan virtual yang Anda tentukan. Topik ini menyediakan tautan ke wizard VPC Amazon dan daftar opsi untuk dipilih.

Buat VPC

Untuk informasi tentang cara membuat VPC Amazon, lihat [Membuat VPC hanya di Panduan Pengguna VPC](#) Amazon dan gunakan tabel berikut untuk menentukan opsi apa yang harus dipilih.

Opsi	Nilai	
Sumber daya untuk dibuat	Hanya VPC	
Nama	Secara opsional berikan nama untuk VPC Anda.	
Blok CIDR IPv4	Input manual IPv4 CIDR Ukuran blok CIDR harus memiliki ukuran antara /16 dan /28.	
Blok CIDR IPv6	Tidak ada blok IPv6 CIDR	
Penghunian	Default	

Untuk informasi lebih lanjut tentang Amazon VPC, lihat [Apa itu Amazon VPC?](#) dalam Panduan Pengguna Amazon VPC.

Langkah Berikutnya

Setelah Anda membuat VPC, pertimbangkan langkah-langkah berikut:

- Buat grup keamanan untuk sumber daya publik dan privat Anda jika sumber daya memerlukan akses jaringan masuk. Untuk informasi selengkapnya, lihat [Bekerja dengan grup keamanan](#) di Panduan Pengguna Amazon VPC.
- Buat lingkungan komputasi terkelola AWS Batch yang meluncurkan sumber daya komputasi ke VPC baru Anda. Untuk informasi selengkapnya, lihat [Membuat lingkungan komputasi](#). Jika Anda menggunakan wizard pembuatan lingkungan komputasi di AWS Batch konsol, Anda dapat menentukan VPC yang baru saja Anda buat dan subnet publik atau pribadi yang ingin Anda luncurkan instance.
- Buat antrean AWS Batch pekerjaan yang dipetakan ke lingkungan komputasi baru Anda. Untuk informasi selengkapnya, lihat [Membuat antrean tugas](#).
- Buat ketentuan tugas untuk menjalankan tugas Anda. Untuk informasi selengkapnya, lihat [Membuat definisi pekerjaan simpul tunggal](#).
- Kirim tugas dengan ketentuan tugas ke antrean tugas baru Anda. Pekerjaan ini mendarat di lingkungan komputasi yang Anda buat dengan VPC dan subnet baru Anda. Untuk informasi selengkapnya, lihat [Mengirimkan pekerjaan](#).

Keamanan di AWS Batch

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](#) mendeskripsikannya 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 Kepatuhan](#) . Untuk mempelajari tentang program kepatuhan yang berlaku AWS Batch, lihat [AWS Layanan dalam Lingkup oleh AWS Layanan Program Kepatuhan](#) .
- Keamanan di cloud – Tanggung jawab Anda ditentukan menurut layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS Batch. Topik berikut menunjukkan cara mengonfigurasi AWS Batch untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan AWS Batch sumber daya Anda.

Topik

- [Identity and Access Management untuk AWS Batch](#)
- [Akses AWS Batch menggunakan titik akhir antarmuka](#)
- [Validasi Kepatuhan untuk AWS Batch](#)
- [Keamanan Infrastruktur di AWS Batch](#)

Identity and Access Management untuk AWS Batch

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang

dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. AWS Batch IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana AWS Batch bekerja dengan IAM](#)
- [AWS Batch eksekusi peran IAM](#)
- [Contoh kebijakan berbasis identitas untuk AWS Batch](#)
- [Pencegahan confused deputy lintas layanan](#)
- [Memecahkan masalah AWS Batch identitas dan akses](#)
- [Menggunakan peran terkait layanan untuk AWS Batch](#)
- [AWS kebijakan terkelola untuk AWS Batch](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan. AWS Batch

Pengguna layanan — Jika Anda menggunakan AWS Batch layanan untuk melakukan pekerjaan Anda, maka administrator Anda memberi Anda kredensial dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak AWS Batch fitur untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara mengelola akses dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di AWS Batch, lihat [Memecahkan masalah AWS Batch identitas dan akses](#).

Administrator layanan — Jika Anda bertanggung jawab atas AWS Batch sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS Batch. Tugas Anda adalah menentukan AWS Batch fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM AWS Batch, lihat [Bagaimana AWS Batch bekerja dengan IAM](#).

Administrator IAM – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih detail tentang cara Anda menulis kebijakan untuk mengelola akses ke AWS Batch. Untuk melihat contoh kebijakan AWS Batch berbasis identitas yang dapat Anda gunakan di IAM, lihat [Contoh kebijakan berbasis identitas untuk AWS Batch](#)

Mengautentikasi dengan identitas

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

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

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

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

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

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk

membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Identitas terfederasi

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

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

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

Pengguna dan grup IAM

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

[Grup IAM](#) adalah identitas yang menentukan kumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

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

Peran IAM

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

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas gabungan, Anda dapat membuat peran dan menentukan izin untuk peran tersebut. Saat identitas terfederasi mengautentikasi, identitas tersebut akan dikaitkan dengan peran dan diberi izin yang ditentukan oleh peran tersebut. Untuk informasi tentang peran-peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) dalam Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda perlu mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengorelasikan izin yang diatur ke peran dalam IAM. Untuk informasi tentang rangkaian izin, lihat [Rangkaian izin](#) dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (pengguna utama tepercaya) dengan akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut

menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.

- Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat permintaan FAS, lihat [Teruskan sesi akses](#).
- Peran layanan – Peran layanan adalah [peran IAM](#) yang diambil oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan dapat menggunakan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensi sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2](#) dalam Panduan Pengguna IAM.

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

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber

daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan dapat menentukan permintaan yang diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan konten dokumen kebijakan JSON, lihat [Gambaran umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

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

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

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk operasi. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan 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.

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

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan

kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya 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

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

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (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.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, silakan lihat [Gambaran umum daftar kontrol akses \(ACL\)](#) di Panduan Developer Layanan Penyimpanan Ringkas Amazon.

Tipe kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda berdasarkan tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan di mana Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang Principal tidak dibatasi oleh batasan izin. Penolakan secara eksplisit terhadap salah satu kebijakan ini akan mengesampingkan izin tersebut. Untuk informasi selengkapnya tentang batasan izin, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di sebuah organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS

Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations .

- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda teruskan sebagai parameter saat Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Beberapa jenis kebijakan

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

Bagaimana AWS Batch bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses AWS Batch, pelajari fitur IAM yang tersedia untuk digunakan. AWS Batch

Fitur IAM yang dapat Anda gunakan AWS Batch

Fitur IAM	AWS Batch dukungan
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci persyaratan kebijakan	Ya
ACL	Tidak
ABAC (tanda dalam kebijakan)	Ya
Kredensial sementara	Ya

Fitur IAM	AWS Batch dukungan
Izin pengguna utama	Ya
Peran layanan	Ya
Peran terkait layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara AWS Batch dan AWS layanan lain bekerja dengan sebagian besar fitur IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Kebijakan berbasis identitas untuk AWS Batch

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 Anda gunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan IAM JSON](#) dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk AWS Batch

Untuk melihat contoh kebijakan AWS Batch berbasis identitas, lihat [Contoh kebijakan berbasis identitas untuk AWS Batch](#)

Tindakan kebijakan untuk AWS Batch

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 AWS Batch tindakan, lihat [Tindakan yang Ditentukan oleh AWS Batch](#) dalam Referensi Otorisasi Layanan.

Tindakan kebijakan AWS Batch menggunakan awalan berikut sebelum tindakan:

```
batch
```

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

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

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

```
"Action": "batch:Describe*"
```

Untuk melihat contoh kebijakan AWS Batch berbasis identitas, lihat [Contoh kebijakan berbasis identitas untuk AWS Batch](#)

Sumber daya kebijakan untuk AWS Batch

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": "*" 
```

Untuk melihat daftar jenis AWS Batch sumber daya dan ARNnya, lihat Sumber [Daya yang Ditentukan oleh AWS Batch dalam Referensi](#) Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat menentukan ARN setiap sumber daya, lihat [Tindakan yang Ditentukan oleh AWS Batch](#).

Untuk melihat contoh kebijakan AWS Batch berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk AWS Batch](#)

Kunci syarat kebijakan untuk AWS Batch

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.

Untuk melihat daftar kunci AWS Batch kondisi, lihat [Kunci Kondisi untuk AWS Batch](#) Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang Ditentukan oleh AWS Batch](#).

Untuk melihat contoh kebijakan AWS Batch berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk AWS Batch](#)

Kontrol akses berbasis atribut (ABAC) dengan AWS Batch

Mendukung ABAC (tanda dalam kebijakan) Ya

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

Menggunakan kredensi sementara dengan AWS Batch

Mendukung kredensial sementara	Ya
--------------------------------	----

Beberapa Layanan AWS tidak berfungsi saat Anda masuk menggunakan kredensi 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 kredensi 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 kredensi sementara tersebut untuk mengakses AWS . AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi lebih lanjut, lihat [Kredensial keamanan sementara di IAM](#).

Izin pelaku utama lintas layanan untuk AWS Batch

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

Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai tindakan lain di layanan yang berbeda. FAS menggunakan izin dari

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

Peran layanan untuk AWS Batch

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.

Warning

Mengubah izin untuk peran layanan dapat merusak AWS Batch fungsionalitas. Edit peran layanan hanya jika AWS Batch memberikan bimbingan untuk melakukannya.

Peran terkait layanan untuk AWS Batch

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

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

Untuk detail tentang pembuatan atau pengelolaan peran terkait layanan, lihat [Layanan AWS yang berfungsi dengan IAM](#). Temukan sebuah layanan dalam tabel yang memiliki Yes di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

AWS Batch eksekusi peran IAM

Peran eksekusi memberikan izin kepada kontainer Amazon ECS dan AWS Fargate agen untuk melakukan panggilan AWS API atas nama Anda.

Note

Peran eksekusi didukung oleh agen kontainer Amazon ECS versi 1.16.0 dan yang lebih baru.

IAM role eksekusi diperlukan bergantung pada keperluan tugas Anda. Anda dapat memiliki beberapa peran eksekusi untuk berbagai tujuan dan layanan yang terkait dengan akun Anda.

Note

Untuk informasi tentang peran instans Amazon ECS, lihat [Peran instans Amazon ECS](#). Untuk informasi tentang peran layanan, lihat [Bagaimana AWS Batch bekerja dengan IAM](#).

Amazon ECS menyediakan kebijakan AmazonECSTaskExecutionRolePolicy terkelola. Kebijakan ini berisi izin yang diperlukan untuk kasus penggunaan umum yang dijelaskan di atas. Mungkin perlu menambahkan kebijakan sebaris ke peran eksekusi Anda untuk kasus penggunaan khusus yang diuraikan di bawah ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

```
}
```

Anda dapat menggunakan prosedur berikut untuk memeriksa apakah akun Anda sudah memiliki peran eksekusi dan melampirkan kebijakan IAM terkelola, jika diperlukan.

Untuk memeriksa **ecsTaskExecutionRole** di konsol IAM

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Peran.
3. Cari daftar peran untuk `ecsTaskExecutionRole`. Jika Anda tidak dapat menemukan peran, lihat [Membuat IAM role eksekusi](#). Jika Anda menemukan peran, pilih peran untuk melihat kebijakan terlampir.
4. Pada tab Izin, verifikasi bahwa kebijakan terkelola `TaskExecutionRolePolicyAmazonECS` dilampirkan ke peran. Jika kebijakan terlampir, peran eksekusi Anda sudah dikonfigurasi dengan benar. Jika tidak, ikuti langkah-langkah di bawah ini untuk melampirkan kebijakan.
 - a. Pilih Tambahkan izin, lalu pilih Lampirkan kebijakan.
 - b. Cari AmazonECS `TaskExecutionRolePolicy`.
 - c. Centang kotak di sebelah kiri kebijakan AmazonECS dan pilih `TaskExecutionRolePolicy` Lampirkan kebijakan.
5. Pilih Hubungan kepercayaan.
6. Verifikasi bahwa hubungan kepercayaan berisi kebijakan berikut. Jika hubungan kepercayaan cocok dengan kebijakan di bawah ini, peran dikonfigurasi dengan benar. Jika hubungan kepercayaan tidak cocok, pilih Edit kebijakan kepercayaan, masukkan yang berikut ini, dan pilih Perbarui kebijakan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
}
```

Membuat IAM role eksekusi

Jika akun Anda belum memiliki peran eksekusi, gunakan langkah-langkah berikut untuk membuat peran tersebut.

Untuk membuat IAM role **ecsTaskExecutionRole**

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Peran.
3. Pilih Buat peran.
4. Untuk jenis entitas Tepercaya, pilih Layanan AWS.
5. Untuk Service atau use case, pilih EC2. Kemudian pilih EC2 lagi.
6. Pilih Berikutnya.
7. Untuk kebijakan Izin, cari TaskExecutionRolePolicyAmazonECS.
8. Pilih kotak centang di sebelah kiri kebijakan AmazonECS, TaskExecutionRolePolicy lalu pilih Berikutnya.
9. Untuk Nama Peran, ecsTaskExecutionRole masukkan lalu pilih Buat peran.

Contoh kebijakan berbasis identitas untuk AWS Batch

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi AWS Batch sumber daya. Mereka juga tidak dapat melakukan tugas dengan menggunakan AWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS API. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh AWS Batch, termasuk format ARN untuk setiap jenis sumber daya, lihat [Tindakan, Sumber Daya, dan Kunci Kondisi untuk AWS Batch](#) dalam Referensi Otorisasi Layanan.

Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan AWS Batch konsol](#)
- [Izinkan pengguna melihat izin mereka sendiri](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus AWS Batch sumber daya di akun Anda. Tindakan ini dikenai biaya untuk Akun AWS Anda. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat [kebijakan yang dikelola AWS](#) atau [kebijakan yang dikelola AWS untuk fungsi pekerjaan](#) di Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukan ini dengan menentukan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk menerapkan izin, lihat [Kebijakan dan izin di IAM](#) di Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Syarat](#) di Panduan Pengguna IAM.
- Menggunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda guna memastikan izin yang aman dan berfungsi – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang

dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat [validasi kebijakan Analizer Akses IAM](#) di Panduan Pengguna IAM.

- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk mewajibkan MFA saat operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Mengonfigurasi akses API yang dilindungi MFA](#) di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan di IAM](#) di Panduan Pengguna IAM.

Menggunakan AWS Batch konsol

Untuk mengakses AWS Batch konsol, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang AWS Batch sumber daya di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebaliknya, izinkan akses hanya ke tindakan yang cocok dengan operasi API yang coba dilakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan AWS Batch konsol, lampirkan juga kebijakan AWS Batch ConsoleAccess atau ReadOnly AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat [Menambahkan izin ke pengguna](#) di Panduan Pengguna IAM.

Izinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan para pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
```

```

    "Effect": "Allow",
    "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

Pencegahan confused deputy lintas layanan

Masalah confused deputy adalah masalah keamanan saat entitas yang tidak memiliki izin untuk melakukan suatu tindakan dapat memaksa entitas yang lebih berhak untuk melakukan tindakan tersebut. 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 menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan dalam kebijakan sumber daya untuk membatasi izin yang AWS Batch memberikan layanan lain ke sumber daya. Jika `aws:SourceArn` nilainya tidak berisi ID akun, seperti ARN bucket Amazon S3, Anda harus menggunakan kedua kunci konteks kondisi global untuk membatasi izin. Jika Anda menggunakan kunci konteks kondisi global dan nilai `aws:SourceArn` berisi ID akun, nilai `aws:SourceAccount` dan akun dalam nilai `aws:SourceArn` harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama. Gunakan `aws:SourceArn` jika Anda hanya ingin satu sumber daya dikaitkan dengan akses lintas layanan. Gunakan `aws:SourceAccount` jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut dikaitkan dengan penggunaan lintas layanan.

Nilai `aws:SourceArn` harus menjadi sumber daya yang AWS Batch menyimpan.

Cara paling efektif untuk melindungi dari masalah confused deputy adalah dengan menggunakan kunci konteks kondisi global `aws:SourceArn` dengan ARN lengkap sumber daya. Jika Anda tidak mengetahui ARN lengkap sumber daya atau jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan karakter wildcard (*) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:service:*:123456789012:*`.

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan AWS Batch untuk mencegah masalah wakil yang membingungkan.

Contoh 1: Peran untuk mengakses hanya satu lingkungan komputasi

Peran berikut hanya dapat digunakan untuk mengakses satu lingkungan komputasi. Nama pekerjaan harus ditentukan * karena antrian pekerjaan dapat dikaitkan dengan beberapa lingkungan komputasi.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

```

    },
    "ArnLike": {
      "aws:SourceArn": [
        "arn:aws:batch:us-east-1:123456789012:compute-environment/testCE",
        "arn:aws:batch:us-east-1:123456789012:job/*"
      ]
    }
  }
}
]
}

```

Contoh 2: Peran untuk mengakses beberapa lingkungan komputasi

Peran berikut dapat digunakan untuk mengakses beberapa lingkungan komputasi. Nama pekerjaan harus ditentukan *** karena antrian pekerjaan dapat dikaitkan dengan beberapa lingkungan komputasi.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/*",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}

```

Memecahkan masalah AWS Batch identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS Batch dan IAM.

Topik

- [Saya tidak diotorisasi untuk melakukan tindakan di AWS Batch](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses AWS Batch sumber daya saya](#)

Saya tidak diotorisasi untuk melakukan tindakan di AWS Batch

Jika AWS Management Console memberitahu Anda bahwa Anda tidak berwenang untuk melakukan tindakan, maka Anda harus menghubungi administrator Anda untuk bantuan. Administrator adalah orang yang memberikan nama pengguna dan kata sandi kepada Anda.

Contoh kesalahan berikut terjadi ketika pengguna `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya `my-example-widget` fiktif, tetapi tidak memiliki izin `batch:GetWidget` fiktif.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
batch:GetWidget on resource: my-example-widget
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakan miliknya agar dia dapat mengakses sumber daya `my-example-widget` dengan menggunakan tindakan `batch:GetWidget`. Untuk informasi selengkapnya tentang pemberian izin untuk meneruskan peran, lihat [Memberikan izin pengguna untuk meneruskan peran ke layanan](#). AWS

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran AWS Batch.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di AWS Batch. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses AWS Batch sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau pengguna di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi pengguna akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mempelajari apakah AWS Batch mendukung fitur-fitur ini, lihat [Bagaimana AWS Batch bekerja dengan IAM](#).
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki](#) di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Memberikan akses kepada pengguna eksternal yang sah \(federasi identitas\)](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Perbedaan antara peran IAM dan kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

Menggunakan peran terkait layanan untuk AWS Batch

AWS Batch menggunakan AWS Identity and Access Management peran [terkait layanan](#) (IAM). Peran terkait layanan adalah jenis unik peran IAM yang ditautkan langsung ke AWS Batch Peran terkait layanan telah ditentukan sebelumnya oleh AWS Batch dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

Peran terkait layanan membuat pengaturan AWS Batch lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. AWS Batch mendefinisikan izin peran terkait layanan, dan kecuali ditentukan lain, hanya AWS Batch dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, serta bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

Note

Lakukan salah satu hal berikut untuk menentukan peran layanan untuk lingkungan AWS Batch komputasi.

- Gunakan string kosong untuk peran layanan. Ini memungkinkan AWS Batch membuat peran layanan.
- Tentukan peran layanan dalam format berikut: `arn:aws:iam::account_number:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`.

Untuk informasi selengkapnya, lihat [the section called “Nama peran atau ARN salah”](#) di Panduan AWS Batch Pengguna.

Anda dapat menghapus peran tertaut layanan hanya setelah menghapus sumber daya terkait terlebih dahulu. Ini melindungi sumber daya AWS Batch karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran yang terhubung dengan layanan, lihat [Layanan AWS yang Berfungsi dengan IAM](#) dan cari layanan yang memiliki Ya di kolom Peran yang Terhubung dengan Layanan. Pilih Ya dengan sebuah tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Izin peran terkait layanan untuk AWS Batch

AWS Batch menggunakan peran terkait layanan bernama `AWSServiceRoleForBatch`. `AWSServiceRoleForBatchPeran` ini memungkinkan AWS Batch untuk membuat dan mengelola AWS sumber daya atas nama Anda.

Peran `AWSServiceRoleForBatch` terkait layanan mempercayai kepala `batch.amazonaws.com` layanan untuk mengambil peran tersebut.

Kebijakan IAM bernama [BatchServiceRolePolicy](#) memungkinkan AWS Batch untuk menyelesaikan tindakan berikut pada sumber daya tertentu:

- `autoscaling`— Memungkinkan AWS Batch untuk membuat dan mengelola sumber daya Auto Scaling Amazon EC2. AWS Batch membuat dan mengelola grup Auto Scaling Amazon EC2 untuk sebagian besar lingkungan komputasi.
- `ec2`— Memungkinkan AWS Batch untuk mengontrol siklus hidup instans Amazon EC2 serta membuat dan mengelola templat dan tag peluncuran. AWS Batch membuat dan mengelola permintaan Armada Spot EC2 untuk beberapa lingkungan komputasi EC2 Spot.
- `ecs`- Memungkinkan AWS Batch untuk membuat dan mengelola cluster Amazon ECS, definisi tugas dan tugas untuk pelaksanaan pekerjaan.
- `eks`- Memungkinkan AWS Batch untuk mendeskripsikan sumber daya kluster Amazon EKS untuk validasi.
- `iam`- Memungkinkan AWS Batch untuk memvalidasi dan meneruskan peran yang disediakan oleh pemilik ke Amazon EC2, Amazon EC2 Auto Scaling dan Amazon ECS.
- `logs`— Memungkinkan AWS Batch untuk membuat dan mengelola grup log dan aliran log untuk AWS Batch pekerjaan.

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi lebih lanjut, lihat [Izin Peran Tertaut Layanan](#) di Panduan Pengguna IAM.

Membuat peran terkait layanan untuk AWS Batch

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda `CreateComputeEnvironment` berada di AWS Management Console, the AWS CLI, atau AWS API, dan tidak menentukan nilai untuk `serviceRole` parameter, akan AWS Batch membuat peran terkait layanan untuk Anda.

Important

Peran tertaut layanan ini dapat muncul di akun Anda jika Anda menyelesaikan tindakan di layanan lain yang menggunakan fitur yang disupport oleh peran ini. Juga, jika Anda menggunakan AWS Batch layanan sebelum 10 Maret 2021, ketika mulai mendukung peran terkait layanan, maka AWS Batch buat `AWSServiceRoleForBatch` peran tersebut di akun Anda. Untuk mempelajari lebih lanjut, lihat [Peran Baru yang Muncul di Akun IAM Saya](#).

Jika Anda menghapus peran yang terhubung dengan layanan ini, lalu ingin membuatnya lagi, Anda dapat menggunakan proses yang sama untuk membuat ulang peran tersebut di akun Anda. Ketika Anda `CreateComputeEnvironment`, AWS Batch ciptakan peran terkait layanan untuk Anda lagi.

Mengedit peran terkait layanan untuk AWS Batch

Dengan AWS Batch, Anda tidak dapat mengedit peran `AWSServiceRoleForBatch` terkait layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat menyunting penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, silakan lihat [Mengedit peran tertaut layanan](#) dalam Panduan Pengguna IAM.

Untuk mengizinkan entitas IAM mengedit deskripsi peran terkait `AWSServiceRoleForBatch` layanan

Tambahkan pernyataan berikut ke kebijakan izin. Anda dapat menggunakan entitas IAM untuk mengedit deskripsi peran yang terhubung dengan layanan.

```
{
  "Effect": "Allow",
  "Action": [
    "iam:UpdateRoleDescription"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
  "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Menghapus peran terkait layanan untuk AWS Batch

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran yang terhubung dengan layanan, kami menyarankan agar Anda menghapus peran tersebut. Dengan begitu, Anda

tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran terkait layanan sebelum menghapusnya secara manual.

Untuk mengizinkan entitas IAM menghapus peran terkait `AWSServiceRoleForBatch` layanan

Tambahkan pernyataan berikut ke kebijakan izin. Hal ini mengizinkan entitas IAM menghapus peran yang terhubung dengan layanan.

```
{
  "Effect": "Allow",
  "Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
  "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Membersihkan peran terkait layanan

Sebelum Anda dapat menggunakan IAM untuk menghapus peran terkait layanan, Anda harus terlebih dahulu mengonfirmasi bahwa peran tersebut tidak memiliki sesi aktif dan menghapus semua lingkungan AWS Batch komputasi yang menggunakan peran di semua AWS Wilayah dalam satu partisi.

Untuk memeriksa apakah peran yang terhubung dengan layanan memiliki sesi aktif

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Peran dan kemudian `AWSServiceRoleForBatch` nama (bukan kotak centang).
3. Di halaman Summary (Ringkasan), pilih Access Advisor (Penasihat Akses) dan tinjau aktivitas terbaru untuk peran yang terhubung dengan layanan.

Note

Jika Anda tidak tahu AWS Batch apakah menggunakan `AWSServiceRoleForBatch` peran tersebut, Anda dapat mencoba menghapus peran tersebut. Jika layanan yang menggunakan peran tersebut, peran akan gagal dihapus. Anda dapat melihat Wilayah

tempat peran tersebut digunakan. Jika peran tersebut sedang digunakan, Anda harus menunggu hingga sesi ini berakhir sebelum dapat menghapus peran tersebut. Anda tidak dapat mencabut sesi untuk peran tertaut layanan.

Untuk menghapus AWS Batch sumber daya yang digunakan oleh peran AWSServiceRoleForBatch terkait layanan

Anda harus menghapus semua lingkungan AWS Batch komputasi yang menggunakan AWSServiceRoleForBatch peran di semua AWS Wilayah sebelum Anda dapat menghapus AWSServiceRoleForBatch peran.

1. Buka AWS Batch konsol di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi, pilih Wilayah untuk digunakan.
3. Di panel navigasi, pilih Compute environments (Lingkungan komputasi).
4. Pilih lingkungan komputasi.
5. Pilih Disable (Nonaktifkan). Tunggu State (Status) berubah menjadi DISABLED (DINONAKTIFKAN).
6. Pilih lingkungan komputasi.
7. Pilih Hapus. Konfirmasikan bahwa Anda ingin menghapus lingkungan komputasi dengan memilih Delete compute environment (Hapus lingkungan komputasi).
8. Ulangi langkah 1-7 untuk semua lingkungan komputasi yang menggunakan peran yang terhubung dengan layanan di semua Wilayah.


Menghapus peran yang terhubung dengan layanan (Konsol IAM)

Anda dapat menggunakan konsol IAM untuk menghapus sebuah peran terkait layanan.

Untuk menghapus peran terkait layanan (konsol)

1. Masuk ke AWS Management Console dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi konsol IAM, pilih Peran. Kemudian pilih kotak centang di sebelah AWSServiceRoleForBatch, bukan nama atau baris itu sendiri.
3. Pilih Hapus peran.

4. Di kotak dialog konfirmasi, tinjau data layanan yang terakhir diakses, yang menunjukkan kapan masing-masing peran yang dipilih terakhir mengakses file Layanan AWS. Hal ini membantu Anda mengonfirmasi aktif tidaknya peran tersebut saat ini. Jika Anda ingin melanjutkan, pilih Ya, Hapus guna mengirimkan peran tertaut layanan untuk penghapusan.
5. Perhatikan notifikasi konsol IAM untuk memantau progres penghapusan peran tertaut layanan. Karena penghapusan peran terkait layanan IAM bersifat asinkron, setelah Anda mengirimkan peran tersebut untuk dihapus, tugas penghapusan dapat berhasil atau gagal.
 - Jika tugas berhasil, maka peran tersebut dihapus dari daftar dan pemberitahuan keberhasilan muncul di bagian atas halaman.
 - Jika tugas tersebut gagal, Anda dapat memilih Lihat rincian atau Lihat Sumber Daya dari pemberitahuan untuk mempelajari alasan penghapusan gagal. Jika penghapusan gagal karena peran tersebut sedang menggunakan sumber daya layanan, maka pemberitahuan mencakup daftar sumber daya, jika layanan tersebut mengembalikan informasi tersebut. Anda kemudian dapat [membersihkan sumber daya](#) dan mengirimkan penghapusan lagi.

 Note

Anda mungkin harus mengulangi proses ini beberapa kali, tergantung pada informasi yang dikembalikan oleh layanan. Misalnya, peran terkait layanan Anda mungkin menggunakan enam sumber daya dan layanan Anda mungkin akan mengembalikan informasi mengenai lima di antaranya. Jika Anda membersihkan lima sumber daya dan mengirim peran untuk penghapusan lagi, penghapusan tersebut gagal dan layanan melaporkan sumber daya yang tersisa. Layanan mungkin mengembalikan semua sumber daya, beberapa di antaranya, atau tidak melaporkan sumber daya apa pun.

- Jika tugas gagal dan pemberitahuan tidak mencakup daftar sumber daya, maka layanan mungkin tidak mengembalikan informasi tersebut. Untuk mempelajari cara membersihkan sumber daya untuk layanan itu, lihat [Layanan AWS yang bekerja dengan IAM](#). Cari layanan Anda dalam tabel, dan pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut

Menghapus peran yang terhubung dengan layanan di IAM (AWS CLI)

Anda dapat menggunakan perintah IAM dari AWS Command Line Interface untuk menghapus peran terkait layanan.

Untuk menghapus peran yang terhubung dengan layanan (CLI)

1. Karena peran yang terhubung dengan layanan tidak dapat dihapus jika sedang digunakan atau memiliki sumber daya terkait, Anda harus mengirimkan permintaan penghapusan. Permintaan tersebut dapat ditolak jika syarat-syarat ini tidak terpenuhi. Anda harus menangkap `deletion-task-id` dari tanggapan untuk memeriksa status tugas penghapusan. Ketik perintah berikut untuk mengirimkan permintaan penghapusan peran yang terhubung dengan layanan:

```
$ aws iam delete-service-linked-role --role-name AWSServiceRoleForBatch
```

2. Gunakan perintah berikut untuk memeriksa status tugas penghapusan:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

Kemungkinan status tugas penghapusan dapat berupa `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, atau `FAILED`. Jika penghapusan gagal, panggilan akan mengembalikan alasan kegagalan panggilan sehingga Anda dapat memecahkan masalah. Jika penghapusan gagal karena peran tersebut sedang menggunakan sumber daya layanan, maka pemberitahuan mencakup daftar sumber daya, jika layanan tersebut mengembalikan informasi tersebut. Anda kemudian dapat [membersihkan sumber daya](#) dan mengirimkan penghapusan lagi.

Note

Anda mungkin harus mengulangi proses ini beberapa kali, tergantung pada informasi yang dikembalikan oleh layanan. Misalnya, peran terkait layanan Anda mungkin menggunakan enam sumber daya dan layanan Anda mungkin akan mengembalikan informasi mengenai lima di antaranya. Jika Anda membersihkan lima sumber daya dan mengirim peran untuk penghapusan lagi, penghapusan tersebut gagal dan layanan melaporkan sumber daya yang tersisa. Layanan mungkin mengembalikan semua sumber daya atau beberapa di antaranya. Layanan mungkin juga tidak melaporkan sumber daya apa pun. Untuk mempelajari cara membersihkan sumber daya untuk layanan yang tidak melaporkan sumber daya apa pun, lihat [Layanan AWS yang bekerja dengan IAM](#). Cari layanan Anda dalam tabel, dan pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut

Menghapus peran yang terhubung dengan layanan di IAM (API AWS)

Anda dapat menggunakan IAM API untuk menghapus peran terkait layanan.

Untuk menghapus peran terkait layanan (API)

1. Untuk mengirimkan permintaan penghapusan untuk roll terkait layanan, hubungi. [DeleteServiceLinkedRole](#) Dalam permintaan, tentukan nama `AWSServiceRoleForBatch` peran.

Karena peran terkait layanan tidak dapat dihapus jika sedang digunakan atau memiliki sumber daya terkait, Anda harus mengirimkan permintaan penghapusan. Permintaan tersebut dapat ditolak jika syarat ini tidak terpenuhi. Anda harus menangkap `DeletionTaskId` dari tanggapan untuk memeriksa status tugas penghapusan.

2. Untuk memeriksa status penghapusan, hubungi. [GetServiceLinkedRoleDeletionStatus](#) Di permintaan tersebut, tentukan `DeletionTaskId`.

Kemungkinan status tugas penghapusan dapat berupa `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, atau `FAILED`. Jika penghapusan gagal, panggilan akan mengembalikan alasan kegagalan panggilan sehingga Anda dapat memecahkan masalah. Jika penghapusan gagal karena peran tersebut sedang menggunakan sumber daya layanan, maka pemberitahuan mencakup daftar sumber daya, jika layanan tersebut mengembalikan informasi tersebut. Anda kemudian dapat [membersihkan sumber daya](#) dan mengirimkan penghapusan lagi.

Note

Anda mungkin harus mengulangi proses ini beberapa kali, tergantung pada informasi yang dikembalikan oleh layanan. Misalnya, peran terkait layanan Anda mungkin menggunakan enam sumber daya dan layanan Anda mungkin akan mengembalikan informasi mengenai lima di antaranya. Jika Anda membersihkan lima sumber daya dan mengirim peran untuk penghapusan lagi, penghapusan tersebut gagal dan layanan melaporkan sumber daya yang tersisa. Layanan mungkin mengembalikan semua sumber daya, beberapa di antaranya, atau tidak melaporkan sumber daya apa pun. Untuk mempelajari cara membersihkan sumber daya untuk layanan yang tidak melaporkan sumber daya apa pun, lihat [Layanan AWS yang berfungsi dengan IAM](#). Cari layanan Anda dalam tabel, dan pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut

Wilayah yang Didukung untuk AWS Batch peran terkait layanan

AWS Batch mendukung penggunaan peran terkait layanan di semua Wilayah tempat layanan tersedia. Untuk informasi lebih lanjut, lihat [Titik akhir AWS Batch](#).

AWS kebijakan terkelola untuk AWS Batch

Anda dapat menggunakan kebijakan AWS terkelola untuk pengelolaan akses identitas yang lebih sederhana untuk tim dan sumber daya yang disediakan. AWS AWS Kebijakan terkelola mencakup berbagai kasus penggunaan umum, tersedia secara default di AWS akun Anda, dan dikelola serta diperbarui atas nama Anda. Anda tidak dapat mengubah izin dalam kebijakan AWS terkelola. Jika Anda membutuhkan fleksibilitas yang lebih besar, Anda dapat memilih untuk membuat kebijakan yang dikelola pelanggan IAM. Dengan cara ini, Anda dapat memberikan sumber daya yang disediakan tim Anda hanya dengan izin persis yang mereka butuhkan.

Untuk informasi selengkapnya tentang kebijakan AWS [AWS terkelola](#), lihat [kebijakan terkelola](#) di Panduan Pengguna IAM.

AWS Layanan memelihara dan memperbarui kebijakan AWS terkelola atas nama Anda. Secara berkala, AWS layanan menambahkan izin tambahan ke kebijakan AWS terkelola. AWS Kebijakan terkelola kemungkinan besar diperbarui saat peluncuran atau operasi fitur baru tersedia. Pembaruan ini secara otomatis memengaruhi semua identitas (pengguna, grup, dan peran) tempat kebijakan dilampirkan. Namun, mereka tidak menghapus izin atau merusak izin yang ada.

Selain itu, AWS mendukung kebijakan terkelola untuk fungsi pekerjaan yang mencakup beberapa layanan. Misalnya, kebijakan `ReadOnlyAccess` AWS terkelola menyediakan akses hanya-baca ke semua AWS layanan dan sumber daya. Saat layanan meluncurkan fitur baru, AWS tambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk melihat daftar dan deskripsi dari kebijakan fungsi tugas, lihat [kebijakan yang dikelola AWS untuk fungsi tugas](#) di Panduan Pengguna IAM.

AWS kebijakan terkelola: `BatchServiceRolePolicy`

Kebijakan IAM `BatchServiceRolePolicy` terkelola digunakan oleh peran [`AWSBatchServiceRoleForBatch`](#) terkait layanan. Ini memungkinkan AWS Batch untuk melakukan tindakan atas nama Anda. Anda tidak dapat melampirkan kebijakan ini ke entitas IAM Anda. Untuk informasi selengkapnya, lihat [Menggunakan peran terkait layanan untuk AWS Batch](#).

Kebijakan ini memungkinkan AWS Batch untuk menyelesaikan tindakan berikut pada sumber daya tertentu:

- `autoscaling`— Memungkinkan AWS Batch untuk membuat dan mengelola sumber daya Auto Scaling Amazon EC2. AWS Batch membuat dan mengelola grup Auto Scaling Amazon EC2 untuk sebagian besar lingkungan komputasi.
- `ec2`— Memungkinkan AWS Batch untuk mengontrol siklus hidup instans Amazon EC2 serta membuat dan mengelola templat dan tag peluncuran. AWS Batch membuat dan mengelola permintaan Armada Spot EC2 untuk beberapa lingkungan komputasi EC2 Spot.
- `ecs`- Memungkinkan AWS Batch untuk membuat dan mengelola cluster Amazon ECS, definisi tugas dan tugas untuk pelaksanaan pekerjaan.
- `eks`- Memungkinkan AWS Batch untuk mendeskripsikan sumber daya kluster Amazon EKS untuk validasi.
- `iam`- Memungkinkan AWS Batch untuk memvalidasi dan meneruskan peran yang disediakan oleh pemilik ke Amazon EC2, Amazon EC2 Auto Scaling dan Amazon ECS.
- `logs`— Memungkinkan AWS Batch untuk membuat dan mengelola grup log dan aliran log untuk AWS Batch pekerjaan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSBatchPolicyStatement1",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstanceAttribute",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeImages",
        "ec2:DescribeImageAttribute",
        "ec2:DescribeSpotInstanceRequests",
        "ec2:DescribeSpotFleetInstances",
        "ec2:DescribeSpotFleetRequests",
        "ec2:DescribeSpotPriceHistory",
```

```

        "ec2:DescribeSpotFleetRequestHistory",
        "ec2:DescribeVpcClassicLink",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:RequestSpotFleet",
        "autoscaling:DescribeAccountLimits",
        "autoscaling:DescribeAutoScalingGroups",
        "autoscaling:DescribeLaunchConfigurations",
        "autoscaling:DescribeAutoScalingInstances",
        "autoscaling:DescribeScalingActivities",
        "eks:DescribeCluster",
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
        "ecs:DescribeTaskDefinition",
        "ecs:DescribeTasks",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListTaskDefinitionFamilies",
        "ecs:ListTaskDefinitions",
        "ecs:ListTasks",
        "ecs:DeregisterTaskDefinition",
        "ecs:TagResource",
        "ecs:ListAccountSettings",
        "logs:DescribeLogGroups",
        "iam:GetInstanceProfile",
        "iam:GetRole"
    ],
    "Resource": "*"
},
{
    "Sid": "AWSBatchPolicyStatement2",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/batch/job*"
},
{
    "Sid": "AWSBatchPolicyStatement3",
    "Effect": "Allow",
    "Action": [
        "logs:PutLogEvents"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/batch/job*:log-stream:*"
}

```

```
    },
    {
      "Sid": "AWSBatchPolicyStatement4",
      "Effect": "Allow",
      "Action": [
        "autoscaling:CreateOrUpdateTags"
      ],
      "Resource": "*",
      "Condition": {
        "Null": {
          "aws:RequestTag/AWSBatchServiceTag": "false"
        }
      }
    },
    {
      "Sid": "AWSBatchPolicyStatement5",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": [
            "ec2.amazonaws.com",
            "ec2.amazonaws.com.cn",
            "ecs-tasks.amazonaws.com"
          ]
        }
      }
    },
    {
      "Sid": "AWSBatchPolicyStatement6",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "spot.amazonaws.com",
            "spotfleet.amazonaws.com",
            "autoscaling.amazonaws.com",
            "ecs.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```



```

    }
  }
},
{
  "Sid": "AWSBatchPolicyStatement7",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateLaunchTemplate"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "aws:RequestTag/AWSBatchServiceTag": "false"
    }
  }
},
{
  "Sid": "AWSBatchPolicyStatement8",
  "Effect": "Allow",
  "Action": [
    "ec2:TerminateInstances",
    "ec2:CancelSpotFleetRequests",
    "ec2:ModifySpotFleetRequest",
    "ec2>DeleteLaunchTemplate"
  ],
  "Resource": "*",
  "Condition": {
    "Null": {
      "aws:ResourceTag/AWSBatchServiceTag": "false"
    }
  }
},
{
  "Sid": "AWSBatchPolicyStatement9",
  "Effect": "Allow",
  "Action": [
    "autoscaling:CreateLaunchConfiguration",
    "autoscaling>DeleteLaunchConfiguration"
  ],
  "Resource":
"arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/AWSBatch*"
},
{
  "Sid": "AWSBatchPolicyStatement10",

```

```

    "Effect": "Allow",
    "Action": [
      "autoscaling:CreateAutoScalingGroup",
      "autoscaling:UpdateAutoScalingGroup",
      "autoscaling:SetDesiredCapacity",
      "autoscaling>DeleteAutoScalingGroup",
      "autoscaling:SuspendProcesses",
      "autoscaling:PutNotificationConfiguration",
      "autoscaling:TerminateInstanceInAutoScalingGroup"
    ],
    "Resource":
      "arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/AWSBatch*"
  },
  {
    "Sid": "AWSBatchPolicyStatement11",
    "Effect": "Allow",
    "Action": [
      "ecs>DeleteCluster",
      "ecs:DeregisterContainerInstance",
      "ecs:RunTask",
      "ecs:StartTask",
      "ecs:StopTask"
    ],
    "Resource": "arn:aws:ecs:*:*:cluster/AWSBatch*"
  },
  {
    "Sid": "AWSBatchPolicyStatement12",
    "Effect": "Allow",
    "Action": [
      "ecs:RunTask",
      "ecs:StartTask",
      "ecs:StopTask"
    ],
    "Resource": "arn:aws:ecs:*:*:task-definition/*"
  },
  {
    "Sid": "AWSBatchPolicyStatement13",
    "Effect": "Allow",
    "Action": [
      "ecs:StopTask"
    ],
    "Resource": "arn:aws:ecs:*:*:task/*/*"
  },
  {

```

```

    "Sid": "AWSBatchPolicyStatement14",
    "Effect": "Allow",
    "Action": [
        "ecs:CreateCluster",
        "ecs:RegisterTaskDefinition"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "aws:RequestTag/AWSBatchServiceTag": "false"
        }
    }
},
{
    "Sid": "AWSBatchPolicyStatement15",
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": [
        "arn:aws:ec2:*:*:image/*",
        "arn:aws:ec2:*:*:snapshot/*",
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:network-interface/*",
        "arn:aws:ec2:*:*:security-group/*",
        "arn:aws:ec2:*:*:volume/*",
        "arn:aws:ec2:*:*:key-pair/*",
        "arn:aws:ec2:*:*:launch-template/*",
        "arn:aws:ec2:*:*:placement-group/*",
        "arn:aws:ec2:*:*:capacity-reservation/*",
        "arn:aws:ec2:*:*:elastic-gpu/*",
        "arn:aws:elastic-inference:*:*:elastic-inference-accelerator/*",
        "arn:aws:resource-groups:*:*:group*"
    ]
},
{
    "Sid": "AWSBatchPolicyStatement16",
    "Effect": "Allow",
    "Action": "ec2:RunInstances",
    "Resource": "arn:aws:ec2:*:*:instance/*",
    "Condition": {
        "Null": {
            "aws:RequestTag/AWSBatchServiceTag": "false"
        }
    }
}
},

```

```

    {
      "Sid": "AWSBatchPolicyStatement17",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": [
            "RunInstances",
            "CreateLaunchTemplate",
            "RequestSpotFleet"
          ]
        }
      }
    }
  ]
}

```

AWS kebijakan terkelola: AWSBatchServiceRolekebijakan

Kebijakan izin peran bernama `AWSBatchServiceRole` memungkinkan AWS Batch untuk menyelesaikan tindakan berikut pada sumber daya tertentu:

Kebijakan IAM `AWSBatchServiceRole` terkelola sering digunakan oleh peran bernama `AWSBatchServiceRole` dan menyertakan izin berikut. Mengikuti saran keamanan standar untuk memberikan hak istimewa paling sedikit, kebijakan yang `AWSBatchServiceRole` dikelola dapat digunakan sebagai panduan. Jika salah satu izin yang diberikan dalam kebijakan terkelola tidak diperlukan untuk kasus penggunaan Anda, buat kebijakan kustom dan hanya tambahkan izin yang Anda perlukan. Kebijakan dan peran AWS Batch terkelola ini dapat digunakan dengan sebagian besar jenis lingkungan komputasi, tetapi penggunaan peran terkait layanan lebih disukai untuk pengalaman terkelola dengan cakupan yang `less-error-prone` lebih baik dan lebih baik.

- `autoscaling`— Memungkinkan AWS Batch untuk membuat dan mengelola sumber daya Auto Scaling Amazon EC2. AWS Batch membuat dan mengelola grup Auto Scaling Amazon EC2 untuk sebagian besar lingkungan komputasi.

- **ec2**— Memungkinkan AWS Batch untuk mengelola siklus hidup instans Amazon EC2 serta membuat dan mengelola templat dan tag peluncuran. AWS Batch membuat dan mengelola permintaan Armada Spot EC2 untuk beberapa lingkungan komputasi EC2 Spot.
- **ecs**- Memungkinkan AWS Batch untuk membuat dan mengelola cluster Amazon ECS, definisi tugas dan tugas untuk pelaksanaan pekerjaan.
- **iam**- Memungkinkan AWS Batch untuk memvalidasi dan meneruskan peran yang disediakan oleh pemilik ke Amazon EC2, Amazon EC2 Auto Scaling dan Amazon ECS.
- **logs**— Memungkinkan AWS Batch untuk membuat dan mengelola grup log dan aliran log untuk AWS Batch pekerjaan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AWSBatchPolicyStatement1",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeAccountAttributes",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstanceAttribute",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeImages",
        "ec2:DescribeImageAttribute",
        "ec2:DescribeSpotInstanceRequests",
        "ec2:DescribeSpotFleetInstances",
        "ec2:DescribeSpotFleetRequests",
        "ec2:DescribeSpotPriceHistory",
        "ec2:DescribeSpotFleetRequestHistory",
        "ec2:DescribeVpcClassicLink",
        "ec2:DescribeLaunchTemplateVersions",
        "ec2:CreateLaunchTemplate",
        "ec2>DeleteLaunchTemplate",
        "ec2:RequestSpotFleet",
        "ec2:CancelSpotFleetRequests",
        "ec2:ModifySpotFleetRequest",
        "ec2:TerminateInstances",
        "ec2:RunInstances",
```

```
    "autoscaling:DescribeAccountLimits",
    "autoscaling:DescribeAutoScalingGroups",
    "autoscaling:DescribeLaunchConfigurations",
    "autoscaling:DescribeAutoScalingInstances",
    "autoscaling:DescribeScalingActivities",
    "autoscaling:CreateLaunchConfiguration",
    "autoscaling:CreateAutoScalingGroup",
    "autoscaling:UpdateAutoScalingGroup",
    "autoscaling:SetDesiredCapacity",
    "autoscaling>DeleteLaunchConfiguration",
    "autoscaling>DeleteAutoScalingGroup",
    "autoscaling:CreateOrUpdateTags",
    "autoscaling:SuspendProcesses",
    "autoscaling:PutNotificationConfiguration",
    "autoscaling:TerminateInstanceInAutoScalingGroup",
    "ecs:DescribeClusters",
    "ecs:DescribeContainerInstances",
    "ecs:DescribeTaskDefinition",
    "ecs:DescribeTasks",
    "ecs:ListAccountSettings",
    "ecs:ListClusters",
    "ecs:ListContainerInstances",
    "ecs:ListTaskDefinitionFamilies",
    "ecs:ListTaskDefinitions",
    "ecs:ListTasks",
    "ecs:CreateCluster",
    "ecs>DeleteCluster",
    "ecs:RegisterTaskDefinition",
    "ecs:DeregisterTaskDefinition",
    "ecs:RunTask",
    "ecs:StartTask",
    "ecs:StopTask",
    "ecs:UpdateContainerAgent",
    "ecs:DeregisterContainerInstance",
    "logs:CreateLogGroup",
    "logs:CreateLogStream",
    "logs:PutLogEvents",
    "logs:DescribeLogGroups",
    "iam:GetInstanceProfile",
    "iam:GetRole"
  ],
  "Resource": "*"
},
{
```

```

    "Sid": "AWSBatchPolicyStatement2",
    "Effect": "Allow",
    "Action": "ecs:TagResource",
    "Resource": [
      "arn:aws:ecs:*:*:task/*_Batch_*"
    ]
  },
  {
    "Sid": "AWSBatchPolicyStatement3",
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": [
      "*"
    ],
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": [
          "ec2.amazonaws.com",
          "ec2.amazonaws.com.cn",
          "ecs-tasks.amazonaws.com"
        ]
      }
    }
  },
  {
    "Sid": "AWSBatchPolicyStatement4",
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "iam:AWSServiceName": [
          "spot.amazonaws.com",
          "spotfleet.amazonaws.com",
          "autoscaling.amazonaws.com",
          "ecs.amazonaws.com"
        ]
      }
    }
  },
  {
    "Sid": "AWSBatchPolicyStatement5",
    "Effect": "Allow",
    "Action": [

```

```

        "ec2:CreateTags"
    ],
    "Resource": [
        "*"
    ],
    "Condition": {
        "StringEquals": {
            "ec2:CreateAction": "RunInstances"
        }
    }
}
]
}

```

AWS kebijakan terkelola: AWSBatchFullAccess

AWSBatchFullAccessKebijakan memberikan AWS Batch tindakan akses penuh ke AWS Batch sumber daya. Ini juga memberikan deskripsi dan daftar akses tindakan untuk Amazon EC2, Amazon ECS, Amazon EKS, dan layanan CloudWatch IAM. Ini agar identitas IAM, baik pengguna atau peran, dapat melihat sumber daya AWS Batch terkelola yang dibuat atas nama mereka. Terakhir, kebijakan ini juga memungkinkan IAM role yang dipilih untuk diteruskan ke layanan tersebut.

Anda dapat melampirkan AWSBatchFullAccesske entitas IAM Anda. AWS Batch juga melampirkan kebijakan ini ke peran layanan yang memungkinkan AWS Batch untuk melakukan tindakan atas nama Anda.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:*",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeImages",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ecs:DescribeClusters",

```



```

    "ecs:Describe*",
    "ecs:List*",
    "eks:DescribeCluster",
    "eks:ListClusters",
    "logs:Describe*",
    "logs:Get*",
    "logs:TestMetricFilter",
    "logs:FilterLogEvents",
    "iam:ListInstanceProfiles",
    "iam:ListRoles"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [
    "arn:aws:iam::*:role/AWSBatchServiceRole",
    "arn:aws:iam::*:role/service-role/AWSBatchServiceRole",
    "arn:aws:iam::*:role/ecsInstanceRole",
    "arn:aws:iam::*:instance-profile/ecsInstanceRole",
    "arn:aws:iam::*:role/iaws-ec2-spot-fleet-role",
    "arn:aws:iam::*:role/aws-ec2-spot-fleet-role",
    "arn:aws:iam::*:role/AWSBatchJobRole*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateServiceLinkedRole"
  ],
  "Resource": "arn:aws:iam::*:role/*Batch*",
  "Condition": {
    "StringEquals": {
      "iam:AWSServiceName": "batch.amazonaws.com"
    }
  }
}
]
}

```

AWS Batch pembaruan kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola AWS Batch sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman Riwayat AWS Batch dokumen.

Perubahan	Deskripsi	Tanggal
BatchServiceRolePolicy kebijakan diperbarui	Diperbarui untuk menambahkan dukungan untuk menjelaskan riwayat dan Amazon EC2 Auto Scaling aktivitas permintaan Armada Spot.	5 Desember 2023
AWSBatchServiceRole kebijakan ditambahkan	Diperbarui untuk menambahkan ID pernyataan, memberikan AWS Batch izin ke <code>ec2:DescribeSpotFleetRequestHistory</code> dan <code>autoscaling:DescribeScalingActivities</code> .	5 Desember 2023
BatchServiceRolePolicy kebijakan diperbarui	Diperbarui untuk menambahkan dukungan untuk mendeskripsikan kluster Amazon EKS.	20 Oktober 2022
AWSBatchFullAccess kebijakan diperbarui	Diperbarui untuk menambahkan dukungan untuk daftar dan menjelaskan kluster Amazon EKS.	20 Oktober 2022
BatchServiceRolePolicy kebijakan diperbarui	Diperbarui untuk menambahkan dukungan untuk grup Reservasi Kapasitas Amazon EC2 yang dikelola oleh AWS Resource Groups Untuk informasi selengkapnya, lihat Bekerja dengan grup Reservasi Kapasitas di Panduan Pengguna Amazon EC2 untuk Instans Linux.	Mei 18, 2022

Perubahan	Deskripsi	Tanggal
BatchServiceRolePolicy dan AWSBatchServiceRolePolicy kebijakan diperbarui	Diperbarui untuk menambahkan dukungan untuk menjelaskan status instans AWS Batch terkelola di Amazon EC2 sehingga instans yang tidak sehat diganti.	Desember 6, 2021
BatchServiceRolePolicy kebijakan diperbarui	Diperbarui untuk menambahkan dukungan untuk grup penempatan, reservasi kapasitas, GPU elastis, dan sumber daya Elastic Inference di Amazon EC2.	26 Maret 2021
BatchServiceRolePolicy kebijakan ditambahkan	Dengan kebijakan BatchServiceRolePolicy terkelola untuk peran AWSServiceRoleForBatch terkait layanan, Anda dapat menggunakan peran terkait layanan yang dikelola oleh AWS Batch. Dengan kebijakan ini, Anda tidak perlu mempertahankan peran Anda sendiri untuk digunakan di lingkungan komputasi Anda.	10 Maret 2021
AWSBatchFullAccess - tambahkan izin untuk menambahkan peran terkait layanan	Tambahkan izin IAM untuk memungkinkan peran AWSServiceRoleForBatch terkait layanan ditambahkan ke akun.	10 Maret 2021
AWS Batch mulai melacak perubahan	AWS Batch mulai melacak perubahan untuk kebijakan yang AWS dikelola.	10 Maret 2021

Akses AWS Batch menggunakan titik akhir antarmuka

Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi pribadi antara VPC Anda dan AWS Batch. Anda dapat mengakses AWS Batch seolah-olah itu ada di VPC Anda, tanpa

menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk mengakses. AWS Batch

Anda membuat koneksi pribadi ini dengan membuat titik akhir antarmuka, yang didukung oleh AWS PrivateLink. Kami membuat antarmuka jaringan endpoint di setiap subnet yang Anda aktifkan untuk titik akhir antarmuka. Ini adalah antarmuka jaringan yang dikelola pemohon yang berfungsi sebagai titik masuk untuk lalu lintas yang ditakdirkan. AWS Batch

Untuk informasi selengkapnya, lihat [Titik akhir VPC Antarmuka di Panduan.AWS PrivateLink](#)

Pertimbangan untuk AWS Batch

Sebelum Anda menyiapkan titik akhir antarmuka AWS Batch, tinjau [properti dan batasan titik akhir Antarmuka](#) dalam Panduan.AWS PrivateLink

AWS Batch mendukung panggilan ke semua tindakan API-nya melalui titik akhir antarmuka.

Sebelum Anda mengatur titik akhir VPC antarmuka untuk AWS Batch, perhatikan pertimbangan berikut:

- Pekerjaan yang menggunakan jenis peluncuran sumber daya Fargate tidak memerlukan titik akhir VPC antarmuka untuk Amazon ECS, tetapi Anda mungkin memerlukan titik akhir VPC antarmuka untuk, Amazon ECR, Secrets Manager, atau AWS Batch Amazon Logs yang dijelaskan dalam poin berikut. CloudWatch
 - Untuk menjalankan pekerjaan, Anda harus membuat titik akhir VPC antarmuka untuk Amazon ECS. Untuk informasi selengkapnya, lihat [Titik Akhir VPC Antarmuka \(AWS PrivateLink\)](#) di Panduan Pengembang Layanan Amazon Elastic Container.
 - Untuk memungkinkan pekerjaan Anda menarik gambar pribadi dari Amazon ECR, Anda harus membuat titik akhir VPC antarmuka untuk Amazon ECR. Untuk informasi selengkapnya, lihat [Antarmuka VPC Endpoints \(AWS PrivateLink\)](#) di Panduan Pengguna Amazon Elastic Container Registry.
 - Untuk memungkinkan pekerjaan Anda menarik data sensitif dari Secrets Manager, Anda harus membuat titik akhir VPC antarmuka untuk Secrets Manager. Untuk informasi selengkapnya, lihat [Menggunakan Secrets Manager dengan VPC Endpoint](#) dalam Panduan Pengguna AWS Secrets Manager .
 - Jika VPC Anda tidak memiliki gateway internet dan pekerjaan Anda menggunakan driver `awslogs` log untuk mengirim informasi log ke Log, Anda harus membuat antarmuka VPC endpoint untuk CloudWatch Log. CloudWatch Untuk informasi selengkapnya, lihat [Menggunakan](#)

[CloudWatch Log dengan Titik Akhir VPC Antarmuka di Panduan Pengguna Amazon CloudWatch Logs.](#)

- Pekerjaan yang menggunakan sumber daya EC2 mengharuskan instans penampung yang diluncurkan untuk menjalankan versi 1.25.1 atau yang lebih baru dari agen penampung Amazon ECS. Untuk informasi selengkapnya, lihat [versi agen penampung Amazon ECS Linux](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.
- VPC endpoint saat ini tidak mendukung permintaan lintas Wilayah. Pastikan bahwa Anda membuat titik akhir Anda di Wilayah yang sama tempat Anda berencana untuk mengeluarkan panggilan API ke AWS Batch.
- Titik akhir VPC hanya mendukung DNS yang disediakan Amazon melalui Amazon Route 53. Jika Anda ingin menggunakan DNS Anda sendiri, Anda dapat menggunakan penerusan DNS bersyarat. Untuk informasi selengkapnya, lihat [Pengaturan DHCP](#) dalam Panduan Pengguna Amazon VPC.
- Grup keamanan yang dilampirkan ke VPC endpoint harus mengizinkan koneksi masuk pada port 443 dari subnet privat VPC.
- AWS Batch tidak mendukung titik akhir antarmuka VPC sebagai berikut: Wilayah AWS
 - Asia Pacific (Osaka) (ap-northeast-3)
 - Asia Pasifik (Jakarta) (ap-southeast-3)

Buat titik akhir antarmuka untuk AWS Batch

Anda dapat membuat titik akhir antarmuka untuk AWS Batch menggunakan konsol VPC Amazon atau () AWS Command Line Interface .AWS CLI Untuk informasi selengkapnya, lihat [Membuat titik akhir antarmuka](#) di AWS PrivateLink Panduan.

Buat titik akhir antarmuka untuk AWS Batch menggunakan nama layanan berikut:

```
com.amazonaws.region.batch
```

Sebagai contoh:

```
com.amazonaws.us-east-2.batch
```

Di aws-cn partisi, formatnya berbeda:

```
cn.com.amazonaws.region.batch
```

Sebagai contoh:

```
cn.com.amazonaws.cn-northwest-1.batch
```

Jika Anda mengaktifkan DNS pribadi untuk titik akhir antarmuka, Anda dapat membuat permintaan API untuk AWS Batch menggunakan nama DNS Regional default. Misalnya, `batch.us-east-1.amazonaws.com`.

Untuk informasi selengkapnya, lihat [Mengakses layanan melalui titik akhir antarmuka](#) di AWS PrivateLink Panduan.

Buat kebijakan titik akhir untuk titik akhir antarmuka Anda

Kebijakan endpoint adalah sumber daya IAM yang dapat Anda lampirkan ke titik akhir antarmuka. Kebijakan endpoint default memungkinkan akses penuh AWS Batch melalui titik akhir antarmuka. Untuk mengontrol akses yang diizinkan AWS Batch dari VPC Anda, lampirkan kebijakan titik akhir kustom ke titik akhir antarmuka.

kebijakan titik akhir mencantumkan informasi berikut:

- Prinsipal yang dapat melakukan tindakan (Akun AWS, pengguna, dan peran IAM).
- Tindakan yang dapat dilakukan.
- Sumber daya untuk melakukan tindakan.

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

Contoh: Kebijakan titik akhir VPC untuk tindakan AWS Batch

Berikut ini adalah contoh kebijakan endpoint kustom. Saat Anda melampirkan kebijakan ini ke titik akhir antarmuka Anda, kebijakan ini akan memberikan akses ke AWS Batch tindakan yang tercantum untuk semua prinsip di semua sumber daya.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
```

```
        "batch:SubmitJob",
        "batch:ListJobs",
        "batch:DescribeJobs"
    ],
    "Resource": "*"
}
]
```

Validasi Kepatuhan untuk AWS Batch

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

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

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

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

Note

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

- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Panduan Kepatuhan Pelanggan](#) - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS

dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).

- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#)— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk 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](#)Ini Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Keamanan Infrastruktur di AWS Batch

Sebagai layanan terkelola, AWS Batch dilindungi oleh keamanan jaringan AWS 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 AWS Batch 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 dapat memanggil operasi API ini dari lokasi jaringan mana pun, AWS Batch tetapi mendukung kebijakan akses berbasis sumber daya, yang dapat mencakup pembatasan berdasarkan alamat IP sumber. Anda juga dapat menggunakan AWS Batch kebijakan untuk mengontrol akses dari titik akhir Amazon Virtual Private Cloud (Amazon VPC) tertentu atau VPC tertentu. Secara efektif, ini mengisolasi akses jaringan ke AWS Batch sumber daya tertentu hanya dari VPC tertentu dalam AWS jaringan.

Menandai sumber daya AWS Batch Anda

Untuk membantu Anda mengelola sumber daya AWS Batch, Anda dapat menetapkan metadata Anda sendiri ke setiap sumber daya dalam bentuk tanda. Topik ini menjelaskan tentang tanda dan menunjukkan kepada Anda cara membuatnya.

Daftar Isi

- [Dasar-dasar tanda](#)
- [Penandaan sumber daya Anda](#)
- [Batasan tanda](#)
- [Bekerja dengan tanda menggunakan konsol](#)
- [Cara menggunakan tanda dengan menggunakan CLI atau API](#)

Dasar-dasar tanda

Tanda merupakan label yang Anda tetapkan ke sumber daya AWS. Setiap tanda terdiri atas sebuah kunci dan sebuah nilai opsional, yang keduanya Anda tentukan.

Tanda memungkinkan Anda untuk mengategorikan sumber daya AWS Anda dengan, misalnya, berdasarkan tujuan, pemilik, atau lingkungan. Saat Anda memiliki banyak sumber daya dengan jenis yang sama, Anda dapat dengan segera mengidentifikasi sumber daya yang spesifik berdasarkan tanda yang telah Anda tetapkan pada sumber daya. Misalnya, Anda dapat menentukan satu set tanda untuk layanan AWS Batch untuk membantu Anda melacak setiap pemilik dan tingkat tumpukan layanan. Kami menyarankan agar Anda merancang serangkaian kunci tanda yang konsisten untuk setiap jenis sumber daya.

Selain itu, tanda tidak dapat menetapkan secara otomatis ke sumber daya Anda. Setelah Anda menambahkan sebuah tanda, Anda dapat mengedit kunci serta nilai tanda atau menghilangkan tanda dari sumber daya kapanpun yang Anda mau. Jika Anda menghapus sebuah sumber daya, tanda apapun untuk sumber daya tersebut juga dihapus.

Tanda tidak memiliki makna semantik pada AWS Batch dan diterjemahkan sebagai serangkaian karakter saja. Anda dapat mengatur nilai tanda menjadi sebuah string kosong, tetapi Anda tidak dapat mengatur nilai tanda menjadi nol. Jika Anda menambahkan tanda yang memiliki kunci yang sama dengan tanda yang ada pada sumber daya tersebut, nilai yang baru akan menimpa nilai yang lama.

Anda dapat bekerja dengan tanda menggunakan AWS Management Console, AWS CLI, dan API AWS Batch.

Jika Anda menggunakan AWS Identity and Access Management (IAM), Anda dapat mengendalikan pengguna yang mana di akun AWS Anda yang memiliki izin untuk membuat, mengedit, atau menghapus tanda.

Penandaan sumber daya Anda

Anda dapat menandai lingkungan AWS Batch komputasi baru atau yang sudah ada, pekerjaan, definisi pekerjaan, antrian pekerjaan, dan kebijakan penjadwalan.

Jika Anda menggunakan konsol AWS Batch, Anda dapat menerapkan tanda ke sumber daya baru ketika itu dibuat atau ke sumber daya yang sudah ada dengan menggunakan tab Tags (Tanda) di panel navigasi kapan saja.

Jika Anda menggunakan API AWS Batch, AWS CLI, atau AWS SDK, Anda dapat menerapkan tanda ke sumber daya baru dengan menggunakan parameter `tags` di pada tindakan API yang relevan atau ke sumber daya yang ada dengan menggunakan tindakan API `TagResource`. Untuk informasi lebih lanjut, lihat [TagResource](#).

Selain itu, beberapa tindakan pembuatan sumber daya memungkinkan Anda menentukan tanda untuk sumber daya saat sumber daya diciptakan. Jika tanda tidak dapat diterapkan selama pembuatan sumber daya, proses pembuatan sumber daya akan gagal. Hal ini memastikan bahwa sumber daya yang ingin Anda tandai pada saat pembuatan dapat dibuat dengan tanda yang ditentukan atau justru tidak dibuat sama sekali. Jika Anda menandai sumber daya pada saat pembuatan, Anda tidak perlu menjalankan skrip penandaan khusus setelah pembuatan sumber daya.

Tabel berikut menjelaskan sumber daya AWS Batch yang dapat ditandai, dan sumber daya yang dapat ditandai saat dibuat.

Dukungan penandaan untuk sumber daya AWS Batch

Sumber daya	Mendukung tanda	Penyebaran tanda Support	Mendukung penandaan saat pembuatan (AWS Batch API, AWS CLI, AWS SDK)
Lingkungan komputasi AWS Batch	Ya	Tidak. Tanda lingkungan komputasi tidak dipropagasi ke sumber daya lainnya. Tag untuk sumber daya ditentukan dalam anggota tag objek <code>ComputeResources</code> yang diteruskan dalam operasi API. CreateComputeEnvironment	Ya
Tugas AWS Batch	Ya	Ya	Ya
Ketentuan tugas AWS Batch	Ya	Tidak	Ya
Antrean tugas AWS Batch	Ya	Tidak	Ya
AWS Batch kebijakan penjadwalan	Ya	Tidak	Ya

Batasan tanda

Batasan dasar berikut berlaku untuk tanda:

- Jumlah maksimum tanda per sumber daya – 50

- Untuk setiap sumber daya, setiap kunci tanda harus unik, dan setiap kunci tanda hanya dapat memuat satu nilai.
- Panjang kunci maksimum – 128 karakter Unicode dalam UTF-8
- Panjang nilai maksimum – 256 karakter Unicode dalam UTF-8
- Jika skema penandaan Anda digunakan di beberapa layanan dan sumber daya AWS, ingatlah bahwa layanan lain mungkin memiliki pembatasan pada karakter yang diizinkan. Karakter yang secara umum diperbolehkan adalah huruf, angka, spasi yang dapat diwakili dalam UTF-8, serta karakter berikut: + - = . _ : / @.
- Kunci dan nilai tanda sensitif huruf besar dan kecil.
- Jangan gunakan `aws :`, `AWS :`, atau kombinasi huruf besar atau huruf kecil dari itu semua sebagai prefiks untuk kunci atau nilai karena itu semua disimpan untuk penggunaan AWS. Anda tidak dapat menyunting atau menghapus kunci atau nilai tanda dengan prefiks ini. Tag dengan awalan ini tidak dihitung terhadap tags-per-resource batas Anda.

Bekerja dengan tanda menggunakan konsol

Menggunakan perintah konsol AWS Batch, Anda dapat mengelola tanda yang terkait dengan lingkungan komputasi, tugas, ketentuan tugas, dan antrean tugas baru atau yang sudah ada.

Menambahkan tanda pada pembuatan sumber daya individu

Anda dapat menambahkan tag untuk AWS Batch menghitung lingkungan, pekerjaan, definisi pekerjaan, antrian pekerjaan, dan kebijakan penjadwalan saat Anda membuatnya.

Penambahan dan penghapusan tanda pada sumber daya individu

AWS Batch memungkinkan Anda untuk menambahkan atau menghapus tanda yang berkaitan dengan klaster Anda langsung dari halaman sumber daya.

Untuk menambahkan atau menghapus sebuah tanda pada sebuah sumber daya individu

1. Buka konsol AWS Batch di <https://console.aws.amazon.com/batch/>.
2. Di bilah navigasi, pilih Wilayah yang akan digunakan.
3. Di panel navigasi, pilih tipe sumber daya (misalnya, Job Queues (Antrean Tugas)).
4. Pilih sumber daya tertentu, lalu pilih Edit tags (Edit tanda).

5. Tambah atau hapus tanda Anda sesuai kebutuhan.

- Untuk menambahkan tanda — tentukan kunci dan nilai dalam kotak teks kosong di akhir daftar.
- Untuk menghapus tanda — pilih tombol

Delete icon
di sebelah tanda.

6. Ulangi proses ini untuk setiap tanda yang ingin Anda tambah atau hapus, lalu pilih Edit tags (Edit tanda) untuk menyelesaikan.

Cara menggunakan tanda dengan menggunakan CLI atau API

Gunakan perintah AWS CLI atau operasi API AWS Batch berikut untuk menambahkan, memperbarui, membuat daftar, dan menghapus tanda sumber daya Anda.

Dukungan penandaan untuk sumber daya AWS Batch

Tugas	Tindakan API	AWS CLI	AWS Tools for Windows PowerShell
Tambahkan atau timpa satu atau beberapa tanda.	TagResource	tag-sumber daya	Add-bat ResourceTag
Hapus satu atau beberapa tanda.	UntagResource	untag-sumber daya	Hapus-kelelawar ResourceTag
Membuat daftar tanda untuk sumber daya	ListTagsForResource	list-tags-for-resource	Dapatkan kelelawar ResourceTag

Contoh-contoh berikut menunjukkan cara menambahkan atau menghilangkan tanda sumber daya menggunakan AWS CLI.

Contoh 1: Menandai sumber daya yang sudah ada

Perintah berikut ini menandai sumber daya yang sudah ada.

```
aws batch tag-resource --resource-arn resource_ARN --tags team=devs
```

Contoh 2: Menghapus tanda sumber daya yang sudah ada

Perintah berikut ini menghapus tanda dari sumber daya yang sudah ada.

```
aws batch untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Contoh 3: Cantumkan tanda untuk sumber daya

Perintah berikut akan mencantumkan tanda terkait dengan sumber daya yang sudah ada.

```
aws batch list-tags-for-resource --resource-arn resource_ARN
```

Beberapa tindakan pembuatan sumber daya memungkinkan Anda untuk menentukan tanda saat membuat sumber daya. Tindakan berikut mendukung penandaan saat pembuatan.

Tugas	Tindakan API	AWS CLI	AWS Tools for Windows PowerShell
Membuat lingkungan komputasi	CreateComputeEnvironment	create-compute-environment	Kelelawar baru ComputeEnvironment
Membuat antrean tugas	CreateJobQueue	create-job-queue	Kelelawar baru JobQueue
Buat kebijakan penjadwalan	CreateSchedulingPolicy	create-scheduling-policy	Kelelawar baru SchedulingPolicy
Mendaftarkan ketentuan tugas	RegisterJobDefinition	register-job-definition	Daftar-kelelawar JobDefinition
Mengirim tugas	SubmitJob	kirimkan pekerjaan	Kirim-Batjob

AWS Batch Service quotas

Tabel berikut menyediakan kuota layanan untuk AWS Batch yang tidak dapat diubah. Setiap kuota bersifat spesifik untuk Wilayah tertentu.

Sumber daya	Kuota
Jumlah maksimum antrean tugas. Untuk informasi selengkapnya, lihat Antrean tugas .	50
Jumlah maksimum lingkungan komputasi di Amazon ECS dan Amazon EKS. Untuk informasi selengkapnya, lihat Lingkungan komputasi .	50
Jumlah maksimum lingkungan komputasi per kluster Amazon EKS.	5
Jumlah maksimum lingkungan komputasi untuk setiap antrean tugas	3
Jumlah maksimum dependensi tugas untuk suatu tugas	20
Ukuran maksimum ketentuan tugas (untuk operasi API RegisterJobDefinition)	24 KiB
Ukuran maksimum muatan tugas (untuk operasi API SubmitJob)	30 KiB
Ukuran maksimum array untuk tugas array	10000
Jumlah maksimum tugas di status SUBMITTED	1000000
Jumlah maksimum transaksi per detik (TPS) untuk setiap akun untuk operasi SubmitJob	50

Tergantung pada cara Anda menggunakan AWS Batch, kuota tambahan mungkin berlaku. Untuk mempelajari tentang kuota Amazon EC2, lihat Amazon [EC2 Service Quotas](#) di. Referensi Umum AWS Untuk informasi selengkapnya tentang kuota Amazon ECS, lihat [Service Quotas Amazon ECS](#) di. Referensi Umum AWS Untuk informasi selengkapnya tentang kuota Amazon EKS, lihat [Amazon EKS Service Quotas](#) di. Referensi Umum AWS

Pemecahan masalah AWS Batch

Anda mungkin perlu memecahkan masalah yang terkait dengan lingkungan komputasi, antrian pekerjaan, definisi pekerjaan, atau pekerjaan Anda. Bab ini menjelaskan cara memecahkan masalah dan menyelesaikan masalah tersebut di lingkungan Anda AWS Batch .

AWS Batch menggunakan kebijakan, peran, dan izin IAM, dan berjalan di Amazon EC2, Amazon AWS Fargate ECS, dan infrastruktur Amazon Elastic Kubernetes Service. Untuk memecahkan masalah yang terkait dengan layanan ini, lihat berikut ini:

- [Memecahkan Masalah IAM di Panduan Pengguna IAM](#)
- [Pemecahan masalah Amazon ECS di Panduan Pengembang Layanan Amazon Elastic Container](#)
- [Pemecahan masalah Amazon EKS di Panduan Pengguna Amazon EKS](#)
- [Memecahkan masalah instans EC2 di Panduan Pengguna Amazon EC2 untuk Instans Linux](#)

Daftar Isi

- [AWS Batch](#)
 - [INVALIDlingkungan komputasi](#)
 - [Nama peran atau ARN salah](#)
 - [Memperbaiki lingkungan INVALID komputasi](#)
 - [Pekerjaan terjebak dalam RUNNABLE status](#)
 - [Instans Spot tidak ditandai pada pembuatan](#)
 - [Instans Spot tidak menskalakan ke bawah](#)
 - [Lampirkan kebijakan SpotFleetTaggingRole terkelola AmazonEC2 ke peran Armada Spot Anda di AWS Management Console](#)
 - [Lampirkan kebijakan SpotFleetTaggingRole terkelola AmazonEC2 ke peran Armada Spot Anda dengan AWS CLI](#)
 - [Tidak dapat mengambil rahasia Secrets Manager](#)
 - [Tidak dapat mengesampingkan persyaratan sumber daya definisi pekerjaan](#)
 - [Pesan galat saat Anda memperbarui desiredvCpus pengaturan](#)
- [AWS Batch di Amazon EKS](#)
 - [INVALIDlingkungan komputasi](#)

- [Versi yang tidak didukung Kubernetes](#)
- [Profil instance tidak ada](#)
- [Namespace tidak valid Kubernetes](#)
- [Lingkungan komputasi yang dihapus](#)
- [Node tidak bergabung dengan cluster Amazon EKS](#)
- [AWS Batch di Amazon EKS pekerjaan macet dalam RUNNABLE status](#)
- [Verifikasi bahwa aws-auth ConfigMap sudah dikonfigurasi dengan benar](#)
- [Izin atau binding RBAC tidak dikonfigurasi dengan benar](#)

AWS Batch

INVALID lingkungan komputasi

Ada kemungkinan bahwa Anda mungkin telah salah mengkonfigurasi lingkungan komputasi terkelola. Jika Anda melakukannya, lingkungan komputasi memasuki INVALID status dan tidak dapat menerima pekerjaan untuk penempatan. Bagian berikut menjelaskan kemungkinan penyebab dan cara memecahkan masalah berdasarkan penyebabnya.

Nama peran atau ARN salah

Penyebab paling umum lingkungan komputasi memasuki INVALID status adalah peran AWS Batch layanan atau peran Armada Spot Amazon EC2 memiliki nama yang salah atau Nama Sumber Daya Amazon (ARN). Ini lebih umum dengan lingkungan komputasi yang dibuat menggunakan AWS CLI atau AWS SDK. Saat Anda membuat lingkungan komputasi di AWS Management Console, AWS Batch membantu Anda memilih layanan yang benar atau peran Armada Spot. Namun, anggaplah Anda memasukkan nama atau ARN secara manual dan memasukkannya dengan tidak benar. Kemudian, lingkungan komputasi yang dihasilkan juga INVALID.

Namun, misalkan Anda memasukkan nama atau ARN secara manual untuk sumber daya IAM dalam AWS CLI perintah atau kode SDK Anda. Dalam hal ini, tidak AWS Batch dapat memvalidasi string. Sebaliknya, AWS Batch harus menerima nilai buruk dan berusaha menciptakan lingkungan. Jika AWS Batch gagal menciptakan lingkungan, lingkungan bergerak ke INVALID status, dan Anda melihat kesalahan berikut.

Untuk peran layanan yang tidak valid:

```
CLIENT_ERROR - Not authorized to perform sts:AssumeRole (Service:
AWSSecurityTokenService; Status Code: 403; Error Code: AccessDenied;
Request ID: dc0e2d28-2e99-11e7-b372-7fcc6fb65fe7)
```

Untuk peran Armada Spot yang tidak valid:

```
CLIENT_ERROR - Parameter: SpotFleetRequestConfig.IamFleetRole
is invalid. (Service: AmazonEC2; Status Code: 400; Error Code:
InvalidSpotFleetRequestConfig; Request ID: 331205f0-5ae3-4cea-
bac4-897769639f8d) Parameter: SpotFleetRequestConfig.IamFleetRole is
invalid
```

Salah satu penyebab umum untuk masalah ini adalah skenario berikut. Anda hanya menentukan nama peran IAM saat menggunakan AWS CLI atau AWS SDK, bukan Nama Sumber Daya Amazon (ARN) lengkap. Bergantung pada bagaimana Anda membuat peran, ARN mungkin berisi awalan `aws-service-role` jalur. Misalnya, jika Anda membuat peran AWS Batch layanan secara manual menggunakan prosedur di [Menggunakan peran terkait layanan untuk AWS Batch](#), ARN peran layanan Anda mungkin terlihat seperti berikut.

```
arn:aws:iam::123456789012:role/AWSBatchServiceRole
```

Namun, jika Anda membuat peran layanan sebagai bagian dari wizard run pertama konsol hari ini, ARN peran layanan Anda mungkin terlihat seperti berikut ini.

```
arn:aws:iam::123456789012:role/aws-service-role/AWSBatchServiceRole
```

Masalah ini juga dapat terjadi jika Anda melampirkan kebijakan AWS Batch tingkat layanan (`AWSBatchServiceRole`) ke peran non-layanan. Misalnya, Anda mungkin menerima pesan galat yang menyerupai berikut ini dalam skenario ini:

```
CLIENT_ERROR - User: arn:aws:sts::account_number:assumed-role/batch-replacement-role/
aws-batch is not
authorized to perform: action on resource ...
```

Untuk mengatasi masalah ini, lakukan salah satu hal berikut.

- Gunakan string kosong untuk peran layanan saat Anda membuat lingkungan AWS Batch komputasi.

- Tentukan peran layanan dalam format berikut: `arn:aws:iam::account_number:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`.

Jika Anda hanya menentukan nama peran IAM saat menggunakan AWS CLI atau AWS SDK, AWS Batch asumsikan bahwa ARN Anda tidak menggunakan awalan jalur. `aws-service-role` Karena itu, kami sarankan Anda menentukan ARN lengkap dari IAM role Anda ketika membuat lingkungan komputasi.

Untuk memperbaiki lingkungan komputasi yang salah dikonfigurasi seperti ini, lihat [Memperbaiki lingkungan INVALID komputasi](#).

Memperbaiki lingkungan **INVALID** komputasi

Bila Anda memiliki lingkungan komputasi dalam INVALID keadaan, perbarui untuk memperbaiki parameter yang tidak valid. Untuk [Nama peran atau ARN salah](#), perbarui lingkungan komputasi menggunakan peran layanan yang benar.

Untuk memperbaiki lingkungan komputasi yang salah dikonfigurasi

1. Buka AWS Batch konsol di <https://console.aws.amazon.com/batch/>.
2. Dari bilah navigasi, pilih yang Wilayah AWS akan digunakan.
3. Di panel navigasi, pilih Compute environments (Lingkungan komputasi).
4. Di halaman Compute environments (Lingkungan komputasi), pilih tombol radio di sebelah lingkungan komputasi untuk mengedit, lalu pilih Edit.
5. Di halaman Update compute environment (Perbarui lingkungan komputasi), untuk Service role (Peran layanan), pilih IAM role yang akan digunakan dengan lingkungan komputasi Anda. Konsol AWS Batch hanya menampilkan peran yang memiliki hubungan kepercayaan yang benar untuk lingkungan komputasi.
6. Pilih Save (Simpan) untuk memperbarui lingkungan komputasi Anda.

Pekerjaan terjebak dalam **RUNNABLE** status

Misalkan lingkungan komputasi Anda berisi sumber daya komputasi, tetapi pekerjaan Anda tidak berkembang melampaui status. **RUNNABLE** Kemudian, kemungkinan ada sesuatu yang mencegah pekerjaan ditempatkan pada sumber daya komputasi dan menyebabkan antrian pekerjaan Anda diblokir. Berikut cara mengetahui apakah pekerjaan Anda sedang menunggu giliran atau macet dan memblokir antrian.

Jika AWS Batch mendeteksi bahwa Anda memiliki `RUNNABLE` pekerjaan di kepala dan memblokir antrian, Anda akan menerima acara [antrian pekerjaan yang diblokir](#) dari Amazon CloudWatch Events dengan alasannya. Alasan yang sama juga diperbarui ke `statusReason` bidang sebagai bagian dari [ListJobs](#) dan panggilan [DescribeJobs](#) API.

Secara opsional, Anda dapat mengonfigurasi `jobStateTimeLimitActions` parameter melalui [CreateJobQueue](#) dan tindakan [UpdateJobQueue](#) API.

Note

Saat ini, satu-satunya tindakan yang dapat Anda gunakan `jobStateLimitActions.action` adalah membatalkan pekerjaan.

`jobStateTimeLimitActionsParameter` ini digunakan untuk menentukan serangkaian tindakan yang AWS Batch dilakukan pada pekerjaan dalam keadaan tertentu. Anda dapat mengatur ambang waktu dalam hitungan detik melalui `maxTimeSeconds` bidang.

Ketika pekerjaan telah dalam `RUNNABLE` keadaan dengan yang ditentukan `statusReason`, AWS Batch melakukan tindakan yang ditentukan setelah `maxTimeSeconds` telah berlalu.

Misalnya, Anda dapat mengatur `jobStateTimeLimitActions` parameter untuk menunggu hingga 4 jam untuk pekerjaan apa pun di `RUNNABLE` negara bagian yang menunggu kapasitas yang cukup tersedia. Anda dapat melakukan ini dengan mengatur `statusReason` ke `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY` dan `maxTimeSeconds` ke 144000 sebelum membatalkan pekerjaan dan memungkinkan pekerjaan berikutnya untuk maju ke kepala antrian pekerjaan.

Berikut ini adalah alasan yang AWS Batch memberikan ketika mendeteksi bahwa antrian pekerjaan diblokir. Daftar ini menyediakan pesan yang dikembalikan dari tindakan `ListJobs` dan `DescribeJobs` API. Ini juga merupakan nilai yang sama yang dapat Anda tentukan untuk `jobStateLimitActions.statusReason` parameter.

1. Alasan: Semua lingkungan komputasi yang terhubung memiliki kesalahan kapasitas yang tidak mencukupi. Saat diminta, AWS Batch mendeteksi instans Amazon EC2 yang mengalami kesalahan kapasitas tidak mencukupi. Membatalkan pekerjaan, baik secara manual atau dengan mengatur `jobStateTimeLimitActions` parameter `statusReason`, memungkinkan pekerjaan berikutnya untuk pindah ke kepala antrian.

- **statusReason** pesan saat pekerjaan macet:
CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY - Service cannot fulfill the capacity requested for instance type [instanceTypeName]
- **reason** digunakan untuk **jobStateTimeLimitActions**:
CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY
- **statusReason** pesan setelah pekerjaan dibatalkan: Canceled by JobStateTimeLimit action due to reason: CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY

Catatan:

- a. Peran AWS Batch layanan memerlukan `autoscaling:DescribeScalingActivities` izin agar deteksi ini berfungsi. Jika Anda menggunakan peran [AWSServiceRoleForBatch](#) terkait layanan (SLR) atau kebijakan [AWSBatchServiceRolePolicy](#) terkelola, Anda tidak perlu mengambil tindakan apa pun karena kebijakan izin mereka diperbarui.
 - b. Jika Anda menggunakan SLR atau kebijakan terkelola, Anda harus menambahkan `autoscaling:DescribeScalingActivities` dan `ec2:DescribeSpotFleetRequestHistory` izin agar Anda dapat menerima peristiwa antrian pekerjaan yang diblokir dan status pekerjaan yang diperbarui saat masuk. `RUNNABLE` Selain itu, AWS Batch perlu izin ini untuk melakukan `cancellation` tindakan melalui `jobStateTimeLimitActions` parameter bahkan jika mereka dikonfigurasi pada antrian pekerjaan.
 - c. Dalam kasus pekerjaan paralel multi-node (MNP), jika lingkungan komputasi Amazon EC2 prioritas tinggi yang dilampirkan mengalami `insufficient capacity` kesalahan, ini memblokir antrian meskipun lingkungan komputasi prioritas rendah mengalami kesalahan ini.
2. Alasan: Semua lingkungan komputasi memiliki `maxvCpus` parameter yang lebih kecil dari persyaratan pekerjaan. Membatalkan pekerjaan, baik secara manual atau dengan mengatur `jobStateTimeLimitActions` parameter `statusReason`, memungkinkan pekerjaan berikutnya untuk pindah ke kepala antrian. Secara opsional, Anda dapat meningkatkan `maxvCpus` parameter lingkungan komputasi utama untuk memenuhi kebutuhan pekerjaan yang diblokir.
 - **statusReason** pesan saat pekerjaan macet:
MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE - CE(s) associated with the job queue cannot meet the CPU requirement of the job.
 - **reason** digunakan untuk **jobStateTimeLimitActions**:
MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE
 - **statusReason** pesan setelah pekerjaan dibatalkan: Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE

3. Alasan: Tidak ada lingkungan komputasi yang memiliki instance yang memenuhi persyaratan pekerjaan. Ketika pekerjaan meminta sumber daya, AWS Batch mendeteksi bahwa tidak ada lingkungan komputasi terlampir yang dapat mengakomodasi pekerjaan yang masuk. Membatalkan pekerjaan, baik secara manual atau dengan mengatur `jobStateTimeLimitActions` parameter `statusReason`, memungkinkan pekerjaan berikutnya untuk pindah ke kepala antrian. Secara opsional, Anda dapat mendefinisikan ulang jenis instans yang diizinkan lingkungan komputasi untuk menambahkan sumber daya pekerjaan yang diperlukan.
 - **statusReason** pesan saat pekerjaan macet:
`MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT - The job resource requirement (vCPU/memory/GPU) is higher than that can be met by the CE(s) attached to the job queue.`
 - **reason** digunakan untuk `jobStateTimeLimitActions`:
`MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`
 - **statusReason** pesan setelah pekerjaan dibatalkan: `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`
4. Alasan: Semua lingkungan komputasi memiliki masalah peran layanan. Untuk mengatasinya, bandingkan izin peran layanan Anda dengan [izin peran layanan AWS Batch terkelola](#) dan atasi celah apa pun.

Ini adalah praktik terbaik untuk menggunakan [AWS Batch SLR untuk lingkungan komputasi](#) untuk menghindari kesalahan serupa.

Membatalkan pekerjaan, baik secara manual atau dengan mengatur `jobStateTimeLimitActions` parameter `statusReason`, memungkinkan pekerjaan berikutnya untuk pindah ke kepala antrian. Tanpa menyelesaikan masalah peran layanan, kemungkinan pekerjaan berikutnya juga akan diblokir. Yang terbaik adalah menyelidiki dan menyelesaikan masalah ini secara manual.

- **statusReason** pesan saat pekerjaan macet:
`MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS - Batch service role has a permission issue.`
- **reason** digunakan untuk `jobStateTimeLimitActions`:
`MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS`
- **statusReason** pesan setelah pekerjaan dibatalkan: `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS`

5. Alasan: Semua lingkungan komputasi tidak valid. Untuk informasi selengkapnya, lihat [lingkungan INVALID komputasi](#). Catatan: Anda tidak dapat mengonfigurasi tindakan yang dapat diprogram melalui `jobStateTimeLimitActions` parameter untuk mengatasi kesalahan ini.
 - **statusReason** pesan saat pekerjaan macet: `ACTION_REQUIRED - CE(s) associated with the job queue are invalid.`
6. Alasan: AWS Batch telah mendeteksi antrian yang diblokir, tetapi tidak dapat menentukan alasannya. Catatan: Anda tidak dapat mengonfigurasi tindakan yang dapat diprogram melalui `jobStateTimeLimitActions` parameter untuk mengatasi kesalahan ini. Untuk informasi selengkapnya tentang pemecahan masalah, lihat [Mengapa AWS Batch pekerjaan saya macet di RUNNABLE di RE:POST. AWS](#)
 - **statusReason** pesan saat pekerjaan macet: `UNDETERMINED - Batch job is blocked, root cause is undetermined.`

Jika Anda tidak menerima acara dari Acara atau Anda menerima CloudWatch peristiwa alasan yang tidak diketahui, berikut adalah beberapa penyebab umum untuk masalah ini.

Driver `awslogs` log tidak dikonfigurasi pada sumber daya komputasi Anda

AWS Batch pekerjaan mengirim informasi log mereka ke CloudWatch Log. Untuk mengaktifkan ini, Anda harus mengonfigurasi sumber daya komputasi Anda untuk menggunakan driver log `awslogs`. Misalkan Anda mendasarkan sumber daya komputasi AMI dari AMI yang dioptimalkan Amazon ECS (atau Amazon Linux). Kemudian, driver ini terdaftar secara default dengan `ecs-init` paket. Sekarang anggaplah Anda menggunakan basis AMI yang berbeda. Kemudian, Anda harus memverifikasi bahwa driver `awslogs` log ditentukan sebagai driver log yang tersedia dengan variabel `ECS_AVAILABLE_LOGGING_DRIVERS` lingkungan saat agen penampung Amazon ECS dimulai. Lihat informasi yang lebih lengkap di [Spesifikasi AMI sumber daya komputasi](#) dan [Membuat AMI sumber daya komputasi](#).

Sumber daya yang tidak mencukupi

Jika definisi pekerjaan Anda menentukan lebih banyak sumber daya CPU atau memori daripada sumber daya komputasi yang dapat dialokasikan, maka pekerjaan Anda tidak akan pernah ditempatkan. Misalnya, pekerjaan Anda menentukan 4 GiB memori, dan sumber daya komputasi Anda memiliki kurang dari yang tersedia. Maka itu adalah kasus bahwa pekerjaan tidak dapat ditempatkan pada sumber daya komputasi tersebut. Dalam hal ini, Anda harus mengurangi memori yang ditentukan dalam ketentuan tugas Anda atau menambahkan sumber daya komputasi yang lebih besar ke lingkungan Anda. Beberapa memori disimpan untuk agen

kontainer Amazon ECS dan proses sistem penting lainnya. Untuk informasi selengkapnya, lihat [Manajemen Memori Sumber Daya Komputasi](#).

Tidak ada akses internet untuk sumber daya komputasi

Sumber daya komputasi memerlukan akses untuk berkomunikasi dengan titik akhir layanan Amazon ECS. Ini dapat dilakukan melalui VPC endpoint antarmuka atau melalui sumber daya komputasi yang memiliki alamat IP publik.

Untuk informasi lebih lanjut tentang VPC endpoint antarmuka, lihat [VPC Endpoint Antarmuka Amazon ECS \(AWS PrivateLink\)](#) dalam Panduan Developer Amazon Elastic Container Service.

Jika Anda tidak memiliki VPC endpoint yang dikonfigurasi dan sumber daya komputasi Anda tidak memiliki alamat IP publik, network address translation (NAT) harus digunakan untuk menyediakan akses ini. Untuk informasi lebih lanjut, lihat [Gateway NAT](#) dalam Panduan Pengguna Amazon VPC. Untuk informasi selengkapnya, lihat [the section called "Buat VPC"](#).

Batas instans Amazon EC2 tercapai

Jumlah instans Amazon EC2 yang dapat diluncurkan akun Anda ditentukan oleh kuota instans Wilayah AWS EC2 Anda. Jenis instans tertentu juga memiliki per-instance-type kuota. Untuk informasi selengkapnya tentang kuota instans Amazon EC2 akun Anda termasuk cara meminta peningkatan batas, lihat [Batas Layanan Amazon EC2 di](#) Panduan Pengguna Amazon EC2 untuk Instans Linux.

Agan kontainer Amazon ECS tidak diinstal

Agan kontainer Amazon ECS harus diinstal pada Amazon Machine Image (AMI) untuk membiarkan pekerjaan AWS Batch berjalan. Agen kontainer Amazon ECS diinstal secara default di AMI Amazon ECS yang dioptimalkan. Untuk informasi selengkapnya tentang agen kontainer Amazon ECS, lihat [Agen kontainer Amazon ECS](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.

Untuk informasi selengkapnya, lihat [Mengapa AWS Batch pekerjaan saya terjebak dalam RUNNABLE status?](#) di re:post.

Instans Spot tidak ditandai pada pembuatan

Penandaan Instans Spot untuk sumber daya AWS Batch komputasi didukung mulai 25 Oktober 2017. Sebelumnya, kebijakan terkelola IAM (`AmazonEC2SpotFleetRole`) yang direkomendasikan untuk

peran Armada Spot Amazon EC2 tidak berisi izin untuk menandai Instans Spot saat diluncurkan. Kebijakan terkelola IAM baru yang disarankan disebut `AmazonEC2SpotFleetTaggingRole`. Ini mendukung penandaan Instans Spot saat peluncuran.

Untuk memperbaiki penandaan Instans Spot saat pembuatan, ikuti prosedur berikut untuk menerapkan kebijakan terkelola IAM yang direkomendasikan saat ini ke peran Armada Spot Amazon EC2 Anda. Dengan begitu, Instans Spot masa depan yang dibuat dengan peran tersebut memiliki izin untuk menerapkan tag instance saat dibuat.

Untuk menerapkan kebijakan terkelola IAM saat ini pada peran Armada Spot Amazon EC2 Anda

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Pilih Roles (Peran), dan pilih peran Armada Spot Amazon EC2 Anda.
3. Pilih Lampirkan kebijakan.
4. Pilih `AmazonEC2SpotFleetTaggingRole` dan pilih Lampirkan kebijakan.
5. Pilih peran Armada Spot Amazon EC2 Anda lagi untuk menghapus kebijakan sebelumnya.
6. Pilih x di sebelah kanan `SpotFleetRole` kebijakan AmazonEC2, dan pilih Lepaskan.

Instans Spot tidak menskalakan ke bawah

AWS Batch memperkenalkan peran `AWSServiceRoleForBatch` terkait layanan pada 10 Maret 2021. Jika tidak ada peran yang ditentukan dalam `serviceRole` parameter lingkungan komputasi, peran terkait layanan ini digunakan sebagai peran layanan. Namun, misalkan peran terkait layanan digunakan dalam lingkungan komputasi EC2 Spot, tetapi peran Spot yang digunakan tidak menyertakan kebijakan terkelola `AmazonEC2SpotFleetTaggingRole`. Kemudian, Instance Spot tidak menurunkan skala. Akibatnya, Anda akan menerima kesalahan dengan pesan berikut: "Anda tidak berwenang untuk melakukan operasi ini." Gunakan langkah-langkah berikut untuk memperbarui peran armada spot yang Anda gunakan dalam `spotIamFleetRole` parameter. Untuk informasi selengkapnya, lihat [Menggunakan peran terkait layanan](#) dan [Membuat peran untuk mendelegasikan izin ke AWS Layanan di Panduan Pengguna IAM](#).

Topik

- [Lampirkan kebijakan `SpotFleetTaggingRole` terkelola AmazonEC2 ke peran Armada Spot Anda di AWS Management Console](#)
- [Lampirkan kebijakan `SpotFleetTaggingRole` terkelola AmazonEC2 ke peran Armada Spot Anda dengan AWS CLI](#)

Lampirkan kebijakan SpotFleetTaggingRole terkelola AmazonEC2 ke peran Armada Spot Anda di AWS Management Console

Untuk menerapkan kebijakan terkelola IAM saat ini pada peran Armada Spot Amazon EC2 Anda

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Pilih Roles (Peran), dan pilih peran Armada Spot Amazon EC2 Anda.
3. Pilih Lampirkan kebijakan.
4. Pilih AmazonEC2 SpotFleetTaggingRole dan pilih Lampirkan kebijakan.
5. Pilih peran Armada Spot Amazon EC2 Anda lagi untuk menghapus kebijakan sebelumnya.
6. Pilih x di sebelah kanan SpotFleetRole kebijakan AmazonEC2, dan pilih Lepaskan.

Lampirkan kebijakan SpotFleetTaggingRole terkelola AmazonEC2 ke peran Armada Spot Anda dengan AWS CLI

Perintah contoh mengasumsikan bahwa peran Armada Spot Amazon EC2 Anda diberi nama `AmazonEC2SpotFleetRole`. Jika peran Anda menggunakan nama yang berbeda, sesuaikan perintah untuk mencocokkannya.

Untuk melampirkan kebijakan SpotFleetTaggingRole terkelola AmazonEC2 ke peran Armada Spot Anda

1. Untuk melampirkan kebijakan IAM SpotFleetTaggingRole terkelola AmazonEC2 ke SpotFleetRole peran `AmazonEC2` Anda, jalankan perintah berikut menggunakan AWS CLI

```
$ aws iam attach-role-policy \  
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \  
  --role-name AmazonEC2SpotFleetRole
```

2. Untuk melepaskan kebijakan IAM SpotFleetRole terkelola AmazonEC2 dari SpotFleetRole peran `AmazonEC2` Anda, jalankan perintah berikut menggunakan AWS CLI

```
$ aws iam detach-role-policy \  
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetRole \  
  --role-name AmazonEC2SpotFleetRole
```

Tidak dapat mengambil rahasia Secrets Manager

Jika Anda menggunakan AMI dengan agen Amazon ECS yang lebih awal dari versi 1.16.0-1, maka Anda harus menggunakan variabel `ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true` konfigurasi agen Amazon ECS untuk menggunakan fitur ini. Anda dapat menambahkannya ke `./etc/ecs/ecs.config` file ke instance container baru saat Anda membuat instance itu. Atau, Anda dapat menambahkannya ke instance yang ada. Jika Anda menambahkannya ke instance yang ada, Anda harus me-restart agen ECS setelah Anda menambahkannya. Untuk informasi lebih lanjut, lihat [Kontainer Amazon ECS Container Agent](#) di Panduan Developer Amazon Elastic Container Service.

Tidak dapat mengesampingkan persyaratan sumber daya definisi pekerjaan

[Penggantian memori dan vCPU yang ditentukan dalam `memory` dan `vcpus` anggota struktur `containerOverrides`, yang diteruskan ke, tidak `SubmitJob` dapat mengganti persyaratan memori dan vCPU yang ditentukan dalam struktur `ResourceRequirements` dalam definisi pekerjaan.](#)

Jika Anda mencoba mengganti persyaratan sumber daya ini, Anda mungkin melihat pesan galat berikut:

“Nilai ini dikirimkan dalam kunci yang tidak digunakan lagi dan mungkin bertentangan dengan nilai yang diberikan oleh persyaratan sumber daya definisi pekerjaan.”

[Untuk memperbaikinya, tentukan persyaratan memori dan vCPU di anggota `ResourceRequirements` dari `ContainerOverrides`.](#) Misalnya, jika penggantian memori dan vCPU Anda ditentukan dalam baris berikut.

```
"containerOverrides": {
  "memory": 8192,
  "vcpus": 4
}
```

Ubah menjadi yang berikut:

```
"containerOverrides": {
  "resourceRequirements": [
    {
      "type": "MEMORY",
      "value": "8192"
    },
    {
      "type": "VCPU",
```

```

        "value": "4"
      }
    ],
  }

```

Lakukan perubahan yang sama pada memori dan persyaratan vCPU yang ditentukan dalam objek [ContainerProperties](#) dalam definisi pekerjaan. Misalnya, jika persyaratan memori dan vCPU Anda ditentukan pada baris berikut.

```

{
  "containerProperties": {
    "memory": 4096,
    "vcpus": 2,
  }
}

```

Ubah menjadi yang berikut:

```

"containerProperties": {
  "resourceRequirements": [
    {
      "type": "MEMORY",
      "value": "4096"
    },
    {
      "type": "VCPU",
      "value": "2"
    }
  ],
}

```

Pesan galat saat Anda memperbarui **desiredvCpus** pengaturan

Anda melihat pesan galat berikut saat menggunakan AWS Batch API untuk memperbarui setelan `desiredvCpus` vCPUs () yang diinginkan.

Manually scaling down compute environment is not supported. Disconnecting job queues from compute environment will cause it to scale-down to `minvCpus`.

Masalah ini terjadi jika `desiredvCpus` nilai yang diperbarui kurang dari `desiredvCpus` nilai saat ini. Saat Anda memperbarui `desiredvCpus` nilainya, kedua hal berikut harus benar:

- `desiredVCpusNilai` harus antara `maxVCpus` nilai `minVCpus` dan.
- `desiredVCpusNilai` yang diperbarui harus lebih besar dari atau sama dengan `desiredVCpus` nilai saat ini.

AWS Batch di Amazon EKS

Topik

- [INVALIDlingkungan komputasi](#)
- [AWS Batch di Amazon EKS pekerjaan macet dalam RUNNABLE status](#)
- [Verifikasi bahwa aws-auth ConfigMap sudah dikonfigurasi dengan benar](#)
- [Izin atau binding RBAC tidak dikonfigurasi dengan benar](#)

INVALIDlingkungan komputasi

Ada kemungkinan bahwa Anda mungkin telah salah mengkonfigurasi lingkungan komputasi terkelola. Jika Anda melakukannya, lingkungan komputasi memasuki INVALID status dan tidak dapat menerima pekerjaan untuk penempatan. Bagian berikut menjelaskan kemungkinan penyebab dan cara memecahkan masalah berdasarkan penyebabnya.

Versi yang tidak didukung Kubernetes

Anda mungkin melihat pesan galat yang menyerupai berikut ini saat menggunakan operasi `CreateComputeEnvironment` API atau operasi `UpdateComputeEnvironment` API untuk membuat atau memperbarui lingkungan komputasi. Masalah ini terjadi jika Anda menentukan Kubernetes versi yang tidak didukung di `EC2Configuration`.

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Untuk mengatasi masalah ini, hapus lingkungan komputasi lalu buat ulang dengan versi yang didukung Kubernetes.

Anda dapat melakukan upgrade versi minor pada kluster Amazon EKS Anda. Misalnya, Anda dapat memutakhirkan cluster dari `1.xx` ke `1.yy` meskipun versi minor tidak didukung.

Namun, status lingkungan komputasi mungkin berubah menjadi INVALID setelah pembaruan versi utama. Misalnya, jika Anda melakukan upgrade versi utama dari `1.xx` ke `2.yy`. Jika versi utama tidak didukung oleh AWS Batch, Anda akan melihat pesan galat yang menyerupai berikut ini.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Untuk mengatasi masalah ini, tentukan Kubernetes versi yang didukung saat Anda menggunakan operasi API untuk membuat atau memperbarui lingkungan komputasi.

AWS Batch di Amazon EKS saat ini mendukung Kubernetes versi berikut:

- 1.29
- 1.28
- 1.27
- 1.26
- 1.25
- 1.24
- 1.23

Profil instance tidak ada

Jika profil instance yang ditentukan tidak ada, status lingkungan komputasi AWS Batch di Amazon EKS akan diubah menjadi `INVALID`. Anda melihat set kesalahan dalam `statusReason` parameter yang menyerupai berikut ini.

```
CLIENT_ERROR - Instance profile arn:aws:iam:.....:instance-profile/<name> does not exist
```

Untuk mengatasi masalah ini, tentukan atau buat profil instans kerja. Untuk informasi lebih lanjut, lihat [IAM role simpel Amazon EKS](#) di Panduan Pengguna Amazon EKS.

Namespace tidak valid Kubernetes

Jika AWS Batch di Amazon EKS tidak dapat memvalidasi namespace untuk lingkungan komputasi, status lingkungan komputasi akan diubah menjadi `INVALID`. Misalnya, masalah ini dapat terjadi jika namespace tidak ada.

Anda melihat pesan galat diatur dalam `statusReason` parameter yang menyerupai berikut ini.

```
CLIENT_ERROR - Unable to validate Kubernetes Namespace
```

Masalah ini dapat terjadi jika salah satu dari berikut ini benar:

- String Kubernetes namespace dalam `CreateComputeEnvironment` panggilan tidak ada. Untuk informasi lebih lanjut, lihat [CreateComputeEnvironment](#).
- Izin Kontrol Akses Berbasis Peran (RBAC) yang diperlukan untuk mengelola namespace tidak dikonfigurasi dengan benar.
- AWS Batch tidak memiliki akses ke titik akhir server Amazon EKS Kubernetes API.

Untuk mengatasi masalah ini, lihat [Verifikasi bahwa aws-auth ConfigMap sudah dikonfigurasi dengan benar](#). Untuk informasi selengkapnya, lihat [Memulai dengan AWS Batch di Amazon EKS](#).

Lingkungan komputasi yang dihapus

Misalkan Anda menghapus kluster Amazon EKS sebelum menghapus lingkungan komputasi AWS Batch Amazon EKS yang dilampirkan. Kemudian, status lingkungan komputasi diubah menjadi `INVALID`. Dalam skenario ini, lingkungan komputasi tidak berfungsi dengan baik jika Anda membuat ulang cluster Amazon EKS dengan nama yang sama.

Untuk mengatasi masalah ini, hapus lalu buat ulang lingkungan komputasi AWS Batch Amazon EKS.

Node tidak bergabung dengan cluster Amazon EKS

AWS Batch di Amazon EKS menurunkan lingkungan komputasi jika menentukan bahwa tidak semua node bergabung dengan cluster Amazon EKS. Saat AWS Batch di Amazon EKS menurunkan skala lingkungan komputasi, status lingkungan komputasi diubah menjadi `INVALID`.

Note

AWS Batch tidak segera mengubah status lingkungan komputasi sehingga Anda dapat men-debug masalah.

Anda melihat pesan kesalahan diatur dalam `statusReason` parameter yang menyerupai salah satu dari berikut ini:

```
Your compute environment has been INVALIDATED and scaled down because none of the instances joined the underlying ECS Cluster. Common issues preventing instances joining are the following: VPC/Subnet configuration preventing communication to ECS, incorrect Instance Profile policy preventing authorization to ECS, or customized AMI or LaunchTemplate configurations affecting ECS agent.
```


Your compute environment has been **INVALIDATED** and scaled down because none of the nodes joined the underlying Amazon EKS Cluster. Common issues preventing nodes joining are the following: networking configuration preventing communication to Amazon EKS Cluster, incorrect Amazon EKS Instance Profile or Kubernetes RBAC policy preventing authorization to Amazon EKS Cluster, customized AMI or LaunchTemplate configurations affecting Amazon EKS/Kubernetes node bootstrap.

Saat menggunakan Amazon EKS AMI default, penyebab paling umum dari masalah ini adalah sebagai berikut:

- Peran instance tidak dikonfigurasi dengan benar. Untuk informasi lebih lanjut, lihat [IAM role simpel Amazon EKS](#) di Panduan Pengguna Amazon EKS.
- Subnet tidak dikonfigurasi dengan benar. Untuk informasi selengkapnya, lihat [Persyaratan dan pertimbangan Amazon EKS VPC dan subnet di Panduan Pengguna](#) Amazon EKS.
- Grup keamanan tidak dikonfigurasi dengan benar. Untuk informasi selengkapnya, lihat [persyaratan dan pertimbangan grup keamanan Amazon EKS](#) di Panduan Pengguna Amazon EKS.

Note

Anda juga dapat melihat pemberitahuan kesalahan di Dashboard Personal Health (PHD).

AWS Batch di Amazon EKS pekerjaan macet dalam **RUNNABLE** status

An `aws-auth` ConfigMap secara otomatis dibuat dan diterapkan ke cluster Anda ketika Anda membuat grup node terkelola atau grup node menggunakan `eksctl`. Awalnya `aws-auth` ConfigMap dibuat untuk memungkinkan node bergabung dengan cluster Anda. Namun, Anda juga menggunakan `aws-auth` ConfigMap untuk menambahkan akses kontrol akses berbasis peran (RBAC) ke pengguna dan peran.

Untuk memverifikasi bahwa `aws-auth` ConfigMap dikonfigurasi dengan benar:

1. Ambil peran yang dipetakan di: `aws-auth` ConfigMap

```
$ kubectl get configmap -n kube-system aws-auth -o yaml
```

2. Verifikasi bahwa `roleARN` dikonfigurasi sebagai berikut.

```
rolearn: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
```

Note

Anda juga dapat meninjau log pesawat kontrol Amazon EKS. Untuk informasi selengkapnya, lihat [pencatatan pesawat kontrol Amazon EKS](#) di Panduan Pengguna Amazon EKS.

Untuk mengatasi masalah di mana pekerjaan macet dalam `RUNNABLE` status, sebaiknya gunakan `kubectl` untuk menerapkan kembali manifes. Untuk informasi selengkapnya, lihat [Langkah 1: Mempersiapkan cluster Amazon EKS Anda AWS Batch](#). Atau, Anda dapat menggunakan `kubectl` untuk mengedit file secara manual `aws-authConfigMap`. Untuk informasi selengkapnya, lihat [Mengaktifkan pengguna IAM dan akses peran ke kluster Anda](#) di Panduan Pengguna Amazon EKS.

Verifikasi bahwa `aws-auth ConfigMap` sudah dikonfigurasi dengan benar

Untuk memverifikasi bahwa `aws-auth ConfigMap` dikonfigurasi dengan benar:

1. Ambil peran yang dipetakan di `aws-auth ConfigMap`

```
$ kubectl get configmap -n kube-system aws-auth -o yaml
```

2. Verifikasi bahwa `roleARN` dikonfigurasi sebagai berikut.

```
rolearn: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
```

Note

Jalur `aws-service-role/batch.amazonaws.com/` telah dihapus dari ARN dari peran terkait layanan. Ini karena masalah dengan peta `aws-auth` konfigurasi. Untuk informasi selengkapnya, lihat [Peran dengan jalur tidak berfungsi saat jalur disertakan dalam ARN mereka di .aws-auth configmap](#)

Note

Anda juga dapat meninjau log pesawat kontrol Amazon EKS. Untuk informasi selengkapnya, lihat [pencatatan pesawat kontrol Amazon EKS](#) di Panduan Pengguna Amazon EKS.

Untuk mengatasi masalah di mana pekerjaan macet dalam `RUNNABLE` status, sebaiknya gunakan `kubectl` untuk menerapkan kembali manifes. Untuk informasi selengkapnya, lihat [Langkah 1: Mempersiapkan cluster Amazon EKS Anda AWS Batch](#). Atau, Anda dapat menggunakan `kubectl` untuk mengedit file secara manual `aws-authConfigMap`. Untuk informasi selengkapnya, lihat [Mengaktifkan pengguna IAM dan akses peran ke kluster Anda](#) di Panduan Pengguna Amazon EKS.

Izin atau binding RBAC tidak dikonfigurasi dengan benar

Jika Anda mengalami izin RBAC atau masalah pengikatan, verifikasi bahwa `aws-batch` Kubernetes peran tersebut dapat mengakses namespace: Kubernetes

```
$ kubectl get namespace namespace --as=aws-batch
```

```
$ kubectl auth can-i get ns --as=aws-batch
```

Anda juga dapat menggunakan `kubectl describe` perintah untuk melihat otorisasi untuk peran kluster atau Kubernetes namespace.

```
$ kubectl describe clusterrole aws-batch-cluster-role
```

Berikut ini adalah output contoh.

```
Name:          aws-batch-cluster-role
Labels:        <none>
Annotations:   <none>
PolicyRule:
  Resources                Non-Resource URLs  Resource Names
  Verbs                    -----
  -----
```

```

configmaps          []          []
[get list watch]
nodes               []          []
[get list watch]
pods                []          []
[get list watch]
daemonsets.apps    []          []
[get list watch]
deployments.apps   []          []
[get list watch]
replicasets.apps   []          []
[get list watch]
statefulsets.apps  []          []
[get list watch]
clusterrolebindings.rbac.authorization.k8s.io []          []
[get list]
clusterroles.rbac.authorization.k8s.io []          []
[get list]
namespaces          []          []
[get]

```

```
$ kubectl describe role aws-batch-compute-environment-role -n my-aws-batch-namespace
```

Berikut ini adalah output contoh.

```

Name:          aws-batch-compute-environment-role
Labels:        <none>
Annotations:   <none>
PolicyRule:
  Resources          Non-Resource URLs  Resource Names      Verbs
  -----
  pods               []                  []                  [create
get list watch delete patch]
  serviceaccounts   []                  []                  [get list]
  rolebindings.rbac.authorization.k8s.io []                  []                  [get list]
  roles.rbac.authorization.k8s.io []                  []                  [get list]

```

Untuk mengatasi masalah ini, terapkan kembali izin dan perintah RBAC. `rolebinding` Untuk informasi selengkapnya, lihat [Langkah 1: Mempersiapkan cluster Amazon EKS Anda AWS Batch](#).

Praktik Terbaik untuk AWS Batch

Anda dapat menggunakan AWS Batch untuk menjalankan berbagai beban kerja komputasi yang menuntut dalam skala besar tanpa mengelola arsitektur yang kompleks. AWS Batch pekerjaan dapat digunakan dalam berbagai kasus penggunaan di berbagai bidang seperti epidemiologi, game, dan pembelajaran mesin.

Topik ini mencakup praktik terbaik untuk dipertimbangkan saat menggunakan AWS Batch dan panduan tentang cara menjalankan dan mengoptimalkan beban kerja Anda saat menggunakan AWS Batch.

Topik

- [Kapan harus menggunakan AWS Batch](#)
- [Daftar periksa untuk dijalankan pada skala](#)
- [Optimalkan wadah dan AMI](#)
- [Pilih sumber daya lingkungan komputasi yang tepat](#)
- [Amazon EC2 Sesuai Permintaan atau Amazon EC2 Spot](#)
- [Gunakan praktik terbaik Amazon EC2 Spot untuk AWS Batch](#)
- [Kesalahan umum dan pemecahan masalah](#)

Kapan harus menggunakan AWS Batch

AWS Batch menjalankan pekerjaan dalam skala besar dan dengan biaya rendah, dan menyediakan layanan antrian dan penskalaan yang dioptimalkan biaya. Namun, tidak setiap beban kerja cocok untuk dijalankan menggunakan AWS Batch.

- Pekerjaan pendek — Jika pekerjaan berjalan hanya beberapa detik, biaya overhead untuk menjadwalkan pekerjaan batch mungkin memakan waktu lebih lama daripada runtime pekerjaan itu sendiri. Sebagai solusinya, tugas binpack Anda bersama sebelum Anda mengirimkannya. AWS Batch Kemudian, konfigurasi AWS Batch pekerjaan Anda untuk mengulangi tugas. Misalnya, masukkan argumen tugas individual ke dalam tabel Amazon DynamoDB atau sebagai file di bucket Amazon S3. Pertimbangkan pengelompokan tugas sehingga pekerjaan masing-masing berjalan 3-5 menit. Setelah Anda binpack melakukan pekerjaan, putar melalui kelompok tugas Anda dalam AWS Batch pekerjaan Anda.

- Pekerjaan yang harus segera dijalankan — AWS Batch dapat memproses pekerjaan dengan cepat. Namun, AWS Batch adalah penjadwal dan mengoptimalkan kinerja biaya, prioritas pekerjaan, dan throughput. AWS Batch mungkin memerlukan waktu untuk memproses permintaan Anda. Jika Anda memerlukan respons dalam waktu kurang dari beberapa detik, maka pendekatan berbasis layanan menggunakan Amazon ECS atau Amazon EKS lebih cocok.

Daftar periksa untuk dijalankan pada skala

Sebelum Anda menjalankan beban kerja yang besar pada 50 ribu atau lebih vCPU, pertimbangkan daftar periksa berikut.

Note

Jika Anda berencana untuk menjalankan beban kerja besar pada satu juta atau lebih vCPU atau membutuhkan panduan berjalan dalam skala besar, hubungi tim Anda. AWS

- Periksa kuota Amazon EC2 Anda - Periksa kuota Amazon EC2 Anda (juga dikenal sebagai batas) di panel Service Quotas pada AWS Management Console. Jika perlu, minta peningkatan kuota untuk jumlah puncak instans Amazon EC2 Anda. Ingatlah bahwa instans Amazon EC2 Spot dan Amazon On-Demand memiliki kuota terpisah. Untuk informasi selengkapnya, lihat [Memulai Service Quotas](#).
- Verifikasi kuota Amazon Elastic Block Store untuk setiap Wilayah — Setiap instans menggunakan volume GP2 atau GP3 untuk sistem operasi. Secara default, kuota untuk masing-masing Wilayah AWS adalah 300 TiB. Namun, setiap instance menggunakan hitungan sebagai bagian dari kuota ini. Jadi, pastikan untuk memperhitungkan hal ini saat Anda memverifikasi kuota Amazon Elastic Block Store untuk setiap Wilayah. Jika kuota tercapai, Anda tidak dapat membuat lebih banyak instance. Untuk informasi selengkapnya, lihat [titik akhir dan kuota Amazon Elastic Block Store](#)
- Gunakan Amazon S3 untuk penyimpanan — Amazon S3 memberikan throughput tinggi dan membantu menghilangkan dugaan tentang berapa banyak penyimpanan yang akan disediakan berdasarkan jumlah pekerjaan dan instans di setiap Availability Zone. Untuk informasi selengkapnya, lihat [Pola desain praktik terbaik: mengoptimalkan kinerja Amazon S3](#).
- Skala secara bertahap untuk mengidentifikasi kemacetan lebih awal — Untuk pekerjaan yang berjalan pada satu juta atau lebih vCPU, mulailah lebih rendah dan secara bertahap tingkatkan sehingga Anda dapat mengidentifikasi kemacetan lebih awal. Misalnya, mulailah dengan menjalankan 50 ribu vCPU. Kemudian, tingkatkan hitungan menjadi 200 ribu vCPU, dan kemudian

500 ribu vCPU, dan seterusnya. Dengan kata lain, terus tingkatkan jumlah vCPU secara bertahap hingga Anda mencapai jumlah vCPU yang diinginkan.

- Pantau untuk mengidentifikasi potensi masalah lebih awal — Untuk menghindari potensi jeda dan masalah saat berjalan dalam skala besar, pastikan untuk memantau aplikasi dan arsitektur Anda. Istirahat dapat terjadi bahkan ketika penskalaan dari 1.000 hingga 5 ribu vCPU. Anda dapat menggunakan CloudWatch Log Amazon untuk meninjau data log atau menggunakan Metrik CloudWatch Tertanam menggunakan pustaka klien. Untuk informasi selengkapnya, lihat [Referensi agen CloudWatch Log](#) dan [aws-embedded-metrics](#)

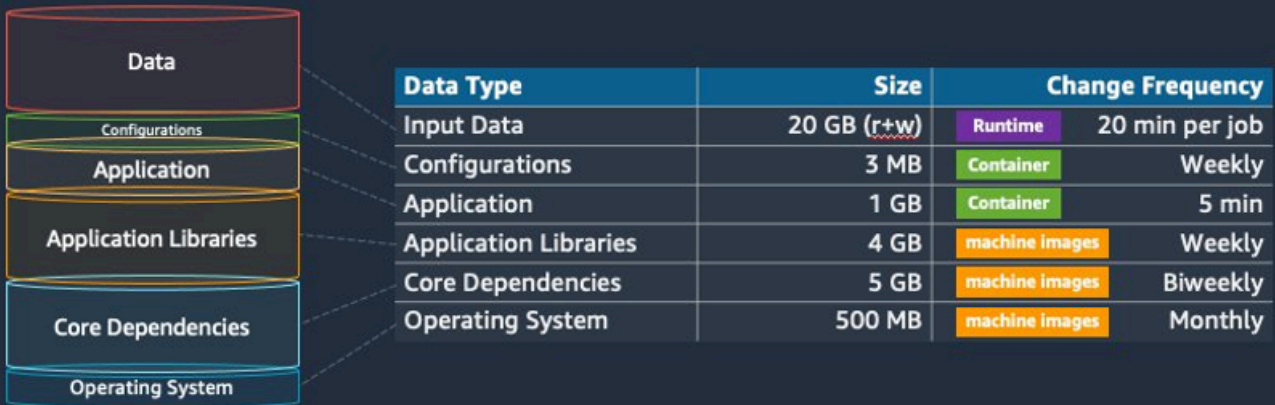
Optimalkan wadah dan AMI

Ukuran dan struktur kontainer penting untuk rangkaian pekerjaan pertama yang Anda jalankan. Ini terutama benar jika wadahnya lebih besar dari 4 GB. Gambar kontainer dibangun berlapis-lapis. Lapisan diambil secara paralel oleh Docker menggunakan tiga utas bersamaan. Anda dapat meningkatkan jumlah utas bersamaan menggunakan `max-concurrent-downloads` parameter. Untuk informasi selengkapnya, lihat [dokumentasi Dockerd](#).

Meskipun Anda dapat menggunakan kontainer yang lebih besar, kami menyarankan Anda mengoptimalkan struktur dan ukuran kontainer untuk waktu startup yang lebih cepat.

- Kontainer yang lebih kecil diambil lebih cepat — Kontainer yang lebih kecil dapat menyebabkan waktu mulai aplikasi lebih cepat. Untuk mengurangi ukuran kontainer, bongkar pustaka atau file yang jarang diperbarui ke Amazon Machine Image (AMI). Anda juga dapat menggunakan `bind mount` untuk memberikan akses ke wadah Anda. Untuk informasi lebih lanjut, lihat [Bind mount](#).
- Membuat layer yang bahkan dalam ukuran dan memecah layer besar - Setiap layer diambil oleh satu thread. Jadi, lapisan besar mungkin berdampak signifikan pada waktu startup pekerjaan Anda. Kami merekomendasikan ukuran lapisan maksimum 2 GB sebagai tradeoff yang baik antara ukuran wadah yang lebih besar dan waktu startup yang lebih cepat. Anda dapat menjalankan `docker history your_image_id` perintah untuk memeriksa struktur gambar kontainer dan ukuran lapisan Anda. Untuk informasi selengkapnya, lihat [Dokumentasi Docker](#).
- Gunakan Amazon Elastic Container Registry sebagai repositori container Anda — Saat Anda menjalankan ribuan pekerjaan secara paralel, repositori yang dikelola sendiri dapat gagal atau menghambat throughput. Amazon ECR bekerja dalam skala besar dan dapat menangani beban kerja hingga lebih dari satu juta vCPU.

Using machine images and containers



Pilih sumber daya lingkungan komputasi yang tepat

AWS Fargate memerlukan pengaturan dan konfigurasi awal yang lebih sedikit daripada Amazon EC2 dan kemungkinan lebih mudah digunakan, terutama jika ini adalah pertama kalinya Anda. Dengan Fargate, Anda tidak perlu mengelola server, menangani perencanaan kapasitas, atau mengisolasi beban kerja kontainer untuk keamanan.

Jika Anda memiliki persyaratan berikut, kami sarankan Anda menggunakan instance Fargate:

- Pekerjaan Anda harus dimulai dengan cepat, khususnya kurang dari 30 detik.
- Persyaratan pekerjaan Anda adalah 16 vCPU atau kurang, tidak ada GPU, dan 120 GiB memori atau kurang.

Untuk informasi selengkapnya, lihat [Kapan saatnya menggunakan Fargate](#).

Jika Anda memiliki persyaratan berikut, kami sarankan Anda menggunakan instans Amazon EC2:

- Anda memerlukan peningkatan kontrol atas pemilihan instans atau memerlukan penggunaan jenis instans tertentu.
- Pekerjaan Anda memerlukan sumber daya yang tidak AWS Fargate dapat disediakan, seperti GPU, lebih banyak memori, AMI khusus, atau Amazon Elastic Fabric Adapter.
- Anda membutuhkan tingkat throughput atau konkurensi yang tinggi.

- Anda perlu menyesuaikan AMI, Template Peluncuran Amazon EC2, atau akses ke parameter Linux khusus.

Dengan Amazon EC2, Anda dapat menyesuaikan beban kerja dengan lebih baik dengan persyaratan spesifik Anda dan berjalan sesuai skala jika diperlukan.

Amazon EC2 Sesuai Permintaan atau Amazon EC2 Spot

Sebagian besar AWS Batch pelanggan menggunakan instans Amazon EC2 Spot karena penghematan atas instans On-Demand. Namun, jika beban kerja Anda berjalan selama beberapa jam dan tidak dapat terganggu, instans On-Demand mungkin lebih cocok untuk Anda. Anda selalu dapat mencoba instans Spot terlebih dahulu dan beralih ke On-Demand jika perlu.

Jika Anda memiliki persyaratan dan harapan berikut, gunakan instans On-Demand Amazon EC2:

- Runtime pekerjaan Anda lebih dari satu jam, dan Anda tidak dapat mentolerir gangguan pada beban kerja Anda.
- Anda memiliki SLO (tujuan tingkat layanan) yang ketat untuk keseluruhan beban kerja Anda dan tidak dapat meningkatkan waktu komputasi.
- Contoh yang Anda butuhkan lebih mungkin untuk melihat gangguan.

Jika Anda memiliki persyaratan dan harapan berikut, gunakan instans Amazon EC2 Spot:

- Runtime untuk pekerjaan Anda biasanya 30 menit atau kurang.
- Anda dapat mentolerir potensi gangguan dan penjadwalan ulang pekerjaan sebagai bagian dari beban kerja Anda. Untuk informasi selengkapnya, lihat [Penasihat Instans Spot](#).
- Pekerjaan yang berjalan lama dapat dimulai ulang dari pos pemeriksaan jika terganggu.

Anda dapat mencampur kedua model pembelian dengan mengirimkan instans Spot terlebih dahulu dan kemudian menggunakan instance On-Demand sebagai opsi fallback. Misalnya, kirimkan pekerjaan Anda pada antrian yang terhubung ke lingkungan komputasi yang berjalan di instans Amazon EC2 Spot. Jika pekerjaan terputus, catch event dari Amazon EventBridge dan kaitkan dengan reklamasi instance Spot. Kemudian, kirim ulang pekerjaan ke antrian On-Demand menggunakan fungsi atau AWS Lambda. AWS Step Functions Untuk informasi selengkapnya [Tutorial: Mengirim Pemberitahuan Amazon Simple Notification Service untuk Peristiwa Tugas Gagal](#), lihat

[Praktik terbaik untuk menangani interupsi Instans Spot Amazon EC2 dan Kelola dengan Step AWS Batch Functions.](#)

Important

Gunakan berbagai jenis instans, ukuran, dan Availability Zone untuk lingkungan komputasi Sesuai Permintaan untuk menjaga ketersediaan kumpulan instans Amazon EC2 Spot dan mengurangi tingkat interupsi.

Gunakan praktik terbaik Amazon EC2 Spot untuk AWS Batch

Saat memilih instans Spot Amazon Elastic Compute Cloud (EC2), Anda mungkin dapat mengoptimalkan alur kerja untuk menghemat biaya, terkadang secara signifikan. Untuk informasi selengkapnya, lihat [Praktik terbaik untuk Amazon EC2 Spot](#).

Untuk mengoptimalkan alur kerja Anda guna menghemat biaya, pertimbangkan praktik terbaik Amazon EC2 Spot berikut untuk: AWS Batch

- Pilih strategi **SPOT_CAPACITY_OPTIMIZED** alokasi — AWS Batch pilih instans Amazon EC2 dari kumpulan kapasitas Amazon EC2 Spot terdalam. Jika Anda khawatir tentang gangguan, ini adalah pilihan yang cocok. Untuk informasi selengkapnya, lihat [Strategi alokasi](#).
- Diversifikasi jenis instans — Untuk mendiversifikasi jenis instans Anda, pertimbangkan ukuran dan keluarga yang kompatibel, lalu AWS Batch pilih berdasarkan harga atau ketersediaan. Misalnya, pertimbangkan `c5.24xlarge` sebagai alternatif untuk `c5.12xlarge` atau `c5a`, `c5nc5d`, `m5`, dan `m5d` keluarga. Untuk informasi selengkapnya, lihat [Fleksibel tentang jenis instans dan Availability Zone](#).
- Kurangi waktu proses atau pos pemeriksaan pekerjaan — Kami menyarankan agar tidak menjalankan pekerjaan yang memakan waktu satu jam atau lebih saat menggunakan instans Spot Amazon EC2 untuk menghindari gangguan. Jika Anda membagi atau memeriksa pekerjaan Anda menjadi bagian-bagian yang lebih kecil yang terdiri dari 30 menit atau kurang, Anda dapat secara signifikan mengurangi kemungkinan interupsi.
- Gunakan percobaan ulang otomatis — Untuk menghindari gangguan pada AWS Batch pekerjaan, atur percobaan ulang otomatis untuk pekerjaan. Pekerjaan batch dapat terganggu karena salah satu alasan berikut: kode keluar bukan nol dikembalikan, kesalahan layanan terjadi, atau reklamasi instance terjadi. Anda dapat mengatur hingga 10 percobaan ulang otomatis. Sebagai permulaan,

kami menyarankan Anda mengatur setidaknya 1-3 percobaan ulang otomatis. [Untuk informasi tentang melacak interupsi Spot Amazon EC2, lihat Dasbor Gangguan Spot.](#)

Untuk AWS Batch, jika Anda mengatur parameter coba lagi, pekerjaan ditempatkan di depan antrian pekerjaan. Artinya, pekerjaan diberi prioritas. Saat Anda membuat definisi pekerjaan atau Anda mengirimkan pekerjaan di AWS CLI, Anda dapat mengonfigurasi strategi coba lagi. Untuk informasi selengkapnya, lihat [submit-job](#).

```
$ aws batch submit-job --job-name MyJob \
  --job-queue MyJQ \
  --job-definition MyJD \
  --retry-strategy attempts=2
```

- Gunakan percobaan ulang khusus — Anda dapat mengonfigurasi strategi coba ulang pekerjaan ke kode keluar aplikasi tertentu atau reklamasi instance. Dalam contoh berikut, jika tuan rumah menyebabkan kegagalan, pekerjaan dapat dicoba ulang hingga lima kali. Namun, jika pekerjaan gagal karena alasan yang berbeda, pekerjaan keluar dan status diatur ke FAILED.

```
"retryStrategy": {
  "attempts": 5,
  "evaluateOnExit":
  [{
    "onStatusReason" : "Host EC2*",
    "action": "RETRY"
  }, {
    "onReason" : "*"
    "action": "EXIT"
  }]
}
```

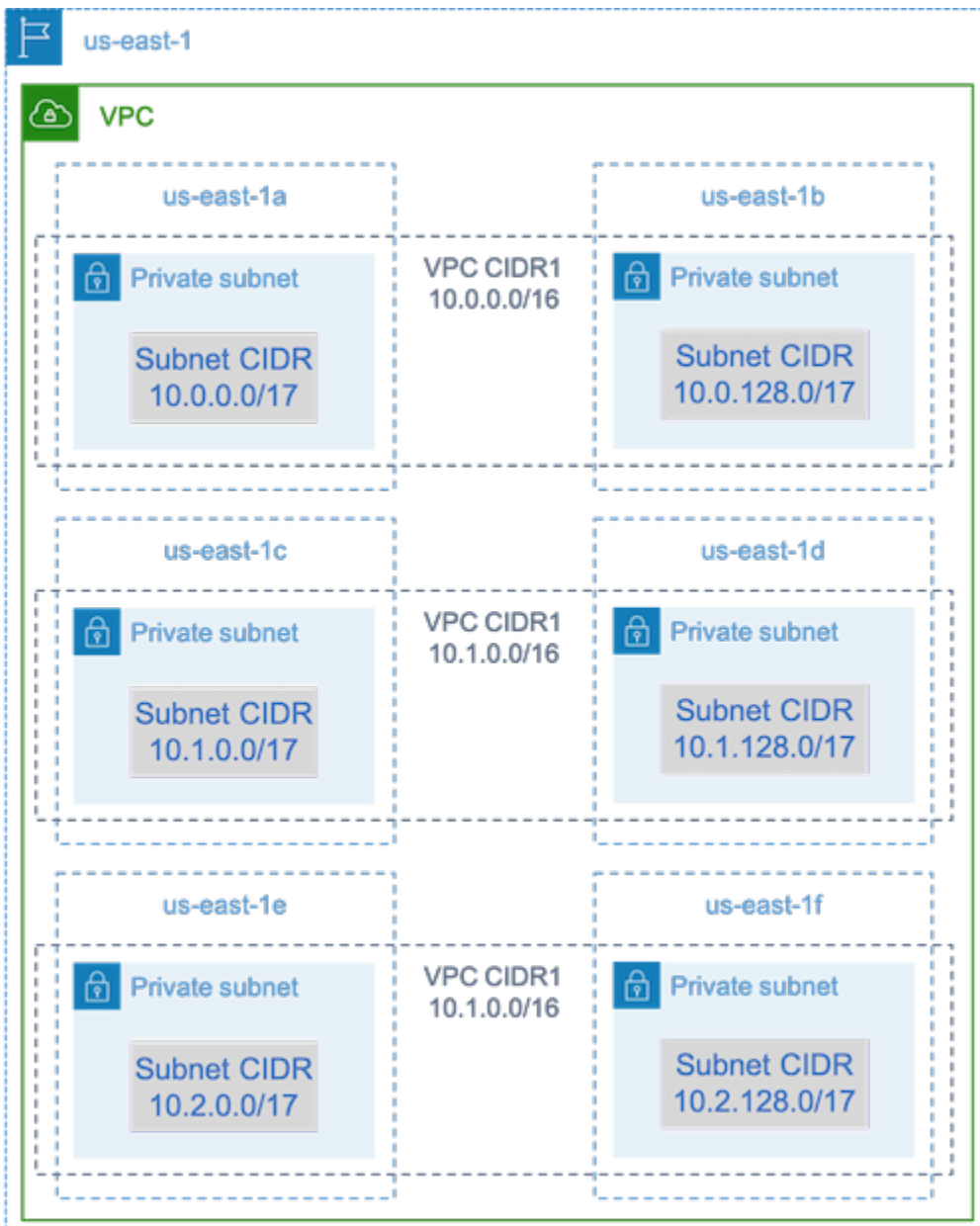
- Gunakan Dasbor Gangguan Spot — Anda dapat menggunakan Dasbor Gangguan Spot untuk melacak interupsi Spot. Aplikasi ini menyediakan metrik pada instans Spot Amazon EC2 yang direklamasi dan Availability Zone tempat instans Spot berada. Untuk informasi selengkapnya, lihat [Dasbor Gangguan Spot](#)

Kesalahan umum dan pemecahan masalah

Kesalahan AWS Batch sering terjadi pada tingkat aplikasi atau disebabkan oleh konfigurasi instans yang tidak memenuhi persyaratan pekerjaan spesifik Anda. Masalah lain termasuk pekerjaan yang macet dalam RUNNABLE status atau lingkungan komputasi yang macet dalam suatu INVALID

keadaan. Untuk informasi selengkapnya tentang pemecahan masalah pekerjaan yang macet dalam `RUNNABLE` status, lihat. [Pekerjaan terjebak dalam `RUNNABLE` status](#) Untuk informasi tentang pemecahan masalah lingkungan komputasi dalam suatu `INVALID` status, lihat. [`INVALID` lingkungan komputasi](#)

- Periksa kuota vCPU Amazon EC2 Spot — Verifikasi bahwa kuota layanan Anda saat ini memenuhi persyaratan pekerjaan. Misalnya, misalkan kuota layanan Anda saat ini adalah 256 vCPU dan pekerjaan membutuhkan 10.000 vCPU. Kemudian, kuota layanan tidak memenuhi persyaratan pekerjaan. Untuk informasi selengkapnya dan petunjuk pemecahan masalah, lihat kuota [layanan Amazon EC2 dan Bagaimana cara meningkatkan kuota layanan Amazon EC2Resources saya?](#) .
- Pekerjaan gagal sebelum aplikasi berjalan — Beberapa pekerjaan mungkin gagal karena `DockerTimeoutError` kesalahan atau `CannotPullContainerError` kesalahan. Untuk informasi pemecahan masalah, lihat [Bagaimana cara mengatasi kesalahan "`DockerTimeoutError`" di? AWS Batch](#) .
- Alamat IP tidak mencukupi — Jumlah alamat IP di VPC dan subnet Anda dapat membatasi jumlah instance yang dapat Anda buat. Gunakan Classless Inter-Domain Routing (CIDR) untuk memberikan lebih banyak alamat IP daripada yang diperlukan untuk menjalankan beban kerja Anda. Jika perlu, Anda juga dapat membuat VPC khusus dengan ruang alamat yang besar. Misalnya, Anda dapat membuat VPC dengan beberapa CIDR `10.x.0.0/16` dan subnet di setiap Availability Zone dengan CIDR. `10.x.y.0/17` Dalam contoh ini, x adalah antara 1-4 dan y adalah 0 atau 128. Konfigurasi ini menyediakan 36.000 alamat IP di setiap subnet.



- Pastikan instans terdaftar di Amazon EC2 — Jika Anda melihat instans di konsol Amazon EC2, tetapi tidak ada instans penampung Amazon Elastic Container Service di kluster Amazon ECS Anda, agen Amazon ECS mungkin tidak diinstal pada Amazon Machine Image (AMI). Agen Amazon ECS, Data Amazon EC2 di AMI Anda, atau templat peluncuran mungkin juga tidak dikonfigurasi dengan benar. Untuk mengisolasi akar penyebab, buat instans Amazon EC2 terpisah atau sambungkan ke instance yang ada menggunakan SSH. Untuk informasi selengkapnya, lihat [konfigurasi agen penampung Amazon ECS](#), [Lokasi File Log Amazon ECS](#), dan [AMI sumber daya komputasi](#)

- Tinjau AWS Dasbor — Tinjau AWS Dasbor untuk memverifikasi bahwa status pekerjaan yang diharapkan dan bahwa skala lingkungan komputasi seperti yang diharapkan. Anda juga dapat meninjau log pekerjaan CloudWatch.
- Verifikasi bahwa instance Anda dibuat — Jika instance dibuat, itu berarti lingkungan komputasi Anda diskalakan seperti yang diharapkan. Jika instance Anda tidak dibuat, cari subnet terkait di lingkungan komputasi Anda untuk diubah. Untuk selengkapnya, lihat [Memverifikasi aktivitas penskalaan untuk grup Auto Scaling](#).

Kami juga menyarankan Anda memverifikasi bahwa instans Anda dapat memenuhi persyaratan pekerjaan terkait Anda. Misalnya, pekerjaan mungkin memerlukan 1 TiB memori, tetapi lingkungan komputasi menggunakan jenis instance C5 yang terbatas pada memori 192 GB.

- Verifikasi bahwa instans Anda diminta oleh AWS Batch — Periksa riwayat grup Auto Scaling untuk memverifikasi bahwa instans Anda diminta oleh AWS Batch. Ini adalah indikasi bagaimana Amazon EC2 mencoba memperoleh instance. Jika Anda menerima kesalahan yang menyatakan Spot Amazon EC2 tidak dapat memperoleh instance di Availability Zone tertentu, ini mungkin karena Availability Zone tidak menawarkan keluarga instance tertentu.
- Verifikasi bahwa instans terdaftar dengan Amazon ECS — Jika Anda melihat instans di konsol Amazon EC2, tetapi tidak ada instans penampung Amazon ECS di kluster Amazon ECS Anda, agen Amazon ECS mungkin tidak diinstal pada Amazon Machine Image (AMI). Selain itu, Agen Amazon ECS, Data Amazon EC2 di AMI Anda, atau templat peluncuran mungkin tidak dikonfigurasi dengan benar. Untuk mengisolasi akar penyebab, buat instans Amazon EC2 terpisah atau sambungkan ke instance yang ada menggunakan SSH. Untuk informasi selengkapnya, lihat [file konfigurasi CloudWatch agen: bagian Log](#), [Lokasi File Log Amazon ECS](#), dan [AMI sumber daya komputasi](#).
- Buka tiket dukungan — Jika Anda masih mengalami masalah setelah beberapa pemecahan masalah dan memiliki rencana dukungan, buka tiket dukungan. Di tiket dukungan, pastikan untuk menyertakan informasi tentang masalah, spesifikasi beban kerja, konfigurasi, dan hasil pengujian. Untuk informasi selengkapnya, lihat [Bandingkan AWS Support Paket](#).
- Tinjau forum AWS Batch dan HPC — Untuk informasi lebih lanjut, lihat forum [AWS Batch](#) dan [HPC](#).
- Tinjau Dasbor Pemantauan AWS Batch Waktu Proses — Dasbor ini menggunakan arsitektur tanpa server untuk menangkap peristiwa dari Amazon ECS, dan AWS Batch Amazon EC2 untuk memberikan wawasan tentang pekerjaan dan instans. Untuk informasi selengkapnya, lihat [Solusi Dasbor Pemantauan AWS Batch Runtime](#).

Riwayat dokumen

Tabel berikut menjelaskan perubahan penting pada dokumentasi sejak rilis awal AWS Batch. Kami juga rutin memperbarui dokumentasi untuk menjawab umpan balik yang Anda kirimkan kepada kami.

Perubahan	Deskripsi	Tanggal
Diperbarui versi Amazon EKS yang AWS Batch didukung	Memperbarui versi Amazon EKS yang AWS Batch mendukung untuk menghapus versi 1.22.	Maret 11, 2024
Diperbarui versi Amazon EKS yang AWS Batch didukung	Memperbarui versi Amazon EKS yang AWS Batch mendukung untuk menyertakan versi 1.29.	Februari 29, 2024
Percobaan ulang pekerjaan otomatis	Mengoreksi sampel kode.	Februari 29, 2024
Menambahkan dukungan untuk pekerjaan multi-kontainer untuk AWS Batch	Menambahkan dukungan untuk pekerjaan multi-kontainer AWS Batch untuk Amazon Elastic Container Service, Amazon Elastic Kubernetes Service, dan AWS Fargate	Februari 28, 2024
Diperbarui versi Amazon EKS yang AWS Batch didukung	Memperbarui versi Amazon EKS yang AWS Batch mendukung untuk menyertakan versi 1.28	Januari 27, 2024
Diperbarui BatchServiceRolePolicy dan AWSServiceRole	BatchServiceRolePolicy Diperbarui untuk menambahkan dukungan	5 Desember 2023

untuk menjelaskan riwayat dan Amazon EC2 Auto Scaling aktivitas permintaan Armada Spot.

AWSBatchServiceRole

Diperbarui untuk menambahkan ID pernyataan, memberikan AWS Batch izin ke `ec2:DescribeSpotFleetRequestHistory` dan `autoscaling:DescribeScalingActivities`.

[AWS Batch di Amazon EKS](#)

AWS Batch menambahkan dukungan untuk menjalankan pekerjaan di kluster Amazon EKS.

25 Oktober 2022

[Pencegahan deputi kebingungan lintas layanan untuk AWS Batch](#)

AWS Batch sekarang menyediakan solusi untuk masalah keamanan wakil yang membingungkan, yang muncul ketika entitas (layanan atau akun) dipaksa oleh entitas yang berbeda untuk melakukan suatu tindakan.

6 Juni 2022

Antarmuka VPC Endpoint ()AWS PrivateLink	Menambahkan dukungan untuk mengonfigurasi titik akhir VPC antarmuka yang didukung oleh. AWS PrivateLink Ini berarti Anda dapat membuat koneksi pribadi antara VPC Anda dan AWS Batch tanpa memerlukan akses melalui instans NAT, koneksi VPN, atau. AWS Direct Connect	15 April 2022
Pembaruan lingkungan komputasi yang disempurnakan	AWS Batch pembaruan dukungan yang disempurnakan untuk menghitung lingkungan.	April 14, 2022
AWS pembaruan kebijakan terkelola - Pembaruan ke kebijakan yang ada	AWS Batch memperbarui kebijakan terkelola yang ada.	Desember 6, 2021
Penjadwalan saham yang adil	AWS Batch menambahkan dukungan untuk menambahkan kebijakan penjadwalan ke antrian pekerjaan.	November 9, 2021
Amazon EFS	AWS Batch menambahkan dukungan untuk menambahkan sistem file Amazon EFS ke definisi pekerjaan Anda.	Selasa, 01 April 2021
Menambahkan peran terkait layanan	AWS Batch menambahkan peran AWSServiceRoleForBatchterkait layanan.	10 Maret 2021

AWS Fargate dukungan	AWS Batch menambahkan dukungan untuk menjalankan pekerjaan pada sumber daya Fargate.	3 Desember 2020
Dukungan Amazon Linux 2	AWS Batch menambahkan dukungan untuk pemilihan otomatis Amazon Linux 2 AMI di lingkungan komputasi menggunakan parameter Konfigurasi EC2.	24 November 2020
Strategi coba lagi yang ditingkatkan	AWS Batch meningkatkan strategi coba lagi untuk pekerjaan. Sekarang tugas dapat dicoba lagi atau menghentikan percobaan kembali lebih lanjut dengan mencocokkan <code>ExitCode</code> , <code>Reason</code> , atau <code>StatusReason</code> tugas dengan pola.	20 Oktober 2020
Penandaan sumber daya	AWS Batch menambahkan dukungan untuk menambahkan tag metadata ke lingkungan komputasi, definisi pekerjaan, antrian pekerjaan, dan pekerjaan Anda.	Selasa, 07 Oktober 2020
Rahasia	AWS Batch menambahkan dukungan untuk menyampaikan rahasia ke pekerjaan.	1 Oktober 2020
Pencatatan log	AWS Batch menambahkan dukungan untuk menentukan driver log tambahan untuk pekerjaan.	1 Oktober 2020

Strategi alokasi	AWS Batch menambahkan dukungan untuk beberapa strategi untuk memilih jenis instance.	16 Oktober 2019
Dukungan EFA	AWS Batch menambahkan dukungan untuk perangkat Elastic Fabric Adapter (EFA).	Selasa, 02 Agustus 2019
Penjadwalan GPU	AWS Batch menambahkan penjadwalan GPU. Dengan fitur ini, Anda dapat menentukan jumlah GPU yang dibutuhkan setiap tugas, dan AWS Batch akan meningkatkan skala instans sesuai kebutuhan.	4 April 2019
Pekerjaan paralel multi-node	AWS Batch menambahkan dukungan untuk pekerjaan paralel multi-node. Anda dapat menggunakan fitur ini untuk menjalankan satu tugas yang mencakup beberapa instans Amazon EC2.	19 November 2018
Izin tingkat sumber daya	AWS Batch mendukung izin tingkat sumber daya pada beberapa operasi API.	12 November 2018
Dukungan template Peluncuran Amazon EC2	AWS Batch menambahkan dukungan untuk menggunakan template peluncuran dengan lingkungan komputasi.	12 November 2018

AWS Batch batas waktu kerja	AWS Batch menambahkan dukungan untuk batas waktu kerja. Dengan dukungan ini, Anda dapat mengonfigurasi durasi batas waktu tertentu untuk pekerjaan Anda sehingga jika pekerjaan berjalan lebih lama dari yang seharusnya, AWS Batch mengakhiri pekerjaan.	5 April 2018
AWS Batch pekerjaan sebagai EventBridge target	AWS Batch pekerjaan tersedia sebagai EventBridge target. Dengan membuat aturan sederhana, Anda dapat mencocokkan peristiwa dan mengirimkan tugas AWS Batch untuk menanggapi.	Selasa, 01 Maret 2018
CloudTrail audit untuk AWS Batch	CloudTrail dapat mengaudit panggilan yang dilakukan ke tindakan AWS Batch API.	10 Januari 2018
Lowongan kerja Array	AWS Batch menambahkan dukungan untuk pekerjaan array. Anda dapat menggunakan tugas array untuk sapuan parameter dan beban kerja Monte Carlo.	28 November 2017

<u>AWS Batch Penandaan yang diperluas</u>	AWS Batch memperluas dukungan untuk fungsi penandaan. Anda dapat menggunakan fungsi ini untuk menentukan tanda untuk Instans Spot Amazon EC2 yang diluncurkan dalam lingkungan komputasi terkelola.	26 Oktober 2017
<u>AWS Batch aliran acara untuk EventBridge</u>	AWS Batch menambahkan aliran acara untuk EventBridge. Anda dapat menggunakan streaming AWS Batch acara untuk menerima pemberitahuan mendekati waktu nyata mengenai status pekerjaan yang dikirimkan ke antrian pekerjaan Anda.	24 Oktober 2017
<u>Percobaan ulang pekerjaan otomatis</u>	AWS Batch menambahkan dukungan untuk percobaan ulang pekerjaan. Dengan pembaruan ini, Anda dapat menerapkan strategi percobaan kembali untuk tugas dan ketentuan tugas Anda yang memungkinkan tugas untuk secara otomatis dicoba lagi jika gagal.	28 Maret 2017
<u>AWS Batch ketersediaan umum</u>	AWS Batch diperkenalkan, dirancang sebagai sarana bagi Anda untuk menjalankan beban kerja komputasi batch di AWS Cloud.	5 Januari 2017

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