



Panduan Pengguna

EC2 Image Builder



EC2 Image Builder: Panduan Pengguna

Copyright © 2023 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 kekayaan masing-masing pemiliknya, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau mungkin tidak.

Table of Contents

Apa itu EC2 Image Builder?	1
Fitur EC2 Image Builder	2
Sistem operasi yang didukung	3
Format gambar yang didukung	3
Konsep	4
Penetapan harga	7
Terkait Layanan AWS	7
Cara kerja EC2 Image Builder	10
Elemen AMI	11
Kuota default	12
AWSWilayah dan Titik Akhir	12
Manajemen komponen	12
Pengujian gambar	12
Versi semantik	13
Sumber daya dibuat	14
Distribusi	14
Berbagi Sumber Daya	15
Kepatuhan	15
Mulai	16
Prasyarat	16
Peran terkait layanan EC2 Image Builder	16
Persyaratan konfigurasi	16
Repositori kontainer (pipa gambar kontainer)	17
AWS Identity and Access Management (IAM)	17
Akses EC2 Image Builder	19
Buat pipeline gambar (AMI)	19
Langkah 1: Tentukan detail pipa	20
Langkah 2: Pilih resep	20
Langkah 3: Tentukan konfigurasi infrastruktur - opsional	23
Langkah 4: Tentukan pengaturan distribusi - opsional	24
Langkah 5: Tinjau	24
Langkah 6: Membersihkan	24
Buat pipeline gambar (Docker)	27
Langkah 1: Tentukan detail pipa	27

Langkah 2: Pilih resep	28
Langkah 3: Tentukan konfigurasi infrastruktur - opsional	30
Langkah 4: Tentukan pengaturan distribusi - opsional	31
Langkah 5: Tinjau	31
Langkah 6: Membersihkan	32
AWSTOEmanajer komponen	35
AWSTOE unduh	35
Wilayah yang Didukung	37
Memulai dengan AWSTOE	39
Verifikasi tanda tangan	39
Langkah 1: Instal AWSTOE	45
Langkah 2: Tetapkan AWS kredensial	46
Langkah 3: Kembangkan dokumen komponen secara lokal	47
Langkah 4: Validasi komponen AWSTOE	49
Langkah 5: Jalankan AWSTOE komponen	49
Gunakan dokumen komponen	51
Alur kerja dokumen komponen	51
Pencatatan komponen	53
Rantai input dan output	54
Skema dan definisi dokumen	56
Skema contoh dokumen	60
Tentukan variabel	64
Gunakan konstruksi perulangan	71
Modul aksi	82
Eksekusi umum	83
Unduh dan unggah file	98
Operasi sistem file	113
Tindakan instalasi perangkat lunak	154
Tindakan sistem	179
Konfigurasi masukan	185
Komponen terkelola paket distributor untuk Windows	189
Prasyarat	190
Konfigurasi izin Distributor Systems Manager	190
Konfigurasi distributor-package-windows sebagai komponen mandiri	193
Konfigurasi aws-vss-components-windows sebagai komponen mandiri	193
Temukan paket Distributor	194

Komponen pengerasan CIS	194
Komponen pengerasan STIG	195
Komponen pengerasan STIG Windows	196
Log riwayat versi STIG untuk Windows	204
Komponen pengerasan STIG Linux	209
Log riwayat versi STIG untuk Linux	215
Komponen validator kepatuhan SCAP	220
Referensi perintah	224
run	224
validasi	228
Kelola sumber daya	230
Komponen-komponen	231
Membuat dokumen komponen YAMAL	233
Parameter komponen	236
Daftar dan lihat komponen	241
Buat komponen (konsol)	245
Buat komponen dengan AWS CLI	246
Impor komponen (AWS CLI)	252
Pembersihan sumber daya	252
Resep	252
Daftar dan lihat resep gambar	253
Daftar dan lihat resep wadah	255
Buat versi baru dari resep gambar	257
Buat versi baru resep wadah	269
Pembersihan sumber daya	277
Citra	277
Buat daftar gambar dan versi build	278
Lihat detail gambar	289
Buat gambar	297
Impor gambar VM	300
Kelola temuan keamanan	305
Pembersihan sumber daya	309
Konfigurasi infrastruktur	310
Daftar dan lihat konfigurasi infrastruktur	311
Membuat konfigurasi infrastruktur	312
Memperbarui konfigurasi infrastruktur	314

Titik akhir VPC (AWS PrivateLink)	317
Pengaturan distribusi	321
Daftar dan lihat pengaturan distribusi	323
Buat dan perbarui distribusi AMI	325
Membuat dan memperbarui distribusi gambar kontainer	337
Mengatur distribusi AMI lintas akun	340
Tentukan template peluncuran AMI	347
Kelola siklus hidup gambar	350
Prasyarat	352
Kebijakan siklus hidup	355
Cara kerja aturan siklus hidup	367
Alur kerja gambar	370
Daftar alur kerja gambar	371
Buat alur kerja gambar	375
Membuat dokumen alur kerja YAMB	378
Impor dan ekspor gambar VM	417
Impor VM ke Image Builder () AWS CLI	417
Mendistribusikan disk VM dari image build () Anda AWS CLI	420
Bagikan sumber daya	420
Bekerja dengan sumber daya bersama	421
Prasyarat untuk berbagi komponen, gambar, dan resep	421
Layanan-layanan terkait	422
Berbagi di seluruh Wilayah	422
Berbagi komponen, gambar, atau resep	422
Membatalkan berbagi komponen, gambar, atau resep bersama	426
Mengidentifikasi komponen, gambar, atau resep bersama	426
Izin komponen, gambar, dan resep bersama	427
Tagihan dan pengukuran	428
Batas sumber daya	428
Memberi tanda pada sumber daya	428
Menandai sumber daya (AWS CLI)	428
Membatalkan tag sumber daya () AWS CLI	429
Daftar semua tag untuk sumber daya tertentu (AWS CLI)	429
Hapus sumber daya	430
Hapus sumber daya (konsol)	430
Hapus sumber daya (AWS CLI)	432

Kelola saluran pipa	434
Daftar dan lihat saluran pipa	435
Daftar pipa gambar () AWS CLI	435
Dapatkan detail pipa gambar (AWS CLI)	435
Membuat dan memperbarui pipeline (AMI)	435
Buat pipeline AMI (AWS CLI)	436
Perbarui pipa (konsol)	438
Perbarui pipa (AWS CLI)	442
Buat dan perbarui pipeline (kontainer)	443
Buat pipeline (AWS CLI)	444
Perbarui pipa (konsol)	446
Perbarui pipa (AWS CLI)	449
Konfigurasi alur kerja gambar	451
Tentukan grup uji untuk alur kerja pengujian	452
Mengatur parameter alur kerja di pipeline Image Builder (konsol)	453
Tentukan peran layanan IAM yang digunakan Image Builder untuk menjalankan tindakan alur kerja	453
Jalankan saluran pipa	454
Gunakan ekspresi cron	454
Nilai yang didukung untuk ekspresi cron di Image Builder	455
Contoh ekspresi cron di EC2 Image Builder	457
Ekspresi rate	459
Gunakan EventBridge aturan	460
EventBridge istilah	460
Lihat EventBridge aturan untuk pipeline Image Builder	461
Gunakan EventBridge aturan untuk menjadwalkan pembuatan pipeline	462
Integrasikan produk dan layanan	464
AWS CloudTrail	466
CloudWatch Log Amazon	466
Amazon EventBridge	467
Amazon Inspector	468
AWS Marketplace	470
Fitur integrasi AWS Marketplace	470
Temukan produk AWS Marketplace gambar dari konsol Image Builder	471
Gunakan produk AWS Marketplace gambar dalam resep Image Builder	474
Layanan Pemberitahuan Sederhana Amazon	475

Topik SNS Terenkripsi	475
Format pesan SNS	477
Produk kepatuhan	483
Memantau	485
CloudTrail log	485
Informasi Image Builder di CloudTrail	485
Keamanan di EC2 Image Builder	487
Perlindungan data	488
Enkripsi dan Manajemen Kunci	489
Penyimpanan data	495
Privasi Lalu Lintas Antar Jaringan	495
Pengelolaan Identitas dan Akses	495
Audiens	495
Mengautentikasi dengan identitas	496
Bagaimana EC2 Image Builder bekerja dengan IAM	496
Kebijakan Berbasis Identitas	508
Kebijakan Berbasis Sumber Daya	511
Kebijakan terkelola	512
Peran tertaut layanan	542
Pemecahan Masalah	544
Validasi kepatuhan	546
Ketahanan	547
Keamanan infrastruktur	548
Manajemen tambalan	549
Praktik terbaik	550
Diperlukan pembersihan pasca-build	551
Ganti skrip pembersihan Linux	556
Memecahkan Masalah Image Builder	560
Memecahkan masalah pembangunan pipa	560
Skenario pemecahan masalah	562
Riwayat dokumen	568
.....	dlxxviii

Apa itu EC2 Image Builder?

EC2 Image Builder adalah Layanan AWS pengelola penuh yang membantu Anda mengotomatiskan pembuatan, pengelolaan, dan penyebaran gambar yang disesuaikan, aman, up-to-date dan server. Anda dapat menggunakan AWS Management Console, AWS Command Line Interface, atau API untuk membuat gambar kustom di Akun AWS.

Anda memiliki gambar yang disesuaikan yang dibuat Image Builder di akun Anda. Anda dapat mengonfigurasi saluran pipa untuk mengotomatiskan pembaruan dan penambalan sistem untuk gambar yang Anda miliki. Anda juga dapat menjalankan perintah yang berdiri sendiri untuk membuat gambar dengan sumber daya konfigurasi yang telah Anda tentukan.

Wizard pipeline Image Builder dapat memandu Anda melalui langkah-langkah untuk membuat gambar kustom, sebagai berikut:

1. Pilih gambar dasar untuk kustomisasi Anda.
2. Tambahkan atau hapus perangkat lunak dari gambar dasar Anda.
3. Sesuaikan pengaturan dan skrip dengan komponen build.
4. Jalankan tes yang dipilih atau buat komponen uji kustom.
5. Bagikan AMI ke Wilayah AWS dan Akun AWS.
6. Jika pipeline Image Builder membuat Amazon Machine Image (AMI) khusus untuk distribusi, Anda dapat mengizinkan organisasi Akun AWS, organisasi, dan OU lain untuk meluncurkannya dari akun Anda. Akun Anda ditagih untuk biaya yang terkait dengan AMI.

Image Builder terintegrasi dengan hal-hal berikut Layanan AWS untuk memberikan metrik peristiwa, pencatatan, dan pemantauan terperinci. Informasi ini membantu Anda melacak aktivitas, memecahkan masalah pembuatan gambar, dan membuat otomatisasi berdasarkan pemberitahuan peristiwa.

Isi Bagian

- [Fitur EC2 Image Builder](#)
- [Sistem operasi yang didukung](#)
- [Format gambar yang didukung](#)
- [Konsep](#)

- [Penetapan harga](#)
- [Terkait Layanan AWS](#)

Fitur EC2 Image Builder

EC2 Image Builder menyediakan fitur-fitur berikut:

Meningkatkan produktivitas dan mengurangi operasi untuk bangunan yang sesuai dan gambar up-to-date

Image Builder mengurangi jumlah pekerjaan yang terlibat dalam membuat dan mengelola gambar dalam skala besar dengan mengotomatiskan pipeline build Anda. Anda dapat mengotomatiskan build dengan memberikan preferensi jadwal eksekusi build. Otomatisasi mengurangi biaya operasional pemeliharaan perangkat lunak Anda dengan tambalan sistem operasi terbaru.

Tingkatkan uptime layanan

Image Builder menyediakan akses ke komponen pengujian yang dapat Anda gunakan untuk menguji gambar Anda sebelum penerapan. Anda juga dapat membuat komponen uji kustom dengan AWS Task Orchestrator and Executor (AWSTOE), dan menggunakannya. Image Builder mendistribusikan gambar Anda hanya jika semua pengujian yang dikonfigurasi telah berhasil.

Naikkan bilah keamanan untuk penerapan

Image Builder memungkinkan Anda membuat gambar yang menghapus eksposur yang tidak perlu terhadap kerentanan keamanan komponen. Anda dapat menerapkan pengaturan AWS keamanan untuk membuat out-of-the-box gambar aman yang memenuhi kriteria keamanan industri dan internal. Image Builder juga menyediakan koleksi pengaturan untuk perusahaan di industri yang diatur. Anda dapat menggunakan pengaturan ini untuk membantu Anda dengan cepat dan mudah membuat gambar yang sesuai untuk standar STIG. Untuk daftar lengkap komponen STIG yang tersedia melalui Image Builder, lihat [Amazon mengelola komponen pengerasan STIG untuk EC2 Image Builder](#).

Penegakan terpusat dan pelacakan garis keturunan

Menggunakan integrasi bawaan dengan AWS Organizations, Image Builder memungkinkan Anda menerapkan kebijakan yang membatasi akun untuk menjalankan instance hanya dari AMI yang disetujui.

Berbagi sumber daya yang disederhanakan di seluruh Akun AWS

EC2 Image Builder terintegrasi AWS Resource Access Manager dengan AWS RAM () untuk memungkinkan Anda berbagi sumber daya tertentu dengan Akun AWS salah satu atau melalui. AWS Organizations Sumber daya EC2 Image Builder yang dapat dibagikan adalah:

- Komponen-komponen
- Citra
- Tanda terima citra
- tanda terima kontainer

Untuk informasi lebih lanjut, lihat [Bagikan sumber daya EC2 Image Builder](#).

Sistem operasi yang didukung

Image Builder mendukung versi sistem operasi berikut:

Sistem operasi/distribusi	Versi yang didukung
Amazon Linux	2 dan 2023
CentOS	7 dan 8
CentOS Aliran	8
Red Hat Enterprise Linux (RHEL)	7 dan 8
SUSE Linux Server Perusahaan (SUSE)	12 dan 15
Ubuntu	18.04 LTS, 20.04 LTS, dan 22.04 LTS
Windows Server	2012 R2, 2016, 2019, dan 2022

Format gambar yang didukung

Untuk gambar AMI kustom Anda, Anda dapat memilih AMI yang ada sebagai titik awal. Untuk gambar kontainer Docker, Anda dapat memilih dari gambar publik yang dihosting DockerHub, gambar kontainer yang ada di Amazon ECR, atau gambar kontainer yang dikelola Amazon.

Konsep

Istilah dan konsep berikut sangat penting untuk pemahaman Anda dan penggunaan EC2 Image Builder.

AMI

Amazon Machine Image (AMI) adalah unit dasar penyebaran di Amazon EC2, dan merupakan salah satu jenis gambar yang dapat Anda buat dengan Image Builder. AMI adalah gambar mesin virtual pra-konfigurasi yang berisi sistem operasi (OS) dan perangkat lunak prainstal untuk menyebarkan instans EC2. Untuk informasi selengkapnya, lihat [Gambar Mesin Amazon \(AMI\)](#).

Pipa gambar

Pipeline gambar menyediakan kerangka kerja otomatisasi untuk membangun AMI dan gambar kontainer yang aman AWS. Pipeline image Builder Image dikaitkan dengan resep gambar atau resep kontainer yang mendefinisikan fase build, validasi, dan pengujian untuk siklus hidup build image.

Alur citra dapat dikaitkan dengan konfigurasi infrastruktur yang menentukan tempat citra Anda dibangun. Anda dapat menentukan atribut, seperti tipe instans, subnet, grup keamanan, pencatatan, dan konfigurasi terkait infrastruktur lainnya. Anda juga dapat mengaitkan alur citra Anda dengan konfigurasi distribusi untuk menentukan cara Anda ingin men-deploy citra Anda.

Gambar terkelola

Gambar terkelola adalah sumber daya di Image Builder yang terdiri dari AMI atau image kontainer, ditambah metadata, seperti versi dan platform. Image terkelola digunakan oleh pipeline Image Builder untuk menentukan image dasar mana yang akan digunakan untuk build. Dalam panduan ini, gambar yang dikelola kadang-kadang disebut sebagai “gambar,” namun, gambar tidak sama dengan AMI.

Resep gambar

Resep image Builder Image adalah dokumen yang mendefinisikan image dasar dan komponen yang diterapkan pada image dasar untuk menghasilkan konfigurasi yang diinginkan untuk gambar AMI keluaran. Anda dapat menggunakan resep gambar untuk menduplikasi build. Resep gambar Image Builder dapat dibagikan, bercabang, dan diedit menggunakan wizard konsol, API AWS CLI, atau API. Anda dapat menggunakan resep gambar dengan perangkat lunak kontrol versi Anda untuk mempertahankan resep gambar berversi yang dapat dibagikan.

Resep kontainer

Resep kontainer Image Builder adalah dokumen yang mendefinisikan gambar dasar dan komponen yang diterapkan pada gambar dasar untuk menghasilkan konfigurasi yang diinginkan untuk gambar kontainer keluaran. Anda dapat menggunakan resep wadah untuk menduplikasi build. Anda dapat berbagi, membuat cabang, dan mengedit resep gambar Image Builder dengan menggunakan wizard konsolAWS CLI, the, atau API. Anda dapat menggunakan resep kontainer dengan perangkat lunak kontrol versi Anda untuk mempertahankan resep wadah yang dapat dibagikan dan berversi.

Gambar dasar

Gambar dasar adalah gambar dan sistem operasi yang dipilih yang digunakan dalam dokumen resep gambar atau wadah Anda, bersama dengan komponennya. Gambar dasar dan definisi komponen digabungkan menghasilkan konfigurasi yang diinginkan untuk gambar output.

Komponen-komponen

Komponen mendefinisikan urutan langkah yang diperlukan untuk menyesuaikan instance sebelum pembuatan gambar (komponen build), atau untuk menguji instance yang diluncurkan dari gambar yang dibuat (komponen pengujian).

Komponen dibuat dari dokumen deklaratif, teks biasa YANG atau JSON yang menjelaskan konfigurasi runtime untuk membangun dan memvalidasi, atau menguji instance yang dihasilkan oleh pipeline Anda. Komponen berjalan pada instance menggunakan aplikasi manajemen komponen. Aplikasi manajemen komponen mem-parsing dokumen dan menjalankan langkah-langkah yang diinginkan.

Setelah mereka dibuat, satu atau lebih komponen dikelompokkan bersama menggunakan resep gambar atau resep wadah untuk menentukan rencana untuk membangun dan menguji mesin virtual atau gambar kontainer. Anda dapat menggunakan komponen publik yang dimiliki dan dikelola olehAWS, atau Anda dapat membuat sendiri. Untuk informasi selengkapnya tentang komponen, lihat[AWS Task Orchestrator and Executormanajer komponen](#).

Dokumen komponen

Dokumen YANG atau JSON deklaratif, teks biasa yang menjelaskan konfigurasi untuk kustomisasi yang dapat Anda terapkan pada gambar Anda. Dokumen ini digunakan untuk membuat komponen build atau test.

Tahapan runtime

EC2 Image Builder memiliki dua tahap runtime: build dan test. Setiap tahap runtime memiliki satu atau lebih fase dengan konfigurasi yang ditentukan oleh dokumen komponen.

Fase konfigurasi

Daftar berikut menunjukkan fase yang berjalan selama tahap pembuatan dan pengujian:

Membangun panggung:

Tahap membangun

Pipeline image dimulai dengan fase build dari tahap build saat dijalankan. Gambar dasar diunduh, dan konfigurasi yang ditentukan untuk fase pembuatan komponen diterapkan untuk membangun dan meluncurkan instance.

Fase validasi

Setelah Image Builder meluncurkan instance dan menerapkan semua penyesuaian fase build, fase validasi dimulai. Selama fase ini, Image Builder memastikan bahwa semua penyesuaian bekerja seperti yang diharapkan, berdasarkan konfigurasi yang ditentukan komponen untuk fase validasi. Jika validasi instance berhasil, Image Builder menghentikan instance, membuat gambar, dan kemudian melanjutkan ke tahap pengujian.

Tahap uji:

Fase uji

Selama fase ini, Image Builder meluncurkan instance dari gambar yang dibuat setelah fase validasi berhasil diselesaikan. Image Builder menjalankan komponen pengujian selama fase ini untuk memverifikasi bahwa instance sehat dan berfungsi seperti yang diharapkan.

Fase uji host kontainer

Setelah Image Builder menjalankan fase pengujian untuk semua komponen yang Anda pilih dalam resep kontainer, Image Builder menjalankan fase ini untuk alur kerja kontainer. Fase uji host kontainer dapat menjalankan pengujian tambahan yang memvalidasi manajemen kontainer dan konfigurasi runtime kustom.

Alur kerja

Alur kerja menentukan urutan langkah yang dilakukan Image Builder saat membuat gambar baru. Semua gambar memiliki alur kerja build dan test. Kontainer memiliki alur kerja tambahan untuk distribusi.

Jenis alur kerja

BUILD

Meliputi konfigurasi tahap build untuk setiap gambar yang dibuat.

TEST

Meliputi konfigurasi tahap uji untuk setiap gambar yang dibuat.

DISTRIBUTION

Meliputi alur kerja distribusi untuk gambar kontainer.

Penetapan harga

Tidak ada biaya untuk menggunakan EC2 Image Builder untuk membuat AMI atau gambar kontainer khusus. Namun, harga standar berlaku untuk layanan lain yang digunakan dalam proses tersebut. Daftar berikut mencakup penggunaan beberapa Layanan AWS yang dapat menimbulkan biaya saat Anda membuat, membangun, menyimpan, dan mendistribusikan AMI atau gambar kontainer kustom Anda, tergantung pada konfigurasi Anda.

- Meluncurkan instans EC2
- Menyimpan log di Amazon S3
- Memvalidasi gambar dengan Amazon Inspector
- Menyimpan Snapshot Amazon EBS untuk AMI Anda
- Menyimpan gambar kontainer di Amazon ECR
- Mendorong dan menarik gambar kontainer masuk dan keluar dari Amazon ECR
- Jika Tingkat Lanjut Systems Manager diaktifkan, dan instans Amazon EC2 dijalankan dengan aktivasi lokal, Anda mungkin akan dikenakan biaya untuk sumber daya melalui Systems Manager

Terkait Layanan AWS

EC2 Image Builder menggunakan yang Layanan AWS lain untuk membangun gambar. Bergantung pada resep gambar Image Builder atau konfigurasi resep kontainer, layanan berikut mungkin digunakan.

AWS License Manager

AWS License Manager memungkinkan Anda membuat dan menerapkan konfigurasi lisensi dari toko konfigurasi lisensi akun. Untuk setiap AMI, Anda dapat menggunakan Image Builder untuk melampirkan ke konfigurasi lisensi yang sudah ada sebelumnya yang dapat Anda akses sebagai bagian dari alur kerja Image Builder. Konfigurasi lisensi hanya dapat diterapkan ke AMI. Image Builder hanya dapat menggunakan konfigurasi lisensi yang sudah ada sebelumnya dan tidak dapat langsung membuat atau memodifikasi konfigurasi lisensi. Pengaturan License Manager tidak akan mereplikasi seluruh Wilayah AWS yang harus diaktifkan di akun Anda, misalnya, antara Wilayah `ap-east-1` (Asia Pasifik: Hong Kong) dan `me-south-1` (Timur Tengah: Bahrain).

AWS Organizations

AWS Organizations memungkinkan Anda menerapkan Kebijakan Kontrol Layanan (SCP) pada akun di organisasi Anda. Anda dapat membuat, mengelola, mengaktifkan, dan menonaktifkan kebijakan individual. Mirip dengan semua AWS artefak dan layanan lainnya, Image Builder menghormati kebijakan yang ditentukan. AWS Organizations menyediakan SCP template untuk skenario umum, seperti memberlakukan batasan pada akun anggota untuk meluncurkan instance dengan hanya AMI yang disetujui.

Amazon Inspector

Image Builder menggunakan Amazon Inspector sebagai agen pemindaian kerentanan default untuk menetapkan garis dasar keamanan untuk Amazon Linux 2, Windows Server 2012, dan Windows Server 2016. Untuk informasi selengkapnya, lihat [Apa itu Amazon Inspector?](#)

AWS Resource Access Manager

AWS Resource Access Manager (AWS RAM) memungkinkan Anda berbagi sumber daya Anda dengan salah satu Akun AWS atau melalui AWS Organizations. Jika Anda memiliki beberapa Akun AWS, Anda dapat membuat sumber daya secara terpusat dan menggunakannya AWS RAM untuk berbagi sumber daya tersebut dengan akun lain. EC2 Image Builder memungkinkan berbagi untuk sumber daya berikut: komponen, gambar, dan resep gambar. Untuk informasi selengkapnya tentang AWS RAM, lihat [AWS Resource Access Manager Panduan Pengguna](#). Untuk informasi tentang berbagi sumber daya Image Builder, lihat [Bagikan sumber daya EC2 Image Builder](#).

CloudWatch Log Amazon

Anda dapat menggunakan Amazon CloudWatch Logs untuk memantau, menyimpan, dan mengakses file log Anda dari instans EC2, AWS CloudTrail Amazon Route 53, dan sumber lainnya.

Amazon Elastic Container Registry (Amazon ECR)

Amazon ECR adalah layanan registri gambar AWS kontainer terkelola yang aman, terukur, dan andal. Gambar kontainer yang Anda buat dengan Image Builder disimpan di Amazon ECR di Wilayah sumber Anda (tempat build Anda berjalan), dan di Wilayah mana pun tempat Anda mendistribusikan image container. Untuk informasi selengkapnya tentang Amazon ECR, lihat [Panduan Pengguna Amazon Elastic Container Registry](#).

Cara kerja EC2 Image Builder

Saat Anda menggunakan wizard konsol pipeline EC2 Image Builder untuk membuat gambar kustom, wizard memandu Anda melalui langkah-langkah berikut.

1. Tentukan detail pipeline — Masukkan informasi tentang pipeline Anda, seperti nama, deskripsi, tag, dan jadwal untuk menjalankan build otomatis. Anda dapat memilih build manual, jika Anda mau.
2. Pilih resep — Pilih antara membangun AMI, atau membuat gambar kontainer. Untuk kedua jenis gambar keluaran, Anda memasukkan nama dan versi untuk resep Anda, pilih gambar dasar, dan pilih komponen yang akan ditambahkan untuk pembuatan dan pengujian. Anda juga dapat memilih versi otomatis, untuk memastikan bahwa Anda selalu menggunakan versi Sistem Operasi (OS) terbaru yang tersedia untuk gambar dasar Anda. Resep kontainer juga menentukan Dockerfiles, dan repositori Amazon ECR target untuk image container Docker keluaran Anda.

Note

Komponen adalah blok bangunan yang dikonsumsi oleh resep gambar atau resep wadah. Misalnya, paket untuk instalasi, langkah pengerasan keamanan, dan pengujian. Gambar dasar dan komponen yang dipilih membentuk resep gambar.

3. Tentukan konfigurasi infrastruktur - Image Builder meluncurkan instans EC2 di akun Anda untuk menyesuaikan gambar dan menjalankan pengujian validasi. Pengaturan konfigurasi Infrastruktur menentukan detail infrastruktur untuk instans yang akan berjalan di Anda Akun AWS selama proses pembuatan.
4. Tentukan setelan distribusi — Pilih AWS Wilayah untuk mendistribusikan gambar Anda setelah build selesai dan telah lulus semua pengujiannya. Pipeline secara otomatis mendistribusikan gambar Anda ke Wilayah tempat ia menjalankan build, dan Anda dapat menambahkan distribusi gambar untuk Wilayah lain.

Gambar yang Anda buat dari gambar dasar kustom Anda ada di dalam gambar AndaAkun AWS. Anda dapat mengonfigurasi pipeline gambar untuk menghasilkan versi gambar yang diperbarui dan ditambal dengan memasukkan jadwal pembuatan. Ketika build selesai, Anda dapat menerima notifikasi melalui [Amazon Simple Notification Service \(SNS\)](#). Selain menghasilkan gambar akhir, wizard konsol Image Builder menghasilkan resep yang dapat digunakan dengan sistem kontrol versi

yang ada dan pipeline continuous integration/continuous deployment (CI/CD) untuk otomatisasi berulang. Anda dapat berbagi dan membuat versi baru resep Anda.

Isi bagian

- [Elemen AMI](#)
- [Kuota default](#)
- [AWS Wilayah dan Titik Akhir](#)
- [Manajemen komponen](#)
- [Versi semantik](#)
- [Sumber daya dibuat](#)
- [Distribusi](#)
- [Berbagi Sumber Daya](#)
- [Kepatuhan](#)

Elemen AMI

Amazon Machine Image (AMI) adalah gambar mesin virtual (VM) yang telah dikonfigurasi sebelumnya yang berisi OS dan perangkat lunak untuk menyebarkan instans EC2.

AMI mencakup elemen-elemen berikut:

- Template untuk volume root VM. Saat Anda meluncurkan Amazon EC2 VM, volume perangkat root berisi gambar untuk mem-boot instance. Saat penyimpanan instance digunakan, perangkat root adalah volume penyimpanan instance yang dibuat dari template di Amazon S3. Untuk informasi selengkapnya, lihat [Volume Perangkat Root Amazon EC2](#).
- Saat Amazon EBS digunakan, perangkat root adalah volume EBS yang dibuat dari snapshot [EBS](#).
- Luncurkan izin yang menentukan Akun AWS yang dapat meluncurkan VM dengan AMI.
- [Blokir data pemetaan perangkat](#) yang menentukan volume yang akan dilampirkan ke instance setelah peluncuran.
- [Pengidentifikasi sumber daya](#) unik untuk setiap Wilayah, untuk setiap akun.
- Payload [metadata](#) seperti tag, dan properti, seperti Wilayah, sistem operasi, arsitektur, jenis perangkat root, penyedia, izin peluncuran, penyimpanan untuk perangkat root, dan status penandatanganan.

- Tanda tangan AMI untuk gambar Windows untuk melindungi dari gangguan yang tidak sah. Untuk informasi selengkapnya, lihat [Dokumen Identitas Instance](#).

Kuota default

Untuk melihat kuota default Image Builder, lihat [Titik Akhir dan Kuota Image Builder](#).

AWS Wilayah dan Titik Akhir

Untuk melihat titik akhir layanan Image Builder, lihat [Titik Akhir dan Kuota Image Builder](#).

Manajemen komponen

EC2 Image Builder menggunakan AWS Task Orchestrator and Executor aplikasi manajemen komponen AWSTOE () yang membantu Anda mengatur alur kerja yang kompleks, memodifikasi konfigurasi sistem, dan menguji sistem Anda dengan komponen skrip berbasis YAML. Karena AWSTOE merupakan aplikasi mandiri, tidak memerlukan pengaturan tambahan. Ini dapat berjalan di infrastruktur cloud apa pun dan di tempat. Untuk mulai menggunakan AWSTOE sebagai aplikasi mandiri, lihat [Memulai dengan AWSTOE](#).

Image Builder menggunakan AWSTOE untuk melakukan semua aktivitas on-instance. Ini termasuk membangun dan memvalidasi gambar Anda sebelum mengambil snapshot, dan menguji snapshot untuk memastikan bahwa itu berfungsi seperti yang diharapkan sebelum membuat gambar akhir. Untuk informasi selengkapnya tentang cara Image Builder menggunakan AWSTOE untuk mengelola komponennya, lihat [Mengelola komponen dengan Image Builder](#). Untuk informasi selengkapnya tentang membuat komponen dengan AWSTOE, lihat [AWS Task Orchestrator and Executormanajer komponen](#).

Pengujian gambar

Anda dapat menggunakan komponen AWSTOE pengujian untuk memvalidasi gambar Anda, dan memastikan bahwa itu berfungsi seperti yang diharapkan, sebelum membuat gambar akhir.

Umumnya, setiap komponen pengujian terdiri dari dokumen YAMM yang berisi skrip pengujian, biner uji, dan metadata pengujian. Skrip pengujian berisi perintah orkestrasi untuk memulai biner pengujian, yang dapat ditulis dalam bahasa apa pun yang didukung oleh OS. Kode status keluar menunjukkan hasil tes. Metadata pengujian menggambarkan tes dan perilakunya; misalnya, nama, deskripsi, jalur untuk menguji biner, dan durasi yang diharapkan.

Versi semantik

Image Builder menggunakan versi semantik untuk mengatur sumber daya dan memastikan bahwa mereka memiliki ID unik. Versi semantik memiliki empat node:

`<major>. <minor>. <patch>/<build>`

Anda dapat menetapkan nilai untuk tiga yang pertama, dan dapat memfilter semuanya.

Versi semantik disertakan dalam Nama Sumber Daya Amazon (ARN) setiap objek, pada tingkat yang berlaku untuk objek tersebut sebagai berikut:

1. ARN tanpa versi dan ARN Nama tidak menyertakan nilai spesifik di salah satu node. Node ditinggalkan seluruhnya, atau ditentukan sebagai wildcard, misalnya: `x.x.x`.
2. `<major>`Versi ARN hanya memiliki tiga node pertama: `<minor>. <patch>`
3. ARN versi build memiliki keempat node, dan menunjuk ke build tertentu untuk versi objek tertentu.

Penugasan: Untuk tiga node pertama Anda dapat menetapkan nilai integer positif, termasuk nol, dengan batas atas $2^{30}-1$, atau 1073741823 untuk setiap node. Image Builder secara otomatis menetapkan nomor build ke node keempat.

Pola: Anda dapat menggunakan pola numerik apa pun yang mematuhi persyaratan penugasan untuk node yang dapat Anda tetapkan. Misalnya, Anda dapat memilih pola versi perangkat lunak, seperti 1.0.0, atau tanggal, seperti 2021.01.01.

Seleksi: Dengan versi semantik, Anda memiliki fleksibilitas untuk menggunakan wildcard (x) untuk menentukan versi atau node terbaru saat memilih gambar dasar atau komponen untuk resep Anda. Saat Anda menggunakan wildcard di node apa pun, semua node di sebelah kanan wildcard pertama juga harus wildcard.

Misalnya, mengingat versi terbaru berikut: 2.2.4, 1.7.8, dan 1.6.8, pemilihan versi menggunakan wildcard menghasilkan hasil sebagai berikut:

- `x.x.x= 2.2.4`
- `1.x.x= 1.7.8`
- `1.6.x= 1.6.8`
- `x.2.x` tidak valid, dan menghasilkan kesalahan
- `1.x.8` tidak valid, dan menghasilkan kesalahan

Sumber daya dibuat

Saat Anda membuat pipeline, tidak ada sumber daya eksternal Image Builder yang dibuat, kecuali yang berikut ini benar:

- Saat gambar dibuat melalui jadwal pipeline
- Bila Anda memilih Run Pipeline dari menu Actions di konsol Image Builder
- Saat Anda menjalankan salah satu perintah ini dari API atau AWS CLI:
StartImagePipelineExecution atau CreateImage

Sumber daya berikut dibuat selama proses pembuatan gambar:

Pipa gambar AMI

- Instans EC2 (sementara)
- Systems Manager Inventory Association (melalui Systems Manager State Manager jika EnhancedImageMetadata diaktifkan) pada instans EC2
- Amazon EC2 AMI
- Snapshot Amazon EBS yang terkait dengan Amazon EC2 AMI

Pipa gambar kontainer

- Kontainer Docker berjalan pada instance EC2 (sementara)
- Systems Manager Inventory Association (melalui Systems Manager State Manager) EnhancedImageMetadata diaktifkan) pada instans EC2
- Gambar kontainer Docker
- Dockerfile

Setelah gambar dibuat, semua sumber daya sementara dihapus.

Distribusi

EC2 Image Builder dapat mendistribusikan AMI atau gambar kontainer ke Wilayah AWS mana pun. Gambar disalin ke setiap Wilayah yang Anda tentukan di akun yang digunakan untuk membuat gambar.

Untuk gambar keluaran AMI, Anda dapat menentukan izin peluncuran AMI untuk mengontrol mana yang Akun AWS diizinkan untuk meluncurkan instans EC2 dengan AMI yang dibuat. Misalnya, Anda dapat membuat gambar pribadi, publik, atau berbagi dengan akun tertentu. Jika Anda berdua mendistribusikan AMI ke Wilayah lain, dan menentukan izin peluncuran untuk akun lain, izin peluncuran akan disebarluaskan ke AMI di semua Wilayah tempat AMI didistribusikan.

Anda juga dapat menggunakan AWS Organizations akun Anda untuk memberlakukan batasan pada akun anggota untuk meluncurkan instans hanya dengan AMI yang disetujui dan sesuai. Untuk informasi selengkapnya, lihat [Mengelola Akun AWS di Organisasi Anda](#).

Untuk memperbarui setelan distribusi menggunakan konsol Image Builder, ikuti langkah-langkah untuk [Buat versi resep gambar baru \(konsol\)](#), atau [Buat versi resep wadah baru dengan konsol](#).

Berbagi Sumber Daya

Untuk berbagi komponen, resep, atau gambar dengan akun lain atau di dalamnya AWS Organizations, lihat [Bagikan sumber daya EC2 Image Builder](#).

Kepatuhan

Untuk CIS, EC2 Image Builder menggunakan Amazon Inspector untuk melakukan penilaian terhadap eksposur, kerentanan, dan penyimpangan dari praktik terbaik dan standar kepatuhan. Misalnya, Image Builder menilai aksesibilitas jaringan yang tidak diinginkan, CVE yang tidak ditambal, konektivitas internet publik, dan aktivasi login root jarak jauh. Amazon Inspector ditawarkan sebagai komponen pengujian yang dapat Anda pilih untuk ditambahkan ke resep gambar Anda. Untuk informasi selengkapnya tentang Amazon Inspector, lihat Panduan Pengguna [Amazon Inspector](#). Untuk pengerasan, EC2 Image Builder memvalidasi dengan STIG. Untuk daftar lengkap komponen STIG yang tersedia melalui Image Builder, lihat [Amazon mengelola komponen pengerasan STIG untuk EC2 Image Builder](#). Untuk informasi selengkapnya, lihat [Tolok Ukur Pusat Keamanan Internet \(CIS\)](#).

Memulai dengan EC2 Image Builder

Bab ini membantu Anda mengatur lingkungan dan membuat pipeline gambar otomatis atau pipeline kontainer untuk pertama kalinya, menggunakan EC2 Image Builder Create image pipeline console wizard.

Daftar Isi

- [Prasyarat](#)
- [Akses EC2 Image Builder](#)
- [Buat pipeline gambar menggunakan wizard konsol EC2 Image Builder](#)
- [Buat pipeline gambar kontainer menggunakan wizard konsol EC2 Image Builder](#)

Prasyarat

Verifikasi prasyarat berikut untuk membuat pipeline gambar dengan EC2 Image Builder. Kecuali dinyatakan lain secara khusus, prasyarat diperlukan untuk semua jenis jaringan pipa.

Peran terkait layanan EC2 Image Builder

EC2 Image Builder menggunakan peran terkait layanan untuk memberikan izin ke layanan AWS lain atas nama Anda. Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda membuat resource Image Builder pertama di AWS Management Console, Management Console AWS CLI, atau AWS API, Image Builder akan membuat peran terkait layanan untuk Anda. Untuk informasi selengkapnya tentang peran terkait layanan yang dibuat Image Builder di akun Anda, lihat. [Menggunakan peran terkait layanan untuk EC2 Image Builder](#)

Persyaratan konfigurasi

- Image Builder mendukung [AWS PrivateLink](#). Untuk informasi selengkapnya tentang mengonfigurasi titik akhir VPC untuk Image Builder, lihat. [EC2 Image Builder dan antarmuka titik akhir VPC \(\) AWS PrivateLink](#)
- Image Builder mendukung EC2-Classic.
- Instance yang digunakan Image Builder untuk membuat image kontainer harus memiliki akses internet untuk mengunduh AWS CLI dari Amazon S3, dan untuk mengunduh gambar dasar dari

repositori Docker Hub, jika ada. Image Builder menggunakan AWS CLI untuk mendapatkan Dockerfile dari resep wadah, di mana ia disimpan sebagai data.

- Instance yang digunakan Image Builder untuk membuat gambar dan menjalankan pengujian harus memiliki akses ke layanan Systems Manager. Persyaratan instalasi tergantung pada sistem operasi Anda.

Untuk melihat persyaratan instalasi untuk gambar dasar Anda, pilih tab yang cocok dengan sistem operasi gambar dasar Anda.

Linux

Untuk instans Amazon EC2 Linux, Image Builder menginstal Agen Systems Manager pada instance build jika belum ada, dan menghapusnya sebelum membuat image.

Windows

Image Builder tidak menginstal Agen Manajer Sistem di instans build Amazon EC2 Windows Server. Jika image dasar Anda tidak diinstal sebelumnya dengan Agen Systems Manager, Anda harus meluncurkan instance dari image sumber Anda, menginstal Systems Manager secara manual pada instance, dan membuat image dasar baru dari instans Anda.

Untuk menginstal agen Systems Manager secara manual di instans Amazon EC2 Windows Server Anda, lihat [Menginstal Agen Manajer Sistem secara manual pada instans EC2 untuk Windows Server](#) di Panduan Pengguna. AWS Systems Manager

Repositori kontainer (pipa gambar kontainer)

Untuk pipeline image container, resep mendefinisikan konfigurasi untuk image Docker yang diproduksi dan disimpan dalam repositori kontainer target. Anda harus membuat repositori target sebelum membuat resep wadah untuk gambar Docker Anda.

Image Builder menggunakan Amazon ECR sebagai repositori target untuk gambar kontainer. Untuk membuat repositori Amazon ECR, ikuti langkah-langkah yang dijelaskan dalam [Membuat repositori di Panduan Pengguna](#) Amazon Elastic Container Registry.

AWS Identity and Access Management (IAM)

Peran IAM yang Anda kaitkan dengan profil instans harus memiliki izin untuk menjalankan komponen build dan test yang disertakan dalam image Anda. Kebijakan peran IAM berikut harus dilampirkan ke peran IAM yang terkait dengan profil instance:

- EC2InstanceProfileForImageBuilder
- EC2InstanceProfileForImageBuilderECRContainerBuilds
- AmazonSSM ManagedInstanceCore

Jika Anda mengonfigurasi logging, profil instance yang ditentukan dalam konfigurasi infrastruktur Anda harus memiliki `s3:PutObject` izin untuk bucket target (`arn:aws:s3:::BucketName/*`). Sebagai contoh:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::bucket-name/*"
    }
  ]
}
```

Lampirkan kebijakan

Langkah-langkah berikut memandu Anda melalui proses melampirkan kebijakan IAM ke peran IAM untuk memberikan izin sebelumnya.

1. Login ke Konsol Manajemen AWS dan buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi di sebelah kiri, pilih Kebijakan.
3. Filter daftar kebijakan dengan EC2InstanceProfileForImageBuilder
4. Pilih bullet di sebelah kebijakan, dan dari daftar tarik-turun Tindakan kebijakan, pilih Lampirkan.
5. Pilih nama peran IAM yang akan dilampirkan kebijakan.
6. Pilih Lampirkan kebijakan.
7. Ulangi langkah 3-6 untuk kebijakan EC2InstanceProfileForImageBuilderECRContainerBuilds dan AmazonSSM ManagedInstanceCore.

Note

Jika ingin menyalin gambar yang dibuat dengan Image Builder ke akun lain, Anda harus membuat `EC2ImageBuilderDistributionCrossAccountRole` peran di semua akun target, dan melampirkan kebijakan [Ec2ImageBuilderCrossAccountDistributionAccesskebijakan](#) terkelola ke peran tersebut. Untuk informasi selengkapnya, lihat [Bagikan sumber daya EC2 Image Builder](#).

Akses EC2 Image Builder

Anda dapat mengelola EC2 Image Builder dari salah satu antarmuka berikut.

- Halaman arahan konsol EC2 Image Builder. Dari konsol [EC2 Image Builder](#).
- AWS Command Line Interface(AWS CLI). Anda dapat menggunakan AWS CLI untuk mengakses operasi AWS API. Untuk informasi selengkapnya, lihat [Menginstal Antarmuka Baris AWS Perintah](#) di Panduan AWS Command Line Interface Pengguna.
- AWS Alat untuk SDK. Anda dapat menggunakan [AWSSDK dan Tools](#) untuk mengakses dan mengelola Image Builder menggunakan bahasa pilihan Anda.

Buat pipeline gambar menggunakan wizard konsol EC2 Image Builder

Tutorial ini memandu Anda melalui pembuatan pipeline otomatis untuk membangun dan memelihara image EC2 Image Builder yang disesuaikan menggunakan wizard konsol Create image pipeline. Untuk membantu Anda melewati langkah-langkah secara efisien, pengaturan default digunakan saat tersedia, dan bagian opsional dilewati.

Buat alur kerja pipa gambar

- [Langkah 1: Tentukan detail pipa](#)
- [Langkah 2: Pilih resep](#)
- [Langkah 3: Tentukan konfigurasi infrastruktur - opsional](#)
- [Langkah 4: Tentukan pengaturan distribusi - opsional](#)
- [Langkah 5: Tinjau](#)
- [Langkah 6: Membersihkan](#)

Langkah 1: Tentukan detail pipa

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Untuk mulai membuat pipeline, pilih Create image pipeline.
3. Di bagian Umum, masukkan nama Pipeline Anda (wajib).

Tip

Koleksi metadata yang disempurnakan diaktifkan secara default. Untuk memastikan kompatibilitas antara komponen dan gambar dasar, tetap menyala.

4. Di bagian Jadwal pembuatan, Anda dapat menyimpan default untuk opsi Jadwal. Perhatikan bahwa zona waktu yang ditampilkan untuk jadwal default adalah Universal Coordinated Time (UTC). Untuk informasi selengkapnya tentang waktu UTC, dan untuk menemukan offset untuk zona waktu Anda, lihat [Singkatan Zona Waktu — Daftar Seluruh Dunia](#).

Untuk pengaturan pembaruan Dependensi, pilih Run pipeline pada waktu yang dijadwalkan jika ada opsi pembaruan ketergantungan. Pengaturan ini menyebabkan pipeline Anda memeriksa pembaruan sebelum memulai pembuatan. Jika tidak ada pembaruan, itu melewatkan pembuatan pipeline terjadwal.

Note


Untuk memastikan bahwa pipeline mengenali pembaruan dan build dependensi seperti yang diharapkan, Anda harus menggunakan versi semantik (x.xx) untuk image dan komponen dasar Anda. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. [Versi semantik](#)

5. Pilih Berikutnya untuk melanjutkan ke langkah berikutnya.

Langkah 2: Pilih resep

1. Image Builder default menggunakan resep yang ada di bagian Resep. Untuk pertama kalinya, pilih opsi Buat resep baru.
2. Di bagian Jenis gambar, pilih opsi Amazon Machine Image (AMI) untuk membuat pipeline gambar yang akan menghasilkan dan mendistribusikan AMI.
3. Di bagian Umum, masukkan kotak yang diperlukan berikut:

- Nama — nama resep Anda
 - Versi — versi resep Anda (gunakan format <major>. <minor>. <patch>, di mana mayor, minor, dan patch adalah nilai integer). Resep baru umumnya dimulai dengan 1.0.0.
4. Di bagian Source image, pertahankan nilai default untuk Select image, Image Operating System (OS), dan Image origin. Ini menghasilkan daftar AMI Amazon Linux 2, yang dikelola oleh Amazon, untuk Anda pilih untuk gambar dasar Anda.
- a. Dari dropdown Nama gambar, pilih gambar.
 - b. Pertahankan default untuk opsi Auto-versioning (Gunakan versi OS terbaru yang tersedia).

 Note

Pengaturan ini memastikan bahwa pipeline Anda menggunakan versi semantik untuk image dasar, untuk mendeteksi pembaruan dependensi untuk pekerjaan yang dijadwalkan secara otomatis. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. [Versi semantik](#)

5. Di bagian konfigurasi Instans, simpan nilai default untuk agen Systems Manager. Hal ini menyebabkan Image Builder mempertahankan agen Systems Manager setelah build dan pengujian selesai, untuk menyertakan agen Systems Manager di image baru Anda.


Simpan data Pengguna kosong untuk tutorial ini. Anda dapat menggunakan area ini di waktu lain untuk memberikan perintah, atau skrip perintah untuk dijalankan saat meluncurkan instance build. Namun, ini menggantikan perintah apa pun yang mungkin ditambahkan Image Builder untuk memastikan bahwa Systems Manager diinstal. Ketika Anda menggunakannya, pastikan bahwa agen Systems Manager sudah diinstal sebelumnya pada image dasar Anda, atau bahwa Anda menyertakan instalasi dalam data pengguna Anda.

6. Di bagian Components, Anda harus memilih setidaknya satu komponen build.

Di panel Build components — Amazon Linux, Anda dapat menelusuri komponen yang tercantum di halaman. Gunakan kontrol pagination di sudut kanan atas untuk menavigasi melalui komponen tambahan yang tersedia untuk OS gambar dasar Anda. Anda juga dapat mencari komponen tertentu, atau membuat komponen build sendiri menggunakan Component manager.

Untuk tutorial ini, pilih komponen yang memperbarui Linux dengan pembaruan keamanan terbaru, sebagai berikut:

- a. Filter hasil dengan memasukkan kata update di bilah pencarian yang terletak di bagian atas panel.
- b. Pilih kotak centang untuk komponen `update-linux build`.
- c. Gulir ke bawah, dan di sudut kanan atas daftar Komponen yang dipilih, pilih Perluas semua.
- d. Pertahankan default untuk opsi Pembuatan Versi (Gunakan versi komponen terbaru yang tersedia).


 Note

Pengaturan ini memastikan bahwa pipeline Anda menggunakan versi semantik untuk komponen yang dipilih, untuk mendeteksi pembaruan dependensi untuk pekerjaan yang dijadwalkan secara otomatis. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat [Versi semantik](#)

Jika Anda memilih komponen yang memiliki parameter input, Anda juga akan melihat parameter di area ini. Parameter tidak tercakup dalam tutorial ini. Untuk informasi selengkapnya tentang penggunaan parameter input dalam komponen Anda, dan menyetelnya di resep, lihat [Kelola parameter AWSTOE komponen dengan EC2 Image Builder](#).

Menyusun ulang komponen (opsional)

Jika Anda telah memilih lebih dari satu komponen untuk disertakan dalam gambar Anda, Anda dapat menggunakan drag-and-drop tindakan untuk mengatur ulang mereka ke dalam urutan yang harus dijalankan selama proses build.

 Note

Komponen pengerasan CIS tidak mengikuti aturan pemesanan komponen standar dalam resep Image Builder. Komponen pengerasan CIS selalu berjalan terakhir untuk memastikan bahwa tes benchmark berjalan terhadap gambar keluaran Anda.

1. Gulir kembali ke daftar komponen yang tersedia.

2. Pilih kotak centang untuk komponen `update-linux-kernel-mainline` build (atau komponen lain pilihan Anda).
 3. Gulir ke bawah ke daftar Komponen yang dipilih, untuk melihat bahwa setidaknya ada dua hasil.
 4. Komponen yang baru ditambahkan mungkin tidak memiliki pengaturan parameter versi atau input yang diperluas. Untuk memperluas pengaturan untuk opsi Pembuatan versi atau Parameter input, Anda dapat memilih panah di sebelah nama pengaturan. Untuk memperluas semua pengaturan untuk semua komponen yang dipilih, Anda dapat mengaktifkan tombol `Expand all off and on`.
 5. Pilih salah satu komponen, dan seret ke atas atau ke bawah untuk mengubah urutan komponen akan berjalan.
 6. Untuk menghapus `update-linux-kernel-mainline` komponen, pilih X dari sudut kanan atas kotak komponen.
 7. Ulangi langkah sebelumnya untuk menghapus komponen lain yang mungkin telah Anda tambahkan, hanya menyisakan `update-linux` komponen yang dipilih.
7. Pilih Berikutnya untuk melanjutkan ke langkah berikutnya.

Langkah 3: Tentukan konfigurasi infrastruktur - opsional

Image Builder meluncurkan instans EC2 di akun Anda untuk menyesuaikan gambar dan menjalankan pengujian validasi. Pengaturan konfigurasi Infrastruktur menentukan detail infrastruktur untuk instance yang akan berjalan Akun AWS selama proses pembuatan.

Di bagian konfigurasi Infrastruktur, opsi Konfigurasi default ke `Create infrastructure configuration using service defaults`. Ini membuat peran IAM dan profil instans terkait untuk instans build dan pengujian EC2 yang digunakan untuk mengonfigurasi gambar Anda. Untuk informasi selengkapnya tentang setelan konfigurasi infrastruktur, lihat [CreateInfrastructureConfiguration](#) di Referensi API EC2 Image Builder.

Untuk tutorial ini, kita menggunakan pengaturan default.

Note

Untuk menentukan subnet yang akan digunakan untuk VPC pribadi, Anda dapat membuat konfigurasi infrastruktur kustom Anda sendiri, atau menggunakan pengaturan yang telah Anda buat.

- Pilih Berikutnya untuk melanjutkan ke langkah berikutnya.

Langkah 4: Tentukan pengaturan distribusi - opsional

Konfigurasi distribusi mencakup nama AMI keluaran, pengaturan Wilayah khusus untuk enkripsi, izin peluncuran, dan, organisasi Akun AWS, dan unit organisasi (OU) yang dapat meluncurkan AMI keluaran, dan konfigurasi lisensi.

Di bagian Pengaturan distribusi, opsi Konfigurasi default ke `Create distribution settings using service defaults`. Opsi ini akan mendistribusikan output AMI ke Wilayah saat ini. Untuk informasi selengkapnya tentang mengonfigurasi setelan distribusi, lihat [Mengelola pengaturan distribusi EC2 Image Builder](#).

Untuk tutorial ini, kita menggunakan pengaturan default.

- Pilih Berikutnya untuk melanjutkan ke langkah berikutnya.

Langkah 5: Tinjau


Bagian Review menampilkan semua pengaturan yang telah Anda konfigurasi. Untuk mengedit informasi di bagian mana pun, pilih tombol Edit yang terletak di sudut kanan atas bagian langkah. Misalnya, jika Anda ingin mengubah nama pipeline Anda, pilih tombol Edit di sudut kanan atas bagian Langkah 1: Detail saluran pipa.

1. Ketika Anda telah meninjau pengaturan Anda, pilih Buat pipeline untuk membuat pipeline Anda.
2. Anda dapat melihat pesan sukses atau gagal di bagian atas halaman, karena sumber daya Anda dibuat untuk pengaturan distribusi, konfigurasi infrastruktur, resep baru, dan pipeline. Untuk melihat detail sumber daya, termasuk pengenalan sumber daya, pilih Lihat detail.
3. Setelah melihat detail sumber daya, Anda dapat melihat detail tentang sumber daya lain dengan memilih jenis sumber daya dari panel navigasi. Misalnya, untuk melihat detail pipeline baru Anda, pilih Pipeline gambar dari panel navigasi. Jika build berhasil, pipeline baru akan ditampilkan di daftar pipeline Image.

Langkah 6: Membersihkan

Lingkungan Image Builder Anda, seperti rumah Anda, membutuhkan perawatan rutin untuk membantu Anda menemukan apa yang Anda butuhkan, dan menyelesaikan tugas Anda tanpa

mengarungi kekacauan. Pastikan untuk secara teratur membersihkan sumber daya sementara yang Anda buat untuk pengujian. Jika tidak, Anda mungkin lupa tentang sumber daya itu, dan kemudian, tidak ingat untuk apa sumber daya itu digunakan. Pada saat itu, mungkin tidak jelas apakah Anda dapat menyingkirkannya dengan aman.

 Tip

Untuk mencegah kesalahan ketergantungan saat Anda menghapus sumber daya, pastikan untuk menghapus sumber daya Anda dalam urutan berikut:

1. Pipa gambar
2. Resep gambar
3. Semua sumber daya yang tersisa

Untuk membersihkan sumber daya yang Anda buat untuk tutorial ini, ikuti langkah-langkah berikut:

Hapus pipa

1. Untuk melihat daftar pipeline build yang dibuat di bawah akun Anda, pilih Pipeline gambar dari panel navigasi.
2. Pilih kotak centang di samping nama Pipeline untuk memilih pipeline yang ingin Anda hapus.
3. Di bagian atas panel pipa gambar, pada menu Tindakan, pilih Hapus.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** dalam kotak, dan pilih Hapus.

Hapus resepnya

1. Untuk melihat daftar resep yang dibuat di bawah akun Anda, pilih Resep gambar dari panel navigasi.
2. Pilih kotak centang di sebelah Nama resep untuk memilih resep yang ingin Anda hapus.
3. Di bagian atas panel Resep gambar, pada menu Tindakan, pilih Hapus resep.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** dalam kotak, dan pilih Hapus.

Hapus konfigurasi infrastruktur

1. Untuk melihat daftar konfigurasi infrastruktur yang dibuat di bawah akun Anda, pilih Konfigurasi infrastruktur dari panel navigasi.
2. Pilih kotak centang di sebelah Nama konfigurasi untuk memilih konfigurasi infrastruktur yang ingin Anda hapus.
3. Di bagian atas panel konfigurasi Infrastruktur, pilih Hapus.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** dalam kotak, dan pilih Hapus.

Hapus pengaturan distribusi

1. Untuk melihat daftar setelan distribusi yang dibuat di bawah akun Anda, pilih Pengaturan distribusi dari panel navigasi.
2. Pilih kotak centang di sebelah Nama konfigurasi untuk memilih pengaturan distribusi yang Anda buat untuk tutorial ini.
3. Di bagian atas panel Pengaturan distribusi, pilih Hapus.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** dalam kotak, dan pilih Hapus.

Hapus gambar

Ikuti langkah-langkah ini untuk memverifikasi bahwa Anda telah menghapus gambar apa pun yang dibuat dari saluran tutorial. Tutorial ini tidak mungkin membuat gambar kecuali waktu yang cukup telah berlalu sejak Anda membuat pipeline yang berjalan, sesuai dengan jadwal pembuatan.

1. Untuk melihat daftar gambar yang dibuat di bawah akun Anda, pilih Gambar dari panel navigasi.
2. Pilih Versi gambar untuk gambar yang ingin Anda hapus. Ini membuka halaman Image build version.
3. Pilih kotak centang di sebelah Versi untuk gambar apa pun yang ingin Anda hapus. Anda dapat memilih lebih dari satu versi gambar sekaligus.
4. Di bagian atas panel Image build version, pilih Delete version.
5. Untuk mengonfirmasi penghapusan, masukkan **Delete** dalam kotak, dan pilih Hapus.

Buat pipeline gambar kontainer menggunakan wizard konsol EC2 Image Builder

Tutorial ini memandu Anda melalui pembuatan pipeline otomatis untuk membangun dan memelihara image EC2 Image Builder Docker yang disesuaikan menggunakan wizard konsol Create image pipeline. Untuk membantu Anda menelusuri langkah-langkah secara efisien, pengaturan default digunakan saat tersedia, dan bagian opsional dilewati.

Buat alur kerja pipa gambar

- [Langkah 1: Tentukan detail pipa](#)
- [Langkah 2: Pilih resep](#)
- [Langkah 3: Tentukan konfigurasi infrastruktur - opsional](#)
- [Langkah 4: Tentukan pengaturan distribusi - opsional](#)
- [Langkah 5: Tinjau](#)
- [Langkah 6: Membersihkan](#)

Langkah 1: Tentukan detail pipa

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Untuk mulai membuat pipeline, pilih Create image pipeline.
3. Di bagian Umum, masukkan nama Pipeline Anda (wajib).
4. Di bagian Jadwal pembuatan, Anda dapat menyimpan default untuk opsi Jadwal. Perhatikan bahwa zona waktu yang ditampilkan untuk jadwal default adalah Universal Coordinated Time (UTC). Untuk informasi selengkapnya tentang waktu UTC, dan untuk menemukan offset untuk zona waktu Anda, lihat [Singkatan Zona Waktu — Daftar Seluruh Dunia](#).

Untuk pengaturan pembaruan Dependensi, pilih Run pipeline pada waktu yang dijadwalkan jika ada opsi pembaruan ketergantungan. Pengaturan ini menyebabkan pipeline Anda memeriksa pembaruan sebelum memulai pembuatan. Jika tidak ada pembaruan, itu melewatkan pembuatan pipeline terjadwal.

Note

Untuk memastikan bahwa pipeline mengenali pembaruan dan build dependensi seperti yang diharapkan, Anda harus menggunakan versi semantik (x.xx) untuk image dan

komponen dasar Anda. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. [Versi semantik](#)

5. Pilih Berikutnya untuk melanjutkan ke langkah berikutnya.

Langkah 2: Pilih resep

1. Image Builder default menggunakan resep yang ada di bagian Resep. Untuk pertama kalinya, pilih opsi Buat resep baru.
2. Di bagian Jenis gambar, pilih opsi image Docker untuk membuat pipeline kontainer yang akan menghasilkan image Docker dan mendistribusikannya ke repositori Amazon ECR di Wilayah target.
3. Di bagian Umum, masukkan kotak yang diperlukan berikut:
 - Nama — nama resep Anda
 - Versi — versi resep Anda (gunakan formatnya <major>. <minor>. <patch>, di mana mayor, minor, dan patch adalah nilai integer). Resep baru umumnya dimulai dengan 1.0.0.
4. Di bagian Source image, pertahankan nilai default untuk Select image, Image Operating System (OS), dan Image origin. Ini menghasilkan daftar gambar kontainer Amazon Linux 2, yang dikelola oleh Amazon, untuk Anda pilih untuk gambar dasar Anda.
 - a. Dari dropdown Nama gambar, pilih gambar.
 - b. Pertahankan default untuk opsi Auto-versioning (Gunakan versi OS terbaru yang tersedia).

Note


Pengaturan ini memastikan bahwa pipeline Anda menggunakan versi semantik untuk image dasar, untuk mendeteksi pembaruan dependensi untuk pekerjaan yang dijadwalkan secara otomatis. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. [Versi semantik](#)

5. Di bagian Components, Anda harus memilih setidaknya satu komponen build.

Di panel Build components — Amazon Linux, Anda dapat menelusuri komponen yang tercantum di halaman. Gunakan kontrol pagination di sudut kanan atas untuk menavigasi melalui komponen tambahan yang tersedia untuk OS gambar dasar Anda. Anda juga dapat mencari komponen tertentu, atau membuat komponen build sendiri menggunakan Component manager.

Untuk tutorial ini, pilih komponen yang memperbarui Linux dengan pembaruan keamanan terbaru, sebagai berikut:

- a. Filter hasil dengan memasukkan kata `update` di bilah pencarian yang terletak di bagian atas panel.
- b. Pilih kotak centang untuk komponen `update-linux build`.
- c. Gulir ke bawah, dan di sudut kanan atas daftar Komponen yang dipilih, pilih Perluas semua.
- d. Pertahankan default untuk opsi Pembuatan Versi (Gunakan versi komponen terbaru yang tersedia).


 Note

Pengaturan ini memastikan bahwa pipeline Anda menggunakan versi semantik untuk komponen yang dipilih, untuk mendeteksi pembaruan dependensi untuk pekerjaan yang dijadwalkan secara otomatis. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat [Versi semantik](#)

Jika Anda telah memilih komponen yang memiliki parameter input, Anda juga akan melihat parameter di area ini. Parameter tidak tercakup dalam tutorial ini. Untuk informasi selengkapnya tentang penggunaan parameter input dalam komponen Anda, dan menyetelnya di resep, lihat [Kelola parameter AWSTOE komponen dengan EC2 Image Builder](#).

Menyusun ulang komponen (opsional)

Jika Anda telah memilih lebih dari satu komponen untuk disertakan dalam gambar Anda, Anda dapat menggunakan drag-and-drop tindakan untuk mengatur ulang mereka ke dalam urutan yang harus dijalankan selama proses build.

 Note

Komponen pengerasan CIS tidak mengikuti aturan pemesanan komponen standar dalam resep Image Builder. Komponen pengerasan CIS selalu berjalan terakhir untuk memastikan bahwa tes benchmark berjalan terhadap gambar keluaran Anda.

1. Gulir kembali ke daftar komponen yang tersedia.
2. Pilih kotak centang untuk komponen `update-linux-kernel-mainline` build (atau komponen lain pilihan Anda).
3. Gulir ke bawah ke daftar Komponen yang dipilih, untuk melihat bahwa setidaknya ada dua hasil.
4. Komponen yang baru ditambahkan mungkin versinya tidak diperluas. Untuk memperluas opsi Pembuatan Versi, Anda dapat memilih panah di samping opsi Pembuatan Versi, atau Anda dapat mematikan dan mengaktifkan tombol Perluas semua untuk memperluas versi untuk semua komponen yang dipilih.
5. Pilih salah satu komponen, dan seret ke atas atau ke bawah untuk mengubah urutan komponen akan berjalan.
6. Untuk menghapus `update-linux-kernel-mainline` komponen, pilih X dari sudut kanan atas kotak komponen.
7. Ulangi langkah sebelumnya untuk menghapus komponen lain yang mungkin telah Anda tambahkan, hanya menyisakan `update-linux` komponen yang dipilih.
6. Di bagian template Dockerfile, pilih opsi Gunakan contoh. Di panel Konten, perhatikan variabel kontekstual tempat Image Builder menempatkan informasi atau skrip build, berdasarkan resep gambar kontainer Anda.
7. Di bagian Repositori Target, tentukan nama repositori Amazon ECR yang Anda buat sebagai prasyarat untuk tutorial ini. Repositori ini digunakan sebagai pengaturan default untuk konfigurasi distribusi di Wilayah tempat pipeline berjalan (Wilayah 1).

Note

Repositori target harus ada di Amazon ECR untuk semua Wilayah target sebelum distribusi.

8. Pilih Berikutnya untuk melanjutkan ke langkah berikutnya.

Langkah 3: Tentukan konfigurasi infrastruktur - opsional

Image Builder meluncurkan instans EC2 di akun Anda untuk menyesuaikan gambar dan menjalankan pengujian validasi. Pengaturan konfigurasi Infrastruktur menentukan detail infrastruktur untuk instans yang akan berjalan di Anda Akun AWS selama proses pembuatan.

Di bagian konfigurasi Infrastruktur, opsi Konfigurasi default ke `Create infrastructure configuration using service defaults`. Ini membuat peran IAM dan profil instance terkait yang digunakan oleh instance build untuk mengonfigurasi gambar kontainer Anda. Anda juga dapat membuat konfigurasi infrastruktur kustom Anda sendiri, atau menggunakan pengaturan yang telah Anda buat. Untuk informasi selengkapnya tentang setelan konfigurasi infrastruktur, lihat [CreateInfrastructureConfiguration](#) di Referensi API EC2 Image Builder.

Untuk tutorial ini, kita menggunakan pengaturan default.

- Pilih Berikutnya untuk melanjutkan ke langkah berikutnya.

Langkah 4: Tentukan pengaturan distribusi - opsional

Pengaturan distribusi terdiri dari Wilayah target, dan nama repositori Amazon ECR target. Gambar Output Docker disebar ke repositori Amazon ECR bernama di setiap Wilayah.

Di bagian Pengaturan distribusi, opsi Konfigurasi default ke `Create distribution settings using service defaults`. Opsi ini akan mendistribusikan image Docker keluaran ke repositori Amazon ECR yang ditentukan dalam resep container Anda untuk Wilayah tempat pipeline Anda berjalan (Wilayah 1). Jika Anda memilih `Create new distribution settings`, Anda dapat mengganti repositori ECR untuk Wilayah saat ini, dan menambahkan lebih banyak Wilayah untuk distribusi.

Untuk tutorial ini, kita menggunakan pengaturan default.

- Pilih Berikutnya untuk melanjutkan ke langkah berikutnya.

Langkah 5: Tinjau

Bagian Review menampilkan semua pengaturan yang telah Anda konfigurasi. Untuk mengedit informasi di bagian mana pun, pilih tombol Edit yang terletak di sudut kanan atas bagian langkah. Misalnya, jika Anda ingin mengubah nama pipeline, pilih tombol Edit di sudut kanan atas bagian Langkah 1: Detail saluran pipa.

1. Ketika Anda telah meninjau pengaturan Anda, pilih Buat pipeline untuk membuat pipeline Anda.
2. Anda dapat melihat pesan sukses atau gagal di bagian atas halaman, karena sumber daya Anda dibuat untuk pengaturan distribusi, konfigurasi infrastruktur, resep baru, dan pipeline. Untuk melihat detail sumber daya, termasuk pengenalan sumber daya, pilih Lihat detail.

3. Setelah melihat detail sumber daya, Anda dapat melihat detail tentang sumber daya lain dengan memilih jenis sumber daya dari panel navigasi. Misalnya, untuk melihat detail pipeline baru Anda, pilih Pipeline gambar dari panel navigasi. Jika build berhasil, pipeline baru akan ditampilkan di daftar pipeline Image.

Langkah 6: Membersihkan

Lingkungan Image Builder Anda, seperti rumah Anda, membutuhkan perawatan rutin untuk membantu Anda menemukan apa yang Anda butuhkan, dan menyelesaikan tugas Anda tanpa mengarungi kekacauan. Pastikan untuk secara teratur membersihkan sumber daya sementara yang Anda buat untuk pengujian. Jika tidak, Anda mungkin lupa tentang sumber daya itu, dan kemudian, tidak ingat untuk apa sumber daya itu digunakan. Pada saat itu, mungkin tidak jelas apakah Anda dapat menyingkirkannya dengan aman.

Tip

Untuk mencegah kesalahan ketergantungan saat Anda menghapus sumber daya, pastikan untuk menghapus sumber daya Anda dalam urutan berikut:

1. Pipa gambar
2. Resep gambar
3. Semua sumber daya yang tersisa

Untuk membersihkan sumber daya yang Anda buat untuk tutorial ini, ikuti langkah-langkah berikut:

Hapus pipa

1. Untuk melihat daftar pipeline build yang dibuat di bawah akun Anda, pilih Pipeline gambar dari panel navigasi.
2. Pilih kotak centang di samping nama Pipeline untuk memilih pipeline yang ingin Anda hapus.
3. Di bagian atas panel pipa gambar, pada menu Tindakan, pilih Hapus.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** di dalam kotak, dan pilih Hapus.

Hapus resep wadah

1. Untuk melihat daftar resep kontainer yang dibuat di bawah akun Anda, pilih Resep kontainer dari panel navigasi.
2. Pilih kotak centang di sebelah Nama resep untuk memilih resep yang ingin Anda hapus.
3. Di bagian atas panel Resep kontainer, pada menu Tindakan, pilih Hapus resep.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** di dalam kotak, dan pilih Hapus.

Hapus konfigurasi infrastruktur

1. Untuk melihat daftar konfigurasi infrastruktur yang dibuat di bawah akun Anda, pilih Konfigurasi infrastruktur dari panel navigasi.
2. Pilih kotak centang di sebelah Nama konfigurasi untuk memilih konfigurasi infrastruktur yang ingin Anda hapus.
3. Di bagian atas panel konfigurasi Infrastruktur, pilih Hapus.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** di dalam kotak, dan pilih Hapus.

Hapus pengaturan distribusi

1. Untuk melihat daftar setelan distribusi yang dibuat di bawah akun Anda, pilih Pengaturan distribusi dari panel navigasi.
2. Pilih kotak centang di sebelah Nama konfigurasi untuk memilih pengaturan distribusi yang Anda buat untuk tutorial ini.
3. Di bagian atas panel Pengaturan distribusi, pilih Hapus.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** di dalam kotak, dan pilih Hapus.

Hapus gambar

Ikuti langkah-langkah ini untuk memverifikasi bahwa Anda telah menghapus gambar apa pun yang dibuat dari saluran tutorial. Tutorial ini tidak mungkin membuat gambar kecuali waktu yang cukup telah berlalu sejak Anda membuat pipeline yang berjalan, sesuai dengan jadwal pembuatan.

1. Untuk melihat daftar gambar yang dibuat di bawah akun Anda, pilih Gambar dari panel navigasi.
2. Pilih Versi gambar untuk gambar yang ingin Anda hapus. Ini membuka halaman Image build version.

3. Pilih kotak centang di sebelah Versi untuk gambar apa pun yang ingin Anda hapus. Anda dapat memilih lebih dari satu versi gambar sekaligus.
4. Di bagian atas panel Image build version, pilih Delete version.
5. Untuk mengonfirmasi penghapusan, masukkan **Delete** di dalam kotak, dan pilih Hapus.

AWS Task Orchestrator and Executormanajer komponen

EC2 Image Builder menggunakan AWS Task Orchestrator and Executor aplikasi AWSTOE () untuk mengatur alur kerja yang kompleks, memodifikasi konfigurasi sistem, dan menguji sistem Anda tanpa menulis kode. Aplikasi ini mengelola dan menjalankan komponen yang mengimplementasikan skema dokumen deklaratifnya.

Karena ini adalah aplikasi mandiri, itu tidak memerlukan pengaturan server tambahan. Ini dapat berjalan di infrastruktur cloud apa pun dan di tempat.

Daftar Isi

- [AWSTOE unduh](#)
- [Wilayah yang Didukung](#)
- [Memulai dengan AWSTOE](#)
- [Gunakan dokumen komponen di AWSTOE](#)
- [Modul aksi didukung oleh manajer AWSTOE komponen](#)
- [Konfigurasi input untuk perintah AWSTOE run](#)
- [Komponen terkelola paket distributor untuk Windows](#)
- [Komponen pengerasan CIS](#)
- [Amazon mengelola komponen pengerasan STIG untuk EC2 Image Builder](#)
- [AWSTOEreferensi perintah](#)

AWSTOE unduh

Untuk menginstalAWSTOE, pilih tautan unduhan untuk arsitektur dan platform Anda. Jika Anda melampirkan ke titik akhir VPC untuk layanan Anda (Image Builder, misalnya), itu harus memiliki kebijakan endpoint khusus yang dilampirkan yang menyertakan akses ke bucket S3 untuk unduhan. AWSTOE Jika tidak, instance build dan test Anda tidak akan dapat mengunduh skrip bootstrap (`bootstrap.sh`) dan menginstal AWSTOE aplikasi. Untuk informasi selengkapnya, lihat [Membuat kebijakan titik akhir VPC untuk Image Builder](#).

Important

AWSmenghapus dukungan untuk TLS versi 1.0 dan 1.1 secara bertahap. Untuk mengakses bucket S3 untuk AWSTOE unduhan, perangkat lunak klien Anda harus menggunakan TLS

versi 1.2 atau yang lebih baru. Untuk informasi lebih lanjut, lihat [posting Blog AWS Keamanan ini](#).

Arsitektur	Platform	Tautan unduhan	Contoh
386	AL 2 dan 2023 RHEL 7 dan 8 Ubuntu 16.04, 18.04, 20.04, dan 22.04 CentOS 7 dan 8 SUSE 12 dan 15	<a href="https://awsstoe-<region>.s3.amazonaws.com/latest/linux/386/awstoe">https://awsstoe-<region>.s3.amazonaws.com/latest/linux/386/awstoe	https://awsstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/386/awstoe
AMD64	Windows Server 2012 R2, 2016, 2019, dan 2022	<a href="https://awsstoe-<region>.s3.amazonaws.com/latest/windows/amd64/awstoe.exe">https://awsstoe-<region>.s3.amazonaws.com/latest/windows/amd64/awstoe.exe	https://awsstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/windows/amd64/awstoe.exe
AMD64	AL 2 dan 2023 RHEL 7 dan 8 Ubuntu 16.04, 18.04, 20.04, dan 22.04 CentOS 7 dan 8 CentOS Aliran 8 SUSE 12 dan 15	<a href="https://awsstoe-<region>.s3.amazonaws.com/latest/linux/amd64/awstoe">https://awsstoe-<region>.s3.amazonaws.com/latest/linux/amd64/awstoe	https://awsstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/amd64/awstoe
ARM64	AL 2 dan 2023 RHEL 7 dan 8	<a href="https://awsstoe-<region>.s3.amazonaws.com/latest/">https://awsstoe-<region>.s3.amazonaws.com/latest/	https://awsstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/

Arsitektur	Platform	Tautan unduhan	Contoh
	Ubuntu 16.04, 18.04, 20.04, dan 22.04	linux/arm64/awstoe	latest/linux/arm64/awstoe
	CentOS 7 dan 8		
	CentOS Aliran 8		
	SUSE 12 dan 15		

Wilayah yang Didukung

AWSTOE didukung sebagai aplikasi mandiri di Wilayah berikut.

Wilayah AWS nama	Wilayah AWS
Timur AS (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
AWS GovCloud (AS-Timur)	us-gov-east-1
AWS GovCloud (AS-Barat)	us-gov-west-1
US West (Northern California)	us-west-1
US West (Oregon)	as-barat-2
Afrika (Cape Town)	af-selatan-1
Asia Pasifik (Hong Kong)	ap-east-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pasifik (Mumbai)	ap-south-1
Asia Pacific (Hyderabad)	ap-south-2

Wilayah AWS nama	Wilayah AWS
Asia Pasifik (Singapura)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Jakarta)	ap-southeast-3
Asia Pasifik (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Zurich)	eu-central-2
Eropa (Stockholm)	eu-north-1
Eropa (Milan)	eu-south-1
Europe (Spain)	eu-south-2
Eropa (Irlandia)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Israel (Tel Aviv)	il-central-1
Timur Tengah (UEA)	eu-central-1
Timur Tengah (Bahrain)	me-selatan-1
Amerika Selatan (São Paulo)	sa-east-1
Tiongkok (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1

Memulai dengan AWSTOE

Aplikasi AWS Task Orchestrator and Executor (AWSTOE) adalah aplikasi mandiri yang membuat, memvalidasi, dan menjalankan perintah dalam kerangka definisi komponen. AWS layanan dapat digunakan AWSTOE untuk mengatur alur kerja, menginstal perangkat lunak, memodifikasi konfigurasi sistem, dan menguji build gambar.

Ikuti langkah-langkah ini untuk menginstal AWSTOE aplikasi dan menggunakannya untuk pertama kalinya.

Verifikasi tanda tangan unduhan AWSTOE instalasi

Bagian ini menjelaskan proses yang disarankan untuk memverifikasi validitas unduhan instalasi untuk AWSTOE sistem operasi berbasis Linux dan Windows.

Topik

- [Verifikasi tanda tangan unduhan AWSTOE instalasi di Linux](#)
- [Verifikasi tanda tangan unduhan AWSTOE instalasi di Windows](#)

Verifikasi tanda tangan unduhan AWSTOE instalasi di Linux

Topik ini menjelaskan proses yang direkomendasikan untuk memverifikasi validitas unduhan instalasi untuk sistem AWSTOE operasi berbasis Linux.

Setiap kali Anda mengunduh aplikasi dari internet, kami sarankan Anda mengautentikasi identitas penerbit perangkat lunak. Juga, periksa apakah aplikasi tidak diubah atau rusak sejak diterbitkan. Hal ini akan melindungi Anda agar tidak menginstal versi aplikasi yang berisi virus atau kode berbahaya lainnya.

Jika, setelah menjalankan langkah-langkah dalam topik ini, Anda menentukan bahwa perangkat lunak untuk diubah atau rusak, jangan jalankan file instalasi. AWSTOE Sebagai gantinya, hubungi AWS Support Untuk informasi selengkapnya tentang opsi dukungan Anda, lihat [AWS Support](#).

AWSTOE file untuk sistem operasi berbasis Linux ditandatangani menggunakan GnuPG, implementasi open source dari standar Pretty Good Privacy (OpenPGP) untuk tanda tangan digital yang aman. GnuPG (juga dikenal sebagai GPG) menyediakan otentikasi dan pemeriksaan integritas melalui tanda tangan digital. Amazon EC2 menerbitkan kunci publik dan tanda tangan yang dapat Anda gunakan

untuk memverifikasi alat Amazon EC2 CLI yang diunduh. Untuk informasi lebih lanjut tentang PGP dan GnuPG (GPG), lihat <http://www.gnupg.org>.

Langkah pertamanya adalah membangun kepercayaan dengan penerbit perangkat lunak. Unduh kunci publik dari penerbit perangkat lunak, periksa apakah pemilik kunci publik adalah benar-benar pemiliknya, lalu tambahkan kunci publik ke dalam keyring Anda. Keyring adalah kumpulan kunci publik yang diketahui. Setelah menetapkan autentikasi kunci publik, Anda dapat menggunakannya untuk memverifikasi tanda tangan aplikasi.

Topik

- [Menginstal alat GPG](#)
- [Mengautentikasi dan mengimpor kunci publik](#)
- [Memverifikasi tanda tangan paket](#)

Menginstal alat GPG

Jika sistem operasi Anda adalah Linux atau Unix, alat GPG mungkin sudah terinstal. Untuk menguji apakah alat ini sudah terinstal di sistem Anda, ketikkan `gpg` pada jendela perintah. Jika alat GPG telah terinstal, Anda akan melihat prompt perintah GPG. Jika alat GPG tidak diinstal, Anda melihat pesan kesalahan yang menyatakan bahwa perintah tidak dapat ditemukan. Anda dapat menginstal paket GnuPG dari repositori.

Untuk menginstal alat GPG pada Linux berbasis Debian

- Dari terminal, jalankan perintah berikut:`apt-get install gnupg`.

Untuk menginstal alat GPG pada Linux berbasis Red Hat

- Dari terminal, jalankan perintah berikut:`yum install gnupg`.

Mengautentikasi dan mengimpor kunci publik

Langkah selanjutnya dalam proses ini adalah mengautentikasi kunci AWSTOE publik dan menambahkannya sebagai kunci tepercaya di GPG keyring Anda.

Untuk mengautentikasi dan mengimpor kunci AWSTOE publik

1. Dapatkan salinan kunci build GPG publik kami dengan melakukan salah satu langkah berikut:

- Unduh kunci dari [https://awstoe-**<region>**.s3.**<region>**.amazonaws.com/assets/awstoe.gpg](https://awstoe-<region>.s3.<region>.amazonaws.com/assets/awstoe.gpg). Sebagai contoh, <https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/assets/awstoe.gpg>.
- Salin kunci dari teks berikut dan tempelkan ke file bernama `awstoe.gpg`. Pastikan untuk memasukkan semua hal berikut ini:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQENBF8UqwsBCACdiRF2bkZYaFSDPFC+LIkWLwFvtUCRwAHtD8KIwTJ6LVn3fHAU
GhuK0ZH9mRrqrT2bq/xJjGsnF9VqTj2AJqndGJdDjz75YCZYM+ocZ+r5HSJaeW9i
S5dykHj7Txti2zHe0G5+W0v7v5bPi2sPHsN7XWQ7+G2AMEPTz8PjxY//I0DvMQns
S1e3l9hz6wCC1z1l9LbBzTyHfSm5ucTXvNe88XX5Gmt370CDM7vf1i0Ctv8WfoLN
6jbxuA/sV71yIkPm9IYp3+GvaKeT870+sn8/J00KE/U4sJV1ppbqmuUzDfhrZUaw
8eW8IN9A1FTIuWiZED/5L83UZuQs1S7s2PjLABEBAAG0GkFXU1RPRSA8YXdzdG9l
QGftYXpvi5jb20+iQE5BBMBCAAjBQJfFKsLAhsDBwsJCAcDAgEGFQgCCQoLBBYC
AwEChgECF4AACgkQ3r3BVvWuvFJGiwf9EVmrBR77+Qe/DUeXZJYoaFr7If/fVDZl
6V3TC6p0J0Veme7uXleRUTF0jzbh+7e5sDX19HrnPquzCnzfMiqbp4lSoeUuNdOf
FcpuTCQH+M+sIEIgpNo4PL10Uj2uE1o++mxmonBl/Krk+hly8hB2L/9n/vW3L7BN
0Mb1L19PmgGPbWipcT8KRdz4SUex9TXGYzj1Wb3jU3uXetdaQY1M3kVKE1siRsRN
YYDtpcjmbbhjpu4xm19aFqNoAHCDctEsXJA/mkU3erwIRocPyjAZE2dn1kL9ZkFZ
z9DQkcIarbCnybDM51emBbdhXJ6hezJE/b17VA0t1fY04MoEkn6oJg==
=oyze
-----END PGP PUBLIC KEY BLOCK-----
```

2. Pada prompt perintah di direktori tempat Anda menyimpan `awstoe.gpg`, gunakan perintah berikut untuk mengimpor kunci AWSTOE publik ke keyring Anda.

```
gpg --import awstoe.gpg
```

Perintah tersebut mengembalikan hasil yang serupa dengan berikut ini:

```
gpg: key F5AEB52: public key "AWSTOE <awstoe@amazon.com>" imported
gpg: Total number processed: 1
gpg:             imported: 1 (RSA: 1)
```

Catat nilai utama; Anda membutuhkannya pada langkah berikutnya. Dalam contoh sebelumnya, nilai utama adalah `F5AEB52`.

3. Verifikasi sidik jari dengan menjalankan perintah berikut, mengganti key-value dengan nilai dari langkah sebelumnya:

```
gpg --fingerprint key-value
```

Perintah ini mengembalikan hasil yang serupa dengan berikut ini:

```
pub 2048R/F5AEBC52 2020-07-19
    Key fingerprint = F6DD E01C 869F D639 15E5 5742 DEBD C156 F5AE BC52
uid [ unknown] AWSTOE <awstoe@amazon.com>
```

Selain itu, string sidik jari harus identik dengan F6DD E01C 869F D639 15E5 5742 DEBD C156 F5AE BC52, seperti yang ditunjukkan dalam contoh sebelumnya. Bandingkan sidik jari kunci yang dikembalikan dengan yang dipublikasikan di halaman ini. Mereka harus cocok. Jika tidak cocok, jangan instal skrip AWSTOE instalasi, dan hubungi AWS Support.

Memverifikasi tanda tangan paket

Setelah Anda menginstal GPG alat, mengautentikasi dan mengimpor kunci AWSTOE publik, dan memverifikasi bahwa kunci publik dipercaya, Anda siap untuk memverifikasi tanda tangan skrip instalasi.

Untuk memverifikasi tanda tangan skrip instalasi

1. Pada prompt perintah, jalankan perintah berikut untuk mengunduh biner aplikasi:

```
curl -O https://awstoe-<region>.s3.<region>.amazonaws.com/latest/
linux/<architecture>/awstoe
```

Sebagai contoh:

```
curl -O https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/amd64/
awstoe
```

Nilai yang didukung untuk **architecture** bisa amd64, x86_64, dan arm64.

2. Pada prompt perintah, jalankan perintah berikut untuk mengunduh file tanda tangan untuk biner aplikasi yang sesuai dari jalur key prefix S3 yang sama:

```
curl -O https://awstoe-<region>.s3.<region>.amazonaws.com/latest/  
linux/<architecture>/awstoe.sig
```

Sebagai contoh:

```
curl -O https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/amd64/  
awstoe.sig
```

Nilai yang didukung untuk **architecture** bisa `amd64`, `386`, dan `arm64`.

3. Verifikasi tanda tangan dengan menjalankan perintah berikut pada prompt perintah di direktori tempat Anda menyimpan `awstoe.sig` dan file AWSTOE instalasi. Kedua file tersebut harus ada.

```
gpg --verify ./awstoe.sig ~/awstoe
```

Output-nya akan terlihat seperti berikut ini:

```
gpg: Signature made Mon 20 Jul 2020 08:54:55 AM IST using RSA key ID F5AEB52  
gpg: Good signature from "AWSTOE awstoe@amazon.com" [unknown]  
gpg: WARNING: This key is not certified with a trusted signature!  
gpg:          There is no indication that the signature belongs to the owner.  
Primary key fingerprint: F6DD E01C 869F D639 15E5 5742 DEBD C156 F5AE BC52
```

Jika output berisi frasa `Good signature from "AWSTOE <awstoe@amazon.com>"`, itu berarti tanda tangan telah berhasil diverifikasi, dan Anda dapat melanjutkan untuk menjalankan skrip AWSTOE instalasi.

Jika output berisi frasa `BAD signature`, periksa apakah Anda melakukan prosedur dengan benar. Jika Anda terus mendapatkan respons ini, jangan jalankan file instalasi yang Anda unduh sebelumnya, dan hubungi AWS Support.

Berikut ini adalah informasi peringatan yang mungkin Anda lihat:

- **PERINGATAN:** Kunci ini tidak disertifikasi dengan tanda tangan tepercaya! Tidak ada indikasi bahwa tanda tangan ini adalah milik pemiliknya. Idealnya, Anda akan mengunjungi AWS kantor dan menerima kunci secara langsung. Namun, kemungkinan besar Anda akan mengunduhnya dari situs web. Dalam hal ini, situs web adalah AWS situs web.

- gpg: tidak ada kunci tepercaya yang ditemukan. Ini berarti bahwa kunci spesifik tidak “pada akhirnya dipercaya” oleh Anda, atau oleh orang lain yang Anda percayai.

Untuk informasi lebih lanjut, lihat <http://www.gnupg.org>.

Verifikasi tanda tangan unduhan AWSTOE instalasi di Windows

Topik ini menjelaskan proses yang direkomendasikan untuk memverifikasi validitas file instalasi untuk AWS Task Orchestrator and Executor aplikasi pada sistem operasi berbasis Windows.

Saat Anda mengunduh aplikasi dari internet, kami merekomendasikan Anda untuk mengautentikasi identitas penerbit perangkat lunak dan memeriksa apakah aplikasi tersebut belum diubah atau rusak setelah diterbitkan. Hal ini akan melindungi Anda agar tidak menginstal versi aplikasi yang berisi virus atau kode berbahaya lainnya.

Jika, setelah menjalankan langkah-langkah dalam topik ini, Anda menentukan bahwa perangkat lunak untuk AWSTOE aplikasi diubah atau rusak, jangan jalankan file instalasi. Sebaliknya, hubungi AWS Support.

Untuk memverifikasi validitas biner awstoe yang diunduh pada sistem operasi berbasis Windows, pastikan bahwa cap jempol sertifikat penandatanganan Amazon Services LLC sama dengan nilai ini:

5B 77 F4 F0 C3 7A 8B 89 D9 A7 8F 54 B6 85 11 CE 9E A3 BF 17

Note

Kami saat ini meluncurkan biner baru. Jika sertifikat penandatanganan Anda tidak cocok dengan sidik jari baru, verifikasi bahwa nilai sidik jari adalah:
FC B6 AC 0D 24 E6 90 1D 35 9D 69 38 4C DF 31 04 A7 3E 91 7A

Untuk memverifikasi nilai ini, lakukan prosedur berikut:

1. Klik kanan `awstoe.exe` yang diunduh, dan buka jendela Properti.
2. Pilih tab Tanda Tangan Digital.
3. Dari Daftar Tanda Tangan, pilih Amazon Services LLC, lalu pilih Detail.
4. Pilih tab Umum, jika belum dipilih, lalu pilih Lihat Sertifikat.
5. Pilih tab Detail, dan kemudian pilih Semua di Tampilkan daftar tarik turun, jika belum dipilih.

6. Gulir ke bawah sampai Anda melihat bidang Cap Jempol lalu pilih Cap Jempol. Ini menampilkan seluruh nilai cap jempol di jendela bawah.

- Jika nilai cap jempol di jendela bawah identik dengan nilai berikut:

5B 77 F4 F0 C3 7A 8B 89 D9 A7 8F 54 B6 85 11 CE 9E A3 BF 17

maka AWSTOE biner yang Anda unduh adalah otentik dan dapat diinstal dengan aman.

Note

Kami saat ini meluncurkan biner baru. Jika sertifikat penandatanganan Anda tidak cocok dengan sidik jari baru, verifikasi bahwa nilai sidik jari adalah:

FC B6 AC 0D 24 E6 90 1D 35 9D 69 38 4C DF 31 04 A7 3E 91 7A

- Jika nilai sidik jari di jendela detail bawah tidak identik dengan nilai sebelumnya, jangan jalankan `awstoe.exe`

Memulai langkah

- [Langkah 1: Instal AWSTOE](#)
- [Langkah 2: Tetapkan AWS kredensial](#)
- [Langkah 3: Kembangkan dokumen komponen secara lokal](#)
- [Langkah 4: Validasi komponen AWSTOE](#)
- [Langkah 5: Jalankan AWSTOE komponen](#)

Langkah 1: Instal AWSTOE

Untuk mengembangkan komponen secara lokal, unduh dan instal AWSTOE aplikasi.

1. Unduh AWSTOE aplikasinya

Untuk menginstal AWSTOE, pilih tautan unduhan yang sesuai untuk arsitektur dan platform Anda. Untuk daftar lengkap tautan unduhan aplikasi, lihat [AWSTOE unduh](#)

⚠ Important

AWS menghapus dukungan untuk TLS versi 1.0 dan 1.1 secara bertahap. Untuk mengakses bucket S3 untuk AWSTOE unduhan, perangkat lunak klien Anda harus menggunakan TLS versi 1.2 atau yang lebih baru. Untuk informasi lebih lanjut, lihat [posting Blog AWS Keamanan](#) ini.

2. Verifikasi tanda tangan

Langkah-langkah untuk memverifikasi unduhan Anda bergantung pada platform server tempat Anda menjalankan AWSTOE aplikasi setelah Anda menginstalnya. Untuk memverifikasi unduhan Anda di server Linux, lihat [Verifikasi tanda tangan di Linux](#). Untuk memverifikasi unduhan Anda di server Windows, lihat [Verifikasi tanda tangan di Windows](#).

⚠ Important

AWSTOE dipanggil langsung dari lokasi unduhannya. Tidak perlu langkah instalasi terpisah. Ini juga berarti bahwa AWSTOE dapat membuat perubahan pada lingkungan lokal. Untuk memastikan bahwa Anda mengisolasi perubahan selama pengembangan komponen, sebaiknya gunakan instans EC2 untuk mengembangkan dan menguji AWSTOE komponen.

Langkah 2: Tetapkan AWS kredensial

AWSTOE memerlukan AWS kredensial untuk terhubung ke yang lain Layanan AWS, seperti Amazon S3 dan CloudWatch Amazon, saat menjalankan tugas, seperti:

- Mengunduh AWSTOE dokumen dari jalur Amazon S3 yang disediakan pengguna.
- Modul lari `S3Download` atau `S3Upload` tindakan.
- Streaming log ke CloudWatch, saat diaktifkan.

Jika Anda menjalankan AWSTOE instans EC2, maka menjalankan AWSTOE menggunakan izin yang sama dengan peran IAM yang dilampirkan ke instans EC2.

Untuk informasi selengkapnya tentang peran IAM untuk EC2, lihat [peran IAM untuk Amazon EC2](#).

Contoh berikut menunjukkan cara mengatur AWS kredensial menggunakan variabel `AWS_ACCESS_KEY_ID` dan `AWS_SECRET_ACCESS_KEY` lingkungan.

Untuk mengatur variabel ini di Linux, macOS, atau Unix, gunakan `export`

```
$ export AWS_ACCESS_KEY_ID=your_access_key_id
```

```
$ export AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

Untuk mengatur variabel-variabel ini pada Windows menggunakan PowerShell, gunakan `$env`.

```
C:\> $env:AWS_ACCESS_KEY_ID=your_access_key_id
```

```
C:\> $env:AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

Untuk mengatur variabel-variabel ini pada Windows menggunakan command prompt, gunakan `set`.

```
C:\> set AWS_ACCESS_KEY_ID=your_access_key_id
```

```
C:\> set AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

Langkah 3: Kembangkan dokumen komponen secara lokal

AWSTOEkomponen ditulis dengan dokumen YAMAL plaintext. Untuk informasi selengkapnya tentang sintaks dokumen, lihat [Gunakan dokumen komponen di AWSTOE](#).

Berikut ini adalah contoh dokumen komponen Hello World yang dapat Anda gunakan untuk mengembangkan dokumen Anda secara lokal.

`hello-world-windows.yml`.

```
name: Hello World
description: This is Hello World testing document for Windows.
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: HelloWorldStep
```

```
    action: ExecutePowerShell
    inputs:
      commands:
        - Write-Host 'Hello World from the build phase.'
- name: validate
  steps:
    - name: HelloWorldStep
      action: ExecutePowerShell
      inputs:
        commands:
          - Write-Host 'Hello World from the validate phase.'
- name: test
  steps:
    - name: HelloWorldStep
      action: ExecutePowerShell
      inputs:
        commands:
          - Write-Host 'Hello World from the test phase.'
```

hello-world-linux.yml.

```
name: Hello World
description: This is hello world testing document for Linux.
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: HelloWorldStep
        action: ExecuteBash
        inputs:
          commands:
            - echo 'Hello World from the build phase.'
  - name: validate
    steps:
      - name: HelloWorldStep
        action: ExecuteBash
        inputs:
          commands:
            - echo 'Hello World from the validate phase.'
  - name: test
    steps:
      - name: HelloWorldStep
        action: ExecuteBash
```



```
inputs:
  commands:
    - echo 'Hello World from the test phase.'
```

Langkah 4: Validasi komponen AWSTOE

Anda dapat memvalidasi sintaks AWSTOE komponen secara lokal dengan aplikasi. AWSTOE Contoh berikut menunjukkan `validate` perintah AWSTOE aplikasi untuk memvalidasi sintaks komponen tanpa menjalankannya.

Note

AWSTOE aplikasi hanya dapat memvalidasi sintaks komponen untuk sistem operasi saat ini. Misalnya, saat berjalan `awstoe.exe` di Windows, Anda tidak dapat memvalidasi sintaks untuk dokumen Linux yang menggunakan modul `ExecuteBash` tindakan.

Windows

```
C:\> awstoe.exe validate --documents C:\Users\user\Documents\hello-world.yml
```

Linux:

```
$ awstoe validate --documents /home/user/hello-world.yml
```

Langkah 5: Jalankan AWSTOE komponen

AWSTOE aplikasi dapat menjalankan satu atau lebih fase dokumen tertentu menggunakan argumen baris `--phases` perintah. Nilai yang didukung untuk `--phases` adalah `build`, `validate`, dan `test`. Beberapa nilai fase dapat dimasukkan sebagai nilai yang dipisahkan koma.

Saat Anda memberikan daftar fase, AWSTOE aplikasi secara berurutan menjalankan fase yang ditentukan dari setiap dokumen. Misalnya, AWSTOE menjalankan `build` dan `validate` fase `document1.yml`, diikuti oleh `build` dan `validate` fase `document2.yml`.

Untuk memastikan bahwa log Anda disimpan dengan aman dan disimpan untuk pemecahan masalah, sebaiknya konfigurasi penyimpanan log di Amazon S3. Di Image Builder, lokasi Amazon

S3 untuk menerbitkan log ditentukan dalam konfigurasi infrastruktur. Untuk informasi selengkapnya tentang konfigurasi infrastruktur, lihat [Kelola konfigurasi infrastruktur EC2 Image Builder](#)

Jika daftar fase tidak disediakan, AWSTOE aplikasi menjalankan semua fase dalam urutan yang tercantum dalam dokumen YAMAL.

Untuk menjalankan fase tertentu dalam satu atau beberapa dokumen, gunakan perintah berikut.

Fase tunggal

```
awstoe run --documents hello-world.yml --phases build
```

Beberapa fase

```
awstoe run --documents hello-world.yml --phases build,test
```

Jalankan dokumen

Jalankan semua fase dalam satu dokumen

```
awstoe run --documents documentName.yaml
```

Jalankan semua fase dalam beberapa dokumen

```
awstoe run --documents documentName1.yaml,documentName2.yaml
```

Masukkan informasi Amazon S3 untuk mengunggah AWSTOE log dari jalur lokal yang ditentukan pengguna (disarankan)

```
awstoe run --documents documentName.yaml --log-s3-bucket-name <S3Bucket> --log-s3-key-prefix <S3KeyPrefix> --log-s3-bucket-owner <S3BucketOwner> --log-directory <local_path>
```

Jalankan semua fase dalam satu dokumen, dan tampilkan semua log di konsol

```
awstoe run --documents documentName.yaml --trace
```

Perintah contoh

```
awstoe run --documents s3://bucket/key/doc.yaml --phases build,validate
```

Jalankan dokumen dengan ID unik

```
awstoe run --documents <documentName>.yaml --execution-id <user provided id> --phases  
<comma separated list of phases>
```

Dapatkan bantuan dengan AWSTOE

```
awstoe --help
```

Gunakan dokumen komponen di AWSTOE

Untuk membuat komponen menggunakan AWS Task Orchestrator and Executor (AWSTOE), Anda harus menyediakan dokumen berbasis YAML yang mewakili fase dan langkah yang berlaku untuk komponen yang Anda buat. Layanan AWS menggunakan komponen Anda saat mereka membuat Amazon Machine Image (AMI) atau image container baru.

Topik

- [Alur kerja dokumen komponen](#)
- [Pencatatan komponen](#)
- [Rantai input dan output](#)
- [Skema dan definisi dokumen](#)
- [Skema contoh dokumen](#)
- [Mendefinisikan dan mereferensikan variabel dalam AWSTOE](#)
- [Gunakan konstruksi perulangan di AWSTOE](#)

Alur kerja dokumen komponen

Dokumen AWSTOE komponen menggunakan fase dan langkah untuk mengelompokkan tugas terkait, dan mengatur tugas-tugas tersebut ke dalam alur kerja logis untuk komponen.

Tip

Layanan yang menggunakan komponen Anda untuk membuat image mungkin menerapkan aturan tentang fase apa yang akan digunakan untuk proses pembuatannya, dan kapan

fase tersebut diizinkan untuk dijalankan. Ini penting untuk dipertimbangkan ketika Anda mendesain komponen Anda.

Fase

Fase mewakili perkembangan alur kerja Anda melalui proses pembuatan gambar. Misalnya, layanan Image Builder menggunakan `build` dan melakukan `validate` fase selama tahap `build` untuk gambar yang dihasilkannya. Ini menggunakan `container-host-test` fase `test` dan selama tahap pengujian untuk memastikan bahwa snapshot gambar atau gambar kontainer menghasilkan hasil yang diharapkan sebelum membuat AMI akhir atau mendistribusikan gambar kontainer.

Ketika komponen berjalan, perintah terkait untuk setiap fase diterapkan dalam urutan yang muncul dalam dokumen komponen.

Aturan untuk fase

- Setiap nama fase harus unik dalam dokumen.
- Anda dapat menentukan banyak fase dalam dokumen Anda.
- Anda harus menyertakan setidaknya satu dari tahapan berikut dalam dokumen Anda:
 - `build` — untuk Image Builder, fase ini umumnya digunakan selama tahap `build`.
 - `validate` — untuk Image Builder, fase ini umumnya digunakan selama tahap `build`.
 - `test` — untuk Image Builder, fase ini umumnya digunakan selama tahap pengujian.
- Fase selalu berjalan dalam urutan yang didefinisikan dalam dokumen. Urutan di mana mereka ditentukan untuk AWSTOE perintah di tidak AWS CLI berpengaruh.

Langkah-langkah

Langkah-langkah adalah unit kerja individu yang menentukan alur kerja dalam setiap fase. Langkah-langkah berjalan secara berurutan. Namun, input atau output untuk satu langkah juga dapat dimasukkan ke langkah berikutnya sebagai input. Ini disebut “rantai”.

Aturan untuk langkah-langkah

- Nama langkah harus unik untuk fase tersebut.
- Langkah harus menggunakan tindakan yang didukung (modul tindakan) yang mengembalikan kode keluar.

Untuk daftar lengkap modul tindakan yang didukung, cara kerjanya, nilai input/output, dan contoh, lihat [Modul aksi didukung oleh manajer AWSTOE komponen](#)

Pencatatan komponen

AWSTOE membuat folder log baru pada instance EC2 yang digunakan untuk membangun dan menguji gambar baru, setiap kali komponen Anda berjalan. Untuk gambar kontainer, folder log disimpan dalam wadah.

Untuk membantu pemecahan masalah jika terjadi kesalahan selama proses pembuatan gambar, dokumen input dan semua file output yang AWSTOE dibuat saat menjalankan komponen disimpan di folder log.

Nama folder log terdiri dari bagian-bagian berikut:

1. Direktori log — ketika layanan menjalankan AWSTOE komponen, ia melewati direktori log, bersama dengan pengaturan lain untuk perintah. Untuk contoh berikut, kami menunjukkan format file log yang digunakan Image Builder.
 - Linux: `/var/lib/amazon/toe/`
 - Windows: `$env:ProgramFiles\Amazon\TaskOrchestratorAndExecutor\`
2. Awalan file — Ini adalah awalan standar yang digunakan untuk semua komponen: `""TOE_.`
3. Run time - Ini adalah stempel waktu dalam format `YYYY-MM-DD_HH-MM-SS_UTC-0`.
4. Execution ID — Ini adalah GUID yang ditetapkan ketika AWSTOE menjalankan satu atau beberapa komponen.

Contoh: `/var/lib/amazon/toe/TOE_2021-07-01_12-34-56_UTC-0_a1bcd2e3-45f6-789a-bcde-0fa1b2c3def4`

AWSTOE menyimpan file inti berikut di folder log:

Berkas masukan

- `document.yaml` — Dokumen yang digunakan sebagai input untuk perintah. Setelah komponen berjalan, file ini disimpan sebagai artefak.

Berkas keluaran

- `application.log` - Log aplikasi berisi informasi tingkat debug stempel waktu dari AWSTOE tentang apa yang terjadi saat komponen sedang berjalan.
- `detailedoutput.json` — File JSON ini memiliki informasi rinci tentang status run, input, output, dan kegagalan untuk semua dokumen, fase, dan langkah-langkah yang berlaku untuk komponen saat berjalan.
- `console.log` - Log konsol berisi semua informasi standar keluar (`stdout`) dan kesalahan standar (`stderr`) yang AWSTOE menulis ke konsol saat komponen sedang berjalan.
- `chaining.json` - File JSON ini mewakili pengoptimalan yang diterapkan untuk menyelesaikan ekspresi rantai. AWSTOE

Note

Folder log mungkin juga berisi file sementara lainnya yang tidak tercakup di sini.

Rantai input dan output

Aplikasi manajemen AWSTOE konfigurasi menyediakan fitur untuk merantai input dan output dengan menulis referensi dalam format berikut:

```
{{ phase_name.step_name.inputs/outputs.variable }}
```

atau

```
{{ phase_name.step_name.inputs/outputs[index].variable }}
```

Fitur chaining memungkinkan Anda untuk mendaur ulang kode dan meningkatkan pemeliharaan dokumen.

Aturan untuk rantai

- Ekspresi rantai hanya dapat digunakan di bagian input dari setiap langkah.
- Pernyataan dengan ekspresi rantai harus dilampirkan dalam tanda kutip. Sebagai contoh:
 - Ekspresi tidak valid: `echo {{ phase.step.inputs.variable }}`
 - Ekspresi yang valid: `"echo {{ phase.step.inputs.variable }}"`
 - Ekspresi yang valid: `'echo {{ phase.step.inputs.variable }}'`

- Ekspresi rantai dapat mereferensikan variabel dari langkah dan fase lain dalam dokumen yang sama. Namun, layanan panggilan mungkin memiliki aturan yang memerlukan ekspresi rantai untuk beroperasi hanya dalam konteks satu tahap. Misalnya, Image Builder tidak mendukung chaining dari tahap build ke tahap pengujian, karena Image Builder menjalankan setiap tahap secara independen.
- Indeks dalam ekspresi rantai mengikuti pengindeksan berbasis nol. Indeks dimulai dengan nol (0) untuk referensi elemen pertama.

Contoh

Untuk merujuk ke variabel sumber pada entri kedua dari langkah contoh berikut, pola rantai adalah `{{ build.SampleS3Download.inputs[1].source }}`.

```
phases:
-
  name: 'build'
  steps:
  -
    name: SampleS3Download
    action: S3Download
    timeoutSeconds: 60
    onFailure: Abort
    maxAttempts: 3
    inputs:
    -
      source: 's3://sample-bucket/sample1.ps1'
      destination: 'C:\sample1.ps1'
    -
      source: 's3://sample-bucket/sample2.ps1'
      destination: 'C:\sample2.ps1'
```

Untuk merujuk ke variabel output (sama dengan “Halo”) dari langkah contoh berikut, pola rantai adalah `{{ build.SamplePowerShellStep.outputs.stdout }}`.

```
phases:
-
  name: 'build'
  steps:
  -
    name: SamplePowerShellStep
    action: ExecutePowerShell
```

```

timeoutSeconds: 120
onFailure: Abort
maxAttempts: 3
inputs:
  commands:
    - 'Write-Host "Hello"'

```

Skema dan definisi dokumen

Berikut ini adalah skema YAMMM untuk dokumen.

```

name: (optional)
description: (optional)
schemaVersion: "string"

phases:
  - name: "string"
    steps:
      - name: "string"
        action: "string"
        timeoutSeconds: integer
        onFailure: "Abort|Continue|Ignore"
        maxAttempts: integer
        inputs:

```

Definisi skema untuk dokumen adalah sebagai berikut.

Bidang	Deskripsi	Tipe	Diperlukan
nama	Nama dokumen.	String	Tidak
deskripsi	Deskripsi dokumen.	String	Tidak
schemaVersion	Versi skema dokumen, saat ini 1.0.	Tali	Ya
fase	Daftar fase dengan langkah-langkahnya.	Daftar	Ya

Definisi skema untuk suatu fase adalah sebagai berikut.

Bidang	Deskripsi	Tipe	Diperlukan
nama	Nama fase.	Tali	Ya
langkah	Daftar langkah-l angkah dalam fase.	Daftar	Ya

Definisi skema untuk suatu langkah adalah sebagai berikut.

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default
nama	Nama yang ditentukan pengguna untuk langkah tersebut.	String		
tindakan	Kata kunci yang berkaitan dengan modul yang menjalankan langkah.	String		
timeoutSeconds	Jumlah detik langkah berjalan sebelum gagal atau mencoba lagi. Juga, mendukung -1 nilai, yang menunjukkan batas waktu tak terbatas. 0 dan nilai negatif lainnya tidak diperbolehkan.	Bulat	Tidak	7.200 detik (120 menit)

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default
onFailure	<p>Menentukan apa yang harus dilakukan langkah jika terjadi kegagalan. Nilai yang valid adalah sebagai berikut:</p> <ul style="list-style-type: none"> • Batalkan — Gagal langkah setelah jumlah upaya maksimum, dan berhenti berjalan. Menetapkan status untuk fase dan dokumen keFailed. • Lanjutkan — Gagal langkah setelah jumlah upaya maksimum, dan terus menjalankan langkah-langkah yang tersisa. Menetapkan status untuk fase dan 	String	Tidak	Batalkan

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default
	<p>dokumen keFailed.</p> <ul style="list-style-type: none"> • Abaikan - Tetapkan langkah ke IgnoredFailure setelah jumlah maksimum upaya yang gagal, dan terus menjalankan langkah-langkah yang tersisa. Menetapkan status untuk fase dan dokumen keSuccessWithIgnoredFailure . 			
maxAttempts	Jumlah maksimum upaya yang diizinkan sebelum gagal langkah.	Bulat	Tidak	1

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default
masukan	Berisi parameter yang diperlukan oleh modul tindakan untuk menjalankan langkah.	Dikte	Ya	

Skema contoh dokumen

Berikut ini adalah contoh skema dokumen untuk menginstal semua pembaruan Windows yang tersedia, menjalankan skrip konfigurasi, memvalidasi perubahan sebelum AMI dibuat, dan menguji perubahan setelah AMI dibuat.

```

name: RunConfig_UpdateWindows
description: 'This document will install all available Windows updates and run a config script. It will then validate the changes before an AMI is created. Then after AMI creation, it will test all the changes.'
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: DownloadConfigScript
        action: S3Download
        timeoutSeconds: 60
        onFailure: Abort
        maxAttempts: 3
        inputs:
          - source: 's3://customer-bucket/config.ps1'
            destination: 'C:\config.ps1'

      - name: RunConfigScript
        action: ExecutePowerShell
        timeoutSeconds: 120
        onFailure: Abort
        maxAttempts: 3
        inputs:
          file: '{{build.DownloadConfigScript.inputs[0].destination}}'

```

```
- name: Cleanup
  action: DeleteFile
  onFailure: Abort
  maxAttempts: 3
  inputs:
    - path: '{{build.DownloadConfigScript.inputs[0].destination}}'

- name: RebootAfterConfigApplied
  action: Reboot
  inputs:
    delaySeconds: 60

- name: InstallWindowsUpdates
  action: UpdateOS

- name: validate
  steps:
    - name: DownloadTestConfigScript
      action: S3Download
      timeoutSeconds: 60
      onFailure: Abort
      maxAttempts: 3
      inputs:
        - source: 's3://customer-bucket/testConfig.ps1'
          destination: 'C:\testConfig.ps1'

    - name: ValidateConfigScript
      action: ExecutePowerShell
      timeoutSeconds: 120
      onFailure: Abort
      maxAttempts: 3
      inputs:
        file: '{{validate.DownloadTestConfigScript.inputs[0].destination}}'

    - name: Cleanup
      action: DeleteFile
      onFailure: Abort
      maxAttempts: 3
      inputs:
        - path: '{{validate.DownloadTestConfigScript.inputs[0].destination}}'

- name: test
  steps:
    - name: DownloadTestConfigScript
```

```

    action: S3Download
    timeoutSeconds: 60
    onFailure: Abort
    maxAttempts: 3
    inputs:
      - source: 's3://customer-bucket/testConfig.ps1'
        destination: 'C:\testConfig.ps1'

- name: ValidateConfigScript
  action: ExecutePowerShell
  timeoutSeconds: 120
  onFailure: Abort
  maxAttempts: 3
  inputs:
    file: '{{test.DownloadTestConfigScript.inputs[0].destination}}'

```

Berikut ini adalah contoh skema dokumen untuk men-download dan menjalankan file biner Linux kustom.

```

name: LinuxBin
description: Download and run a custom Linux binary file.
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: Download
        action: S3Download
        inputs:
          - source: s3://<replaceable>mybucket</replaceable>/
            <replaceable>myapplication</replaceable>
            destination: /tmp/<replaceable>myapplication</replaceable>
      - name: Enable
        action: ExecuteBash
        onFailure: Continue
        inputs:
          commands:
            - 'chmod u+x {{ build.Download.inputs[0].destination }}'
      - name: Install
        action: ExecuteBinary
        onFailure: Continue
        inputs:
          path: '{{ build.Download.inputs[0].destination }}'
          arguments:

```

```

    - '--install'
  - name: Delete
    action: DeleteFile
    inputs:
      - path: '{{ build.Download.inputs[0].destination }}'

```

Berikut ini adalah contoh skema dokumen untuk menginstal AWS CLI pada instance Windows, menggunakan file setup.

```

name: InstallCLISetUp
description: Install &CLI; using the setup file
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: Download
        action: S3Download
        inputs:
          - source: s3://aws-cli/AWSCLISetup.exe
            destination: C:\Windows\temp\AWSCLISetup.exe
      - name: Install
        action: ExecuteBinary
        onFailure: Continue
        inputs:
          path: '{{ build.Download.inputs[0].destination }}'
          arguments:
            - '/install'
            - '/quiet'
            - '/norestart'
      - name: Delete
        action: DeleteFile
        inputs:
          - path: '{{ build.Download.inputs[0].destination }}'

```

Berikut ini adalah contoh skema dokumen untuk menginstal AWS CLI menggunakan installer MSI.

```

name: InstallCLIMSI
description: Install &CLI; using the MSI installer
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: Download

```

```
    action: S3Download
    inputs:
      - source: s3://aws-cli/AWSCLI64PY3.msi
        destination: C:\Windows\temp\AWSCLI64PY3.msi
  - name: Install
    action: ExecuteBinary
    onFailure: Continue
    inputs:
      path: 'C:\Windows\System32\msiexec.exe'
      arguments:
        - '/i'
        - '{{ build.Download.inputs[0].destination }}'
        - '/quiet'
        - '/norestart'
  - name: Delete
    action: DeleteFile
    inputs:
      - path: '{{ build.Download.inputs[0].destination }}'
```

Mendefinisikan dan mereferensikan variabel dalam AWSTOE

Variabel menyediakan cara untuk memberi label data dengan nama yang bermakna yang dapat digunakan di seluruh aplikasi. Anda dapat menentukan variabel kustom dengan format sederhana dan mudah dibaca untuk alur kerja yang kompleks, dan mereferensikannya dalam dokumen komponen aplikasi YAMAL untuk komponen. AWSTOE

Bagian ini menyediakan informasi untuk membantu Anda menentukan variabel untuk AWSTOE komponen Anda dalam dokumen komponen aplikasi YAMAL, termasuk sintaks, batasan nama, dan contoh.

Parameter

Parameter adalah variabel yang dapat berubah, dengan pengaturan yang dapat disediakan aplikasi panggilan saat runtime. Anda dapat menentukan parameter di `Parameters` bagian dokumen YAMB.

Aturan untuk nama parameter

- Nama harus antara 3 dan 128 karakter panjangnya.
- Nama hanya dapat berisi karakter alfanumerik (a-z, A-Z, 0-9), tanda hubung (-), atau garis bawah (_).
- Nama harus unik di dalam dokumen.

- Nama harus ditentukan sebagai string YAMAL.

Sintaksis

```
parameters:
  - <name>:
    type: <parameter type>
    default: <parameter value>
    description: <parameter description>
```

Nama kunci	Diperlukan	Deskripsi
name	Ya	Nama parameter. Harus unik untuk dokumen (tidak boleh sama dengan nama parameter atau konstanta lainnya).
type	Ya	Tipe data parameter. Jenis yang didukung meliputi: <code>string</code> .
default	Tidak	Nilai default untuk parameter.
description	Tidak	Menjelaskan parameter.

Nilai parameter referensi dalam dokumen

Anda dapat mereferensikan parameter dalam input langkah atau loop di dalam dokumen YAMAL. Anda, sebagai berikut:

- Referensi parameter peka huruf besar/kecil, dan namanya harus sama persis.
- Nama harus dilampirkan dalam kawat gigi keriting ganda. `{{ MyParameter }}`
- Spasi diperbolehkan dalam kurung kurawal, dan secara otomatis dipangkas. Misalnya, semua referensi berikut valid:

```
{{ MyParameter }}, {{ MyParameter }}, {{MyParameter }}, {{MyParameter}}
```

- Referensi dalam dokumen YAMB harus ditentukan sebagai string (terlampir dalam tanda kutip tunggal atau ganda).

Misalnya: - `{{ MyParameter }}` tidak valid, karena tidak diidentifikasi sebagai string.

Namun, referensi berikut ini valid: - `'{{ MyParameter }}'` dan- `"{{ MyParameter }}"`.

Contoh

Contoh berikut menunjukkan cara menggunakan parameter dalam dokumen YAMB Anda:

- Lihat parameter dalam input langkah:

```
name: Download AWS CLI version 2
schemaVersion: 1.0
parameters:
  - Source:
    type: string
    default: 'https://awscli.amazonaws.com/AWSCLIV2.msi'
    description: The AWS CLI installer source URL.
phases:
  - name: build
    steps:
      - name: Download
        action: WebDownload
        inputs:
          - source: '{{ Source }}'
            destination: 'C:\Windows\Temp\AWSCLIV2.msi'
```

- Lihat parameter dalam input loop:

```
name: PingHosts
schemaVersion: 1.0
parameters:
  - Hosts:
    type: string
    default: 127.0.0.1,amazon.com
    description: A comma separated list of hosts to ping.
phases:
  - name: build
    steps:
      - name: Ping
```

```
action: ExecuteBash
loop:
  forEach:
    list: '{{ Hosts }}'
    delimiter: ','
inputs:
  commands:
    - ping -c 4 {{ loop.value }}
```

Ganti parameter saat runtime

Anda dapat menggunakan `--parameters` opsi dari AWS CLI with a key-value pair untuk menetapkan nilai parameter saat runtime.

- `<name><value>` Tentukan pasangan kunci-nilai parameter sebagai nama dan nilai, dipisahkan oleh tanda sama dengan (=).
- Beberapa parameter harus dipisahkan dengan koma.
- Nama parameter yang tidak ditemukan dalam dokumen komponen YAMB diabaikan.
- Nama parameter dan nilai keduanya diperlukan.

Important

Parameter komponen adalah nilai teks biasa, dan masuk AWS CloudTrail. Kami menyarankan Anda menggunakan AWS Secrets Manager atau AWS Systems Manager Parameter Store untuk menyimpan rahasia Anda. Untuk informasi selengkapnya tentang Secrets Manager, lihat [Apa itu Secrets Manager?](#) dalam AWS Secrets Manager User Guide. Untuk informasi selengkapnya tentang AWS Systems Manager Parameter Store, lihat [AWS Systems Manager Parameter Store](#) di Panduan AWS Systems Manager Pengguna.

Sintaksis

```
--parameters name1=value1,name2=value2...
```

Opsi CLI	Diperlukan	Deskripsi
<code>--parameter <i>nama</i> = <i>nilai</i>,...</code>	Tidak	Opsi ini mengambil daftar pasangan kunci-nilai, dengan nama parameter sebagai kunci.

Contoh

Contoh berikut menunjukkan cara menggunakan parameter dalam dokumen YAMB Anda:

- Pasangan nilai kunci parameter yang ditentukan dalam `--parameter` opsi ini tidak valid:

```
--parameters ntp-server=
```

- Tetapkan satu parameter pasangan kunci-nilai dengan `--parameter` opsi di: AWS CLI

```
--parameters ntp-server=ntp-server-windows-qe.us-east1.amazon.com
```

- Tetapkan beberapa pasangan nilai kunci parameter dengan `--parameter` opsi di: AWS CLI

```
--parameters ntp-server=ntp-server.amazon.com,http-url=https://internal-us-east1.amazon.com
```

Konstanta

Konstanta adalah variabel yang tidak dapat diubah yang tidak dapat dimodifikasi atau diganti setelah ditentukan. Konstanta dapat didefinisikan menggunakan nilai-nilai di `constants` bagian AWSTOE dokumen.

Aturan untuk nama konstan

- Nama harus antara 3 dan 128 karakter panjangnya.
- Nama hanya dapat berisi karakter alfanumerik (a-z, A-Z, 0-9), tanda hubung (-), atau garis bawah (_).
- Nama harus unik di dalam dokumen.
- Nama harus ditentukan sebagai string YAMAL.

Sintaks

```
constants:
  - <name>:
    type: <constant type>
    value: <constant value>
```

Nama kunci	Diperlukan	Deskripsi
name	Ya	Nama konstanta. Harus unik untuk dokumen (tidak boleh sama dengan nama parameter atau konstanta lainnya).
value	Ya	Nilai konstanta.
type	Ya	Jenis konstanta. Jenis yang didukung adalah <code>string</code> .

Referensi nilai konstan dalam dokumen

Anda dapat mereferensikan konstanta dalam input langkah atau loop di dalam dokumen YAMAL Anda, sebagai berikut:

- Referensi konstan peka huruf besar/kecil, dan namanya harus sama persis.
- Nama harus dilampirkan dalam kawat gigi keriting ganda. `{{ MyConstant }}`
- Spasi diperbolehkan dalam kurung kurawal, dan secara otomatis dipangkas. Misalnya, semua referensi berikut valid:

```
{{ MyConstant }}, {{ MyConstant }}, {{MyConstant }}, {{MyConstant}}
```

- Referensi dalam dokumen YAMAL harus ditentukan sebagai string (terlampir dalam tanda kutip tunggal atau ganda).

Misalnya - `{{ MyConstant }}` tidak valid, karena tidak diidentifikasi sebagai string.

Namun, referensi berikut ini valid: - `'{{ MyConstant }{' dan - "{{ MyConstant }}".`

Contoh

Konstan direferensikan dalam input langkah

```
name: Download AWS CLI version 2
schemaVersion: 1.0
constants:
  - Source:
      type: string
      value: https://awscli.amazonaws.com/AWSCLIV2.msi
phases:
  - name: build
    steps:
      - name: Download
        action: WebDownload
        inputs:
          - source: '{{ Source }}'
            destination: 'C:\Windows\Temp\AWSCLIV2.msi'
```

Konstan direferensikan dalam input loop

```
name: PingHosts
schemaVersion: 1.0
constants:
  - Hosts:
      type: string
      value: 127.0.0.1,amazon.com
phases:
  - name: build
    steps:
      - name: Ping
        action: ExecuteBash
        loop:
          forEach:
            list: '{{ Hosts }}'
            delimiter: ','
        inputs:
          commands:
            - ping -c 4 {{ loop.value }}
```

Gunakan konstruksi perulangan di AWSTOE

Bagian ini memberikan informasi untuk membantu Anda membuat konstruksi perulangan di AWSTOE. Konstruksi perulangan mendefinisikan urutan instruksi yang berulang. Anda dapat menggunakan jenis konstruksi perulangan berikut di AWSTOE:

- `for` konstruksi - Iterasi atas urutan bilangan bulat yang dibatasi.
- `forEach` membangun
 - `forEachLoop` dengan daftar masukan - Iterasi atas kumpulan string yang terbatas.
 - `forEachLoop` dengan daftar terbatas - Iterasi di atas kumpulan string terbatas yang bergabung dengan pembatas.

Note

Konstruksi perulangan hanya mendukung tipe data string.

Topik konstruksi perulangan

- [Variabel iterasi referensi](#)
- [Jenis konstruksi perulangan](#)
- [Bidang langkah](#)
- [Output langkah dan iterasi](#)

Variabel iterasi referensi

Untuk merujuk pada indeks dan nilai variabel iterasi saat ini, ekspresi referensi `{{ loop.* }}` harus digunakan dalam badan input dari langkah yang berisi konstruksi perulangan. Ekspresi ini tidak dapat digunakan untuk merujuk ke variabel iterasi dari konstruksi perulangan dari langkah lain.

Ekspresi referensi terdiri dari anggota berikut:

- `{{ loop.index }}`— Posisi ordinal dari iterasi saat ini, yang diindeks di 0.
- `{{ loop.value }}`— Nilai yang terkait dengan variabel iterasi saat ini.

Nama loop

Semua konstruksi perulangan memiliki bidang nama opsional untuk identifikasi. Jika nama loop disediakan, dapat digunakan untuk merujuk ke variabel iterasi di badan input langkah. Untuk merujuk pada indeks iterasi dan nilai loop bernama, gunakan `{{ <loop_name>.* }}` dengan `{{ loop.* }}` di badan input langkah. Ekspresi ini tidak dapat digunakan untuk merujuk pada konstruksi perulangan bernama dari langkah lain.

Ekspresi referensi terdiri dari anggota berikut:

- `{{ <loop_name>.index }}`— Posisi ordinal dari iterasi saat ini dari loop bernama, yang diindeks di `0`
- `{{ <loop_name>.value }}`— Nilai yang terkait dengan variabel iterasi saat ini dari loop bernama.

Selesaikan ekspresi referensi

AWSTOEMengatasi ekspresi referensi sebagai berikut:

- `{{ <loop_name>.* }}`— AWSTOE menyelesaikan ekspresi ini menggunakan logika berikut:
 - Jika loop dari langkah yang sedang berjalan cocok dengan `<loop_name>` nilainya, maka ekspresi referensi menyelesaikan konstruksi perulangan dari langkah yang sedang berjalan.
 - `<loop_name>` menyelesaikan ke konstruksi perulangan bernama jika muncul dalam langkah yang sedang berjalan.
- `{{ loop.* }}`— AWSTOE menyelesaikan ekspresi menggunakan konstruksi perulangan yang ditentukan dalam langkah yang sedang berjalan.

Jika ekspresi referensi digunakan dalam langkah yang tidak mengandung loop, maka AWSTOE tidak menyelesaikan ekspresi dan mereka muncul di langkah tanpa penggantian.

Note

Ekspresi referensi harus diapit tanda kutip ganda untuk ditafsirkan dengan benar oleh kompiler YAMAL.

Jenis konstruksi perulangan

Bagian ini memberikan informasi dan contoh tentang jenis konstruksi perulangan yang dapat digunakan dalam AWSTOE

Jenis konstruksi perulangan

- [forlingkaran](#)
- [forEachloop dengan daftar masukan](#)
- [forEachloop dengan daftar yang dibatasi](#)

forlingkaran

forLoop iterasi pada rentang bilangan bulat yang ditentukan dalam batas yang diuraikan oleh awal dan akhir variabel. Nilai iterasi ada di set [start, end] dan termasuk nilai batas.

AWSTOEmemverifikasistart,end, dan updateBy nilai untuk memastikan bahwa kombinasi tidak menghasilkan loop tak terbatas.

for skema lingkaran

```
name: "StepName"
action: "ActionModule"
loop:
  name: "string"
  for:
    start: int
    end: int
    updateBy: int
inputs:
  ...
```

for masukan loop

Bidang	Deskripsi	Tipe	Diperlukan	Default
name	Nama unik dari loop. Itu harus unik dibandingkan dengan	String	Tidak	""

Bidang	Deskripsi	Tipe	Diperlukan	Default
	nama loop lain dalam fase yang sama.			
start	Nilai awal iterasi. Tidak menerima ekspresi rantai.	Bulat	Ya	T/A
end	Nilai akhir iterasi. Tidak menerima ekspresi rantai.	Bulat	Ya	T/A
updateBy	Perbedaan dimana nilai iterasi diperbarui melalui penambahan. Itu harus nilai negatif atau positif bukan nol. Tidak menerima ekspresi rantai.	Bulat	Ya	T/A

for contoh masukan loop

```

name: "CalculateFileUploadLatencies"
action: "ExecutePowerShell"
loop:
  for:
    start: 100000
    end: 1000000
    updateBy: 100000
inputs:
  commands:
    - |
      $f = new-object System.IO.FileStream c:\temp\test{{ loop.index }}.txt, Create,
      ReadWrite

```

```

    $f.SetLength({{ loop.value }}MB)
    $f.Close()
    - c:\users\administrator\downloads\latencyTest.exe --file c:\temp
\test{{ loop.index }}.txt
    - AWS s3 cp c:\users\administrator\downloads\latencyMetrics.json s3://bucket/
latencyMetrics.json
    - |
    Remove-Item -Path c:\temp\test{{ loop.index }}.txt
    Remove-Item -Path c:\users\administrator\downloads\latencyMetrics.json

```

forEachloop dengan daftar masukan

forEachLoop iterasi pada daftar eksplisit nilai, yang dapat berupa string dan ekspresi dirantai.

forEachloop dengan skema daftar masukan

```

name: "StepName"
action: "ActionModule"
loop:
  name: "string"
  forEach:
    - "string"
inputs:
  ...

```

forEachloop dengan masukan daftar masukan

Bidang	Deskripsi	Tipe	Diperlukan	Default
name	Nama unik dari loop. Itu harus unik dibandingkan dengan nama loop lain dalam fase yang sama.	String	Tidak	""
Daftar string loop forEach	Daftar string untuk iterasi. Menerima ekspresi berantai	Daftar string	Ya	T/A

Bidang	Deskripsi	Tipe	Diperlukan	Default
	<p>sebagai string dalam daftar. Ekspresi berantai harus diapit oleh tanda kutip ganda agar kompiler YAMAL dapat menafsirkannya dengan benar.</p>			

forEachloop dengan contoh daftar masukan 1

```

name: "ExecuteCustomScripts"
action: "ExecuteBash"
loop:
  name: BatchExecLoop
  forEach:
    - /tmp/script1.sh
    - /tmp/script2.sh
    - /tmp/script3.sh
inputs:
  commands:
    - echo "Count {{ BatchExecLoop.index }}"
    - sh "{{ loop.value }}"
    - |
      retVal=$?
      if [ $retVal -ne 0 ]; then
        echo "Failed"
      else
        echo "Passed"
      fi

```

forEachloop dengan contoh daftar masukan 2

```

name: "RunMSIWithDifferentArgs"
action: "ExecuteBinary"
loop:

```

```

name: MultiArgLoop
forEach:
  - "ARG1=C:\Users ARG2=1"
  - "ARG1=C:\Users"
  - "ARG1=C:\Users ARG3=C:\Users\Administrator\Documents\f1.txt"
inputs:
  commands:
    path: "c:\users\administrator\downloads\runner.exe"
    args:
      - "{{ MultiArgLoop.value }}"

```

forEachloop dengan contoh daftar masukan 3

```

name: "DownloadAllBinaries"
action: "S3Download"
loop:
  name: MultiArgLoop
  forEach:
    - "bin1.exe"
    - "bin10.exe"
    - "bin5.exe"
inputs:
  -
    source: "s3://bucket/{{ loop.value }}"
    destination: "c:\temp\{{ loop.value }}"

```

forEachloop dengan daftar yang dibatasi

Loop iterasi atas string yang berisi nilai-nilai yang dipisahkan oleh pembatas. Untuk mengulangi konstituen string, AWSTOE gunakan pembatas untuk membagi string menjadi array yang cocok untuk iterasi.

forEachloop dengan skema daftar yang dibatasi

```

name: "StepName"
action: "ActionModule"
loop:
  name: "string"
  forEach:
    list: "string"
    delimiter: ".,;:\n\t -_"
inputs:

```

...

forEachloop dengan input daftar yang dibatasi

Bidang	Deskripsi	Tipe	Diperlukan	Default
<code>name</code>	Nama unik yang diberikan ke loop. Ini harus unik jika dibandingkan dengan nama loop lain dalam fase yang sama.	String	Tidak	""
<code>list</code>	String yang terdiri dari string konstituen bergabung dengan karakter pembatas umum. Juga menerima ekspresi dirantai. Dalam kasus ekspresi berantai, pastikan bahwa ekspresi tersebut diapit oleh tanda kutip ganda untuk interpretasi yang benar oleh kompiler YAMAL.	Tali	Ya	T/A
<code>delimiter</code>	Karakter yang digunakan untuk	String	Tidak	Koma: ", "

Bidang	Deskripsi	Tipe	Diperlukan	Default
	<p>memisahkan string dalam blok. Default adalah karakter koma. Hanya satu karakter pembatas yang diizinkan dari daftar yang diberikan:</p> <ul style="list-style-type: none"> • Titik: "." • Koma: "," • Titik koma: ";" • Usus besar: ":" • Baris baru: "\n" • Tab: "\t" • Ruang: " " • Tanda hubung: "-" • Garis bawah: "_" <p>Ekspresi rantai tidak dapat digunakan.</p>			

Note

Nilai diperlakukan sebagai list string yang tidak dapat diubah. Jika sumber list diubah selama runtime, itu tidak akan tercermin selama menjalankan.

forEachloop dengan daftar yang dibatasi contoh 1

```
// Uses changing expression ({{ <phase_name>.<step_name>.inputs/outputs.<var_name> }})
// to refer to another step's input/output variables for code re-use.
name: "RunMSIs"
action: "ExecuteBinary"
loop:
  forEach:
    list: "{{ build.GetAllMSIPathsForInstallation.outputs.stdout }}"
    delimiter: "\n"
inputs:
  commands:
    path: "{{ loop.value }}"
```

forEachloop dengan daftar yang dibatasi contoh 2

```
name: "UploadMetricFiles"
action: "S3Upload"
loop:
  forEach:
    list: "/tmp/m1.txt,/tmp/m2.txt,/tmp/m3.txt,..."
inputs:
  commands:
    -
      source: "{{ loop.value }}"
      destination: "s3://bucket/key/{{ loop.value }}"
```

Bidang langkah

Loop adalah bagian dari langkah. Bidang apa pun yang terkait dengan menjalankan langkah tidak diterapkan pada iterasi individu. Bidang langkah hanya berlaku pada tingkat langkah, sebagai berikut:

- **TimeoutSeconds** - Semua iterasi loop harus dijalankan dalam jangka waktu yang ditentukan oleh bidang ini. Jika loop AWSTOE habis waktu, jalankan kebijakan coba lagi langkah tersebut dan atur ulang parameter batas waktu untuk setiap upaya baru. Jika loop run melebihi nilai batas waktu

setelah mencapai jumlah maksimum percobaan ulang, pesan kegagalan langkah menyatakan bahwa loop run telah habis waktunya.

- OnFailure - Penanganan kegagalan diterapkan pada langkah sebagai berikut:
 - Jika onFailure disetel keAbort, AWSTOE keluar dari loop dan mencoba ulang langkah sesuai dengan kebijakan coba lagi. Setelah jumlah maksimum percobaan ulang, AWSTOE tandai langkah saat ini sebagai gagal, dan berhenti menjalankan proses.

AWSTOEmenetapkan kode status untuk fase induk dan dokumen keFailed.

Note

Tidak ada langkah lebih lanjut yang berjalan setelah langkah gagal.

- Jika onFailure disetel keContinue, AWSTOE keluar dari loop dan mencoba ulang langkah sesuai dengan kebijakan coba lagi. Setelah jumlah maksimum percobaan ulang, AWSTOE tandai langkah saat ini sebagai gagal, dan lanjutkan untuk menjalankan langkah berikutnya.

AWSTOEmenetapkan kode status untuk fase induk dan dokumen keFailed.

- Jika onFailure disetel keIgnore, AWSTOE keluar dari loop dan mencoba ulang langkah sesuai dengan kebijakan coba lagi. Setelah jumlah maksimum percobaan ulang, AWSTOE tandai langkah saat ini sebagaiIgnoredFailure, dan lanjutkan untuk menjalankan langkah berikutnya.

AWSTOEmenetapkan kode status untuk fase induk dan dokumen keSuccessWithIgnoredFailure.

Note

Ini masih dianggap sebagai proses yang berhasil, tetapi termasuk informasi untuk memberi tahu Anda bahwa satu atau lebih langkah gagal dan diabaikan.

- MaxAttempts — Untuk setiap percobaan ulang, seluruh langkah dan semua iterasi dijalankan dari awal.
- status — Status keseluruhan dari menjalankan langkah. status tidak mewakili status iterasi individu. Status langkah dengan loop ditentukan sebagai berikut:
 - Jika iterasi tunggal gagal dijalankan, status langkah menunjuk ke kegagalan.
 - Jika semua iterasi berhasil, status langkah menunjukkan kesuksesan.

- **StartTime** — Waktu mulai keseluruhan dari menjalankan langkah. Tidak mewakili waktu mulai iterasi individu.
- **EndTime** — Waktu akhir keseluruhan dari menjalankan langkah. Tidak mewakili waktu akhir iterasi individu.
- **FailureMessage** — Termasuk indeks iterasi yang gagal jika terjadi kesalahan non-timeout. Jika terjadi kesalahan batas waktu, pesan menyatakan bahwa loop run telah gagal. Pesan kesalahan individual untuk setiap iterasi tidak disediakan untuk meminimalkan ukuran pesan kegagalan.

Output langkah dan iterasi

Setiap iterasi berisi output. Pada akhir loop run, AWSTOE mengkonsolidasikan semua output iterasi yang berhasil di `detailedOutput.json`. Output konsolidasi adalah pengumpulan nilai yang termasuk dalam kunci output yang sesuai seperti yang didefinisikan dalam skema output modul tindakan. Contoh berikut menunjukkan bagaimana output dikonsolidasikan:

Output dari **ExecuteBash** untuk Iterasi 1

```
[{"stdout": "Hello"}]
```

Output dari **ExecuteBash** untuk Iterasi 2

```
[{"stdout": "World"}]
```

Output dari **ExecuteBash** untuk Langkah

```
[{"stdout": "Hello\nWorld"}]
```

Misalnya, `ExecuteBashExecutePowerShell`, dan `ExecuteBinary` merupakan modul tindakan yang kembali STDOUT sebagai output modul tindakan. STDOUT pesan digabungkan dengan karakter baris baru untuk menghasilkan output keseluruhan dari langkah masuk `detailedOutput.json`.

AWSTOE tidak akan mengkonsolidasikan output dari iterasi yang gagal.

Modul aksi didukung oleh manajer AWSTOE komponen

Layanan pembuatan gambar, seperti EC2 Image Builder, AWSTOE menggunakan modul tindakan untuk membantu mengonfigurasi instans EC2 yang digunakan untuk membangun dan menguji

gambar mesin yang disesuaikan. Bagian ini menjelaskan fitur modul AWSTOE tindakan yang umum digunakan, dan cara mengonfigurasinya, termasuk contoh.

AWSTOEkomponen ditulis dengan dokumen YAMM plaintext. Untuk informasi selengkapnya tentang sintaks dokumen, lihat [Gunakan dokumen komponen di AWSTOE](#).

Note

Semua modul tindakan menggunakan akun yang sama dengan agen Systems Manager saat dijalankan, yang ada `root` di Linux, dan `NT Authority\SYSTEM` di Windows.

Jenis modul tindakan

- [Modul eksekusi umum](#)
- [Modul unduhan dan unggah file](#)
- [Modul operasi sistem file](#)
- [Tindakan instalasi perangkat lunak](#)
- [Modul tindakan sistem](#)

Modul eksekusi umum

Bagian berikut berisi rincian untuk modul tindakan yang melakukan perintah eksekusi umum dan instruksi.

Modul tindakan eksekusi umum

- [ExecuteBash](#)
- [ExecuteBinary](#)
- [ExecuteDocument](#)
- [ExecutePowerShell](#)

ExecuteBash

Modul ExecuteBashaksi memungkinkan Anda menjalankan skrip bash dengan kode/perintah shell sebaris. Modul ini mendukung Linux.

Semua perintah dan instruksi yang Anda tentukan di blok perintah diubah menjadi file (misalnya, `input.sh`) dan dijalankan dengan shell bash. Hasil menjalankan file shell adalah kode keluar dari langkah tersebut.

ExecuteBashModul menangani sistem restart jika skrip keluar dengan kode keluar dari 194 Saat dimulai, aplikasi melakukan salah satu tindakan berikut:

- Aplikasi menyerahkan kode keluar ke pemanggil jika dijalankan oleh Agen Systems Manager. Agen Systems Manager menangani reboot sistem dan menjalankan langkah yang sama yang memulai restart, seperti yang dijelaskan dalam [Memulai Ulang Instans Terkelola dari Skrip](#).
- Aplikasi menyimpan `arusexecutionstate`, mengonfigurasi pemicu restart untuk menjalankan kembali aplikasi, dan memulai ulang sistem.

Setelah sistem restart, aplikasi menjalankan langkah yang sama yang memulai restart. Jika Anda memerlukan fungsi ini, Anda harus menulis skrip idempoten yang dapat menangani beberapa pemanggilan dari perintah shell yang sama.

Input

Primitif	Deskripsi	Tipe	Diperlukan
<code>commands</code>	Berisi daftar instruksi atau perintah untuk dijalankan sesuai sintaks bash. YAML multi-baris diizinkan.	Daftar	Ya

Contoh masukan: Sebelum dan sesudah reboot

```
name: ExitCode194Example
description: This shows how the exit code can be used to restart a system with
  ExecuteBash
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: RestartTrigger
        action: ExecuteBash
        inputs:
```

```

commands:
  - |
    REBOOT_INDICATOR=/var/tmp/reboot-indicator
    if [ -f "${REBOOT_INDICATOR}" ]; then
      echo 'The reboot file exists. Deleting it and exiting with success.'
      rm "${REBOOT_INDICATOR}"
      exit 0
    fi
    echo 'The reboot file does not exist. Creating it and triggering a
restart.'
    touch "${REBOOT_INDICATOR}"
    exit 194

```

Output

Bidang	Deskripsi	Tipe
stdout	Output standar eksekusi perintah.	string

Jika Anda memulai reboot dan mengembalikan kode keluar 194 sebagai bagian dari modul tindakan, build akan dilanjutkan pada langkah modul tindakan yang sama yang memulai reboot. Jika Anda memulai reboot tanpa kode keluar, proses pembuatan mungkin gagal.

Contoh keluaran: Sebelum reboot (pertama kali melalui dokumen)

```

{
  "stdout": "The reboot file does not exist. Creating it and triggering a restart."
}

```

Contoh keluaran: Setelah reboot, (kedua kalinya melalui dokumen)

```

{
  "stdout": "The reboot file exists. Deleting it and exiting with success."
}

```

ExecuteBinary

Modul `ExecuteBinary` tindakan memungkinkan Anda menjalankan file biner dengan daftar argumen baris perintah.

ExecuteBinaryModul ini menangani sistem restart jika file biner keluar dengan kode keluar dari 194 (Linux) atau 3010 (Windows). Ketika ini terjadi, aplikasi melakukan salah satu tindakan berikut:

- Aplikasi menyerahkan kode keluar ke pemanggil jika dijalankan oleh Agen Systems Manager. Agen Systems Manager menangani memulai ulang sistem dan menjalankan langkah yang sama yang memulai restart, seperti yang dijelaskan dalam [Memulai Ulang Instans Terkelola dari Skrip](#).
- Aplikasi menyimpan `arusexecutionstate`, mengonfigurasi pemacu restart untuk menjalankan kembali aplikasi, dan memulai ulang sistem.

Setelah sistem restart, aplikasi menjalankan langkah yang sama yang memulai restart. Jika Anda memerlukan fungsi ini, Anda harus menulis skrip idempoten yang dapat menangani beberapa pemanggilan dari perintah shell yang sama.

Input

Primitif	Deskripsi	Tipe	Diperlukan
<code>path</code>	Path ke file biner untuk eksekusi.	Tali	Ya
<code>arguments</code>	Berisi daftar argumen baris perintah untuk digunakan saat menjalankan biner.	Daftar String	Tidak

Contoh masukan: instal .NET

```
name: "InstallDotnet"
action: ExecuteBinary
inputs:
  path: C:\PathTo\dotnet_installer.exe
  arguments:
    - /qb
    - /norestart
```

Output

Bidang	Deskripsi	Tipe
stdout	Output standar eksekusi perintah.	string

Contoh keluaran

```
{
  "stdout": "success"
}
```

ExecuteDocument

Modul `ExecuteDocument` tindakan menambahkan dukungan untuk dokumen komponen bersarang, menjalankan beberapa dokumen komponen dari satu dokumen. `AWSTOEmemvalidasi` dokumen yang diteruskan dalam parameter input pada waktu berjalan.

Pembatasan

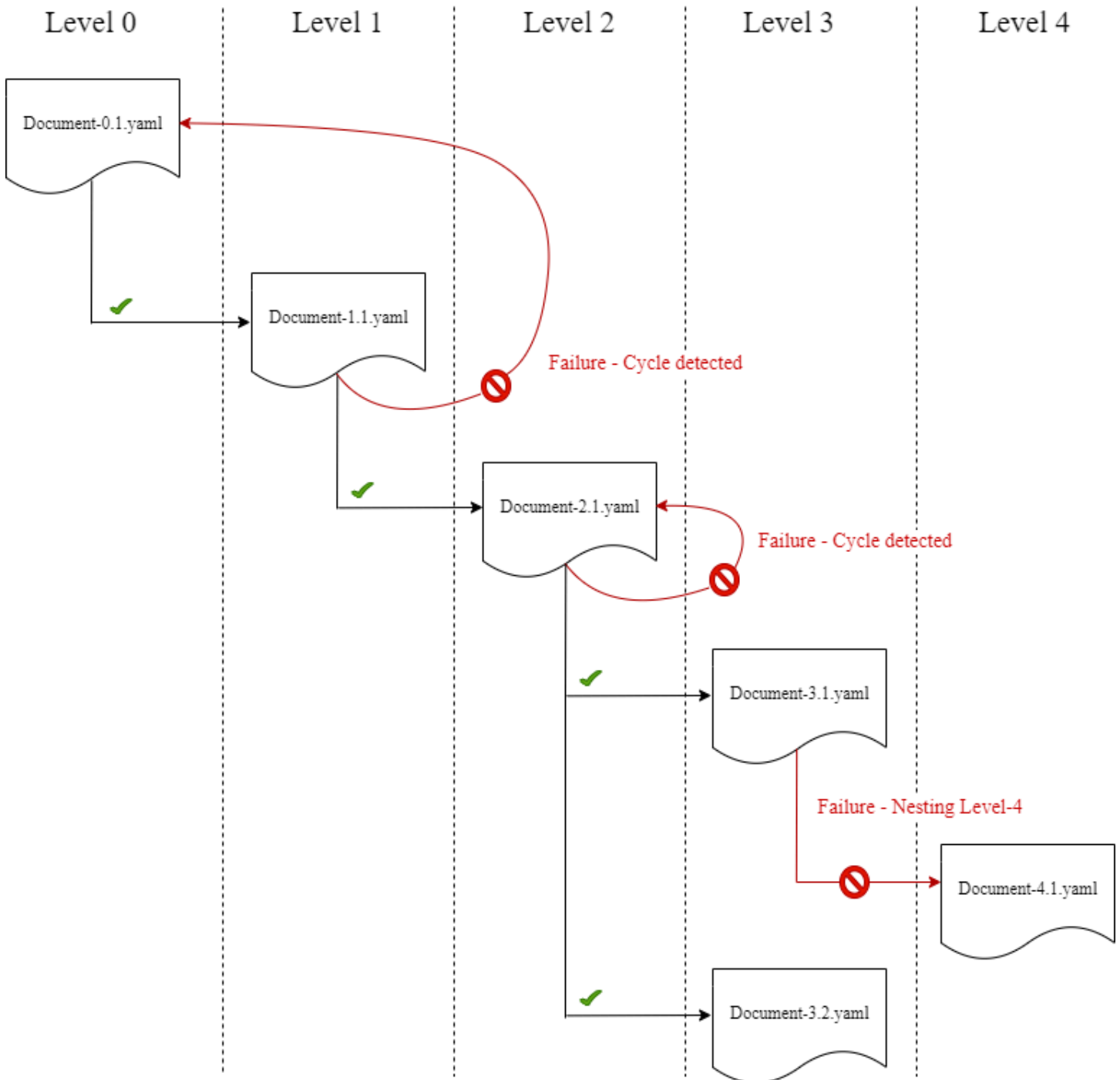
- Modul tindakan ini berjalan satu kali, tanpa percobaan ulang yang diizinkan, dan tidak ada opsi untuk menetapkan batas waktu tunggu. `ExecuteDocument` menetapkan nilai default berikut, dan mengembalikan kesalahan jika Anda mencoba untuk mengubahnya.
 - `timeoutSeconds`: -1
 - `maxAttempts`: 1

Note

Anda dapat membiarkan nilai-nilai ini kosong, dan `AWSTOEmenggunakan` nilai default.

- Penyarangan dokumen diperbolehkan, hingga kedalaman tiga tingkat, tetapi tidak lebih dari itu. Tiga tingkat bersarang diterjemahkan ke empat tingkat dokumen, karena tingkat atas tidak bersarang. Dalam skenario ini, dokumen tingkat terendah tidak boleh memanggil dokumen lain.
- Eksekusi siklik dokumen komponen tidak diperbolehkan. Setiap dokumen yang menyebut dirinya di luar konstruksi perulangan, atau yang memanggil dokumen lain yang lebih tinggi dalam rantai eksekusi saat ini, memulai siklus yang dapat menghasilkan loop tanpa akhir. Ketika `AWSTOEmendeteksi` eksekusi siklik, itu menghentikan eksekusi dan mencatat kegagalan.

ExecuteDocument action module Component document nesting levels



Jika dokumen komponen mencoba menjalankannya sendiri, atau menjalankan salah satu dokumen komponen yang lebih tinggi dalam rantai eksekusi saat ini, eksekusi gagal.

Masukan

Primitif	Deskripsi	Tipe	Diperlukan
document	<p>Jalur dokumen komponen. Pilihan yang valid meliputi:</p> <ul style="list-style-type: none"> • Jalur file lokal • URI S3 • ARN versi pembuatan komponen EC2 Image Builder 	Tali	Ya
document-s3-bucket-owner	ID akun pemilik bucket S3 untuk bucket S3 tempat dokumen komponen disimpan. (Direkomendasikan jika Anda menggunakan URI S3 di dokumen komponen Anda.)	String	Tidak
phases	Fase untuk dijalankan dalam dokumen komponen, dinyatakan sebagai daftar dipisahkan koma. Jika tidak ada fase yang ditentukan, maka semua fase berjalan.	String	Tidak
parameters	Parameter masukan yang diteruskan ke dokumen komponen	Daftar Peta Parameter	Tidak

Primitif	Deskripsi	Tipe	Diperlukan
	saat runtime sebagai pasangan nilai kunci.		

Masukan peta parameter

Primitif	Deskripsi	Tipe	Diperlukan
name	Nama parameter input untuk diteruskan ke dokumen komponen yang sedang dijalankan modul ExecuteDocument tindakan.	Tali	Ya
value	Nilai parameter input.	Tali	Ya

Contoh masukan

Contoh berikut menunjukkan variasi input untuk dokumen komponen Anda, tergantung pada jalur instalasi Anda.

Contoh masukan: Jalur dokumen lokal

```
# main.yaml
schemaVersion: 1.0

phases:
  - name: build
    steps:
      - name: ExecuteNestedDocument
        action: ExecuteDocument
        inputs:
          document: Sample-1.yaml
          phases: build
          parameters:
            - name: parameter-1
              value: value-1
```

```
- name: parameter-2
  value: value-2
```

Contoh masukan: URI S3 sebagai jalur dokumen

```
# main.yaml
schemaVersion: 1.0

phases:
  - name: build
    steps:
      - name: ExecuteNestedDocument
        action: ExecuteDocument
        inputs:
          document: s3://my-bucket/Sample-1.yaml
          document-s3-bucket-owner: 123456789012
          phases: build,validate
          parameters:
            - name: parameter-1
              value: value-1
            - name: parameter-2
              value: value-2
```

Contoh masukan: Komponen EC2 Image Builder ARN sebagai jalur dokumen

```
# main.yaml
schemaVersion: 1.0

phases:
  - name: build
    steps:
      - name: ExecuteNestedDocument
        action: ExecuteDocument
        inputs:
          document: arn:aws:imagebuilder:us-west-2:aws:component/Sample-Test/1.0.0
          phases: test
          parameters:
            - name: parameter-1
              value: value-1
            - name: parameter-2
              value: value-2
```

Menggunakan ForEach loop untuk menjalankan dokumen

```
# main.yaml
schemaVersion: 1.0

phases:
  - name: build
    steps:
      - name: ExecuteNestedDocument
        action: ExecuteDocument
        loop:
          name: 'myForEachLoop'
          forEach:
            - Sample-1.yaml
            - Sample-2.yaml
        inputs:
          document: "{{myForEachLoop.value}}"
          phases: test
          parameters:
            - name: parameter-1
              value: value-1
            - name: parameter-2
              value: value-2
```

Menggunakan loop For untuk menjalankan dokumen

```
# main.yaml
schemaVersion: 1.0

phases:
  - name: build
    steps:
      - name: ExecuteNestedDocument
        action: ExecuteDocument
        loop:
          name: 'myForLoop'
          for:
            start: 1
            end: 2
            updateBy: 1
        inputs:
          document: "Sample-{{myForLoop.value}}.yaml"
          phases: test
          parameters:
            - name: parameter-1
```

```

    value: value-1
  - name: parameter-2
    value: value-2

```

Output

AWSTOEmembuat file output yang disebut `detailedoutput.json` setiap kali dijalankan. File berisi rincian tentang setiap fase dan langkah dari setiap dokumen komponen yang dipanggil saat sedang berjalan. Untuk modul `ExecuteDocument`tindakan, Anda dapat menemukan ringkasan runtime singkat di `outputs` lapangan, dan detail tentang fase, langkah, dan dokumen yang dijalankan di `detailedOutput`.

```

"outputs": "[{"executedStepCount":1,"executionId":"97054e22-06cc-11ec-9b14-acde48001122","failedStepCount":0,"failureMessage":"","ignoredFailedStepCount":0,"logUrl":"","status":"success"}]",

```

Setiap objek ringkasan keluaran dokumen komponen berisi rincian berikut, seperti yang ditunjukkan di sini, dengan nilai sampel:

- `executedStepCount`: 1
- `ExecutionID` : "12345a67-89bc-01de-2f34-abcd56789012"
- `failedStepCount`: 0
- `FailureMessage`: ""
- `ignoredFailedStepHitung` : 0
- `logUrl`: ""
- `status` : "sukses"

Contoh keluaran

Contoh berikut menunjukkan output dari modul `ExecuteDocument`tindakan ketika eksekusi bersarang terjadi. Dalam contoh ini, dokumen `main.yaml` komponen berhasil menjalankan dokumen `Sample-1.yaml` komponen.

```

{
  "executionId": "12345a67-89bc-01de-2f34-abcd56789012",
  "status": "success",
  "startTime": "2021-08-26T17:20:31-07:00",
  "endTime": "2021-08-26T17:20:31-07:00",

```

```

"failureMessage": "",
"documents": [
  {
    "name": "",
    "filePath": "main.yaml",
    "status": "success",
    "description": "",
    "startTime": "2021-08-26T17:20:31-07:00",
    "endTime": "2021-08-26T17:20:31-07:00",
    "failureMessage": "",
    "phases": [
      {
        "name": "build",
        "status": "success",
        "startTime": "2021-08-26T17:20:31-07:00",
        "endTime": "2021-08-26T17:20:31-07:00",
        "failureMessage": "",
        "steps": [
          {
            "name": "ExecuteNestedDocument",
            "status": "success",
            "failureMessage": "",
            "timeoutSeconds": -1,
            "onFailure": "Abort",
            "maxAttempts": 1,
            "action": "ExecuteDocument",
            "startTime": "2021-08-26T17:20:31-07:00",
            "endTime": "2021-08-26T17:20:31-07:00",
            "inputs": "[{\"document\":\"Sample-1.yaml\",\"document-s3-
bucket-owner\":\"\",\"phases\":\"\",\"parameters\":null}]",
            "outputs": "[{\"executedStepCount\":1,\"executionId\":
\"98765f43-21ed-09cb-8a76-fedc54321098\",\"failedStepCount\":0,\"failureMessage\":\"\",
\"ignoredFailedStepCount\":0,\"logUrl\":\"\",\"status\":\"success\"}]",
            "loop": null,
            "detailedOutput": [
              {
                "executionId": "98765f43-21ed-09cb-8a76-
fedc54321098",
                "status": "success",
                "startTime": "2021-08-26T17:20:31-07:00",
                "endTime": "2021-08-26T17:20:31-07:00",
                "failureMessage": "",
                "documents": [
                  {

```


}

ExecutePowerShell

Modul ExecutePowerShell tindakan memungkinkan Anda menjalankan PowerShell skrip dengan kode/perintah shell sebaris. Modul ini mendukung platform Windows dan Windows PowerShell.

Semua perintah/instruksi yang ditentukan dalam blok perintah diubah menjadi file skrip (misalnya, `input.ps1`) dan dijalankan menggunakan Windows PowerShell. Hasil menjalankan file shell adalah kode keluar.

ExecutePowerShellModul menangani sistem restart jika perintah shell keluar dengan kode keluar dari. Saat dimulai, aplikasi melakukan salah satu tindakan berikut:

- Menyerahkan kode keluar ke pemanggil jika dijalankan oleh Agen Systems Manager. Agen Systems Manager menangani reboot sistem dan menjalankan langkah yang sama yang memulai restart, seperti yang dijelaskan dalam [Memulai Ulang Instans Terkelola dari Skrip](#).
- Menyimpan `arusexecutionstate`, mengonfigurasi pemacu restart untuk menjalankan kembali aplikasi, dan me-reboot sistem.

Setelah sistem restart, aplikasi menjalankan langkah yang sama yang memulai restart. Jika Anda memerlukan fungsi ini, Anda harus menulis skrip idempoten yang dapat menangani beberapa pemanggilan dari perintah shell yang sama.

Input

Primitif	Deskripsi	Tipe	Diperlukan
<code>commands</code>	Berisi daftar instruksi atau perintah untuk dijalankan sesuai PowerShell sintaks. YAML multi-baris diizinkan.	Daftar String	Ya. Harus menentukan <code>commands</code> atau <code>file</code> , tidak keduanya.
<code>file</code>	Berisi path ke file PowerShell script. PowerShell akan berjalan melawan	String	Ya. Harus menentukan <code>commands</code>

Primitif	Deskripsi	Tipe	Diperlukan
	file ini menggunakan argumen baris <code>-file</code> perintah. Jalur harus menunjuk ke <code>.ps1</code> file.		atau <code>file</code> , tidak keduanya.

Contoh masukan: Sebelum dan sesudah reboot

```
name: ExitCode3010Example
description: This shows how the exit code can be used to restart a system with
  ExecutePowerShell
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: RestartTrigger
        action: ExecutePowerShell
        inputs:
          commands:
            - |
              $rebootIndicator = Join-Path -Path $env:SystemDrive -ChildPath 'reboot-
indicator'
              if (Test-Path -Path $rebootIndicator) {
                Write-Host 'The reboot file exists. Deleting it and exiting with
success.'
                Remove-Item -Path $rebootIndicator -Force | Out-Null
                [System.Environment]::Exit(0)
              }
              Write-Host 'The reboot file does not exist. Creating it and triggering a
restart.'
              New-Item -Path $rebootIndicator -ItemType File | Out-Null
              [System.Environment]::Exit(3010)
```

Output

Bidang	Deskripsi	Tipe
<code>stdout</code>	Output standar eksekusi perintah.	string

Jika Anda menjalankan reboot dan mengembalikan kode keluar 3010 sebagai bagian dari modul tindakan, build akan dilanjutkan pada langkah modul tindakan yang sama yang memulai reboot. Jika Anda menjalankan reboot tanpa kode keluar, proses build mungkin gagal.

Contoh keluaran: Sebelum reboot (pertama kali melalui dokumen)

```
{
  "stdout": "The reboot file does not exist. Creating it and triggering a restart."
}
```

Contoh keluaran: Setelah reboot, (kedua kalinya melalui dokumen)

```
{
  "stdout": "The reboot file exists. Deleting it and exiting with success."
}
```

Modul unduhan dan unggah file

Bagian berikut berisi rincian untuk modul tindakan yang melakukan perintah dan instruksi download dan upload.

Unduh dan unggah modul tindakan

- [S3Unduh](#)
- [S3Unggah](#)
- [WebDownload](#)

S3Unduh

Dengan modul `S3Download` tindakan, Anda dapat mengunduh objek Amazon S3, atau sekumpulan objek, ke file atau folder lokal yang Anda tentukan dengan jalur `destination`. Jika ada file yang sudah ada di lokasi yang ditentukan, dan `overwrite` bendera disetel ke `true`, `S3Download` timpa file tersebut.

`sourceLokasi` Anda dapat menunjuk ke objek tertentu di Amazon S3, atau Anda dapat menggunakan key prefix dengan wildcard asterisk `*` () untuk mengunduh sekumpulan objek yang cocok dengan jalur awalan key. Saat Anda menentukan key prefix di `source` lokasi Anda, modul `S3Download` tindakan akan mengunduh semua yang cocok dengan awalan (termasuk file dan folder). Pastikan bahwa key prefix diakhiri dengan garis miring maju, diikuti dengan tanda bintang

(/*), sehingga Anda mengunduh semua yang cocok dengan awalan. Sebagai contoh: `s3://my-bucket/my-folder/*`.

Note

Semua folder di jalur tujuan harus ada sebelum mengunduh, atau unduhan gagal.

Jika `S3Download` tindakan untuk key prefix tertentu gagal selama download, konten folder tidak digulung kembali ke statusnya sebelum kegagalan. Folder tujuan tetap seperti pada saat kegagalan.

Kasus penggunaan yang didukung

Modul `S3Download` tindakan mendukung kasus penggunaan berikut:

- Objek Amazon S3 diunduh ke folder lokal, seperti yang ditentukan dalam jalur unduhan.
- Objek Amazon S3 (dengan awalan kunci di jalur file Amazon S3) diunduh ke folder lokal yang ditentukan, yang secara rekursif menyalin semua objek Amazon S3 yang cocok dengan awalan key ke folder lokal.

Persyaratan IAM

Peran IAM yang Anda kaitkan dengan profil instans Anda harus memiliki izin untuk menjalankan modul `S3Download` tindakan. Kebijakan IAM berikut harus dilampirkan pada peran IAM yang terkait dengan profil instans:

- File tunggal: `s3:GetObject` terhadap ember/objek (misalnya,). `arn:aws:s3:::BucketName/*`
- Beberapa file: `s3:ListBucket` terhadap ember/objek (misalnya, `arn:aws:s3:::BucketName`) dan `s3:GetObject` terhadap ember/objek (misalnya,). `arn:aws:s3:::BucketName/*`

Input


Primitif	Deskripsi	Tipe	Diperlukan	Default
<code>source</code>	Bucket Amazon S3 yang merupakan sumber unduhan Anda.	Tali	Ya	T/A

Primitif	Deskripsi	Tipe	Diperlukan	Default
	Anda dapat menentukan path ke objek tertentu, atau menggunakan key prefix, yang diakhiri dengan garis miring maju, diikuti dengan wildcard asterisk (/*), untuk mengunduh sekumpulan objek yang cocok dengan key prefix.			
destination	Jalur lokal tempat objek Amazon S3 diunduh. Untuk mengunduh satu file, Anda harus menentukan nama file sebagai bagian dari jalur. Sebagai contoh, <i>/myfolder/package.zip</i> .	Tali	Ya	T/A

Primitif	Deskripsi	Tipe	Diperlukan	Default
<code>expectedBucketOwner</code>	ID akun pemilik yang diharapkan dari bucket yang disediakan di source jalur. Kami menyarankan Anda memverifikasi kepemilikan bucket Amazon S3 yang ditentukan dalam sumbernya.	String	Tidak	T/A

Primitif	Deskripsi	Tipe	Diperlukan	Default
<code>overwrite</code>	<p>Ketika disetel ke <code>true</code>, jika file dengan nama yang sama sudah ada di folder tujuan untuk jalur lokal yang ditentukan, file unduhan menimpa file lokal. Ketika disetel ke <code>false</code>, file yang ada di sistem lokal dilindungi agar tidak ditimpa, dan modul tindakan gagal dengan kesalahan unduhan.</p> <p>Sebagai contoh, <code>Error: S3Download: File already exists and "overwrite" property for "destination" file is set to false.</code></p>	Boolean	Tidak	betul

Primitif	Deskripsi	Tipe	Diperlukan	Default
	Cannot download. .			

 Note

Untuk contoh berikut, jalur folder Windows dapat diganti dengan jalur Linux. Misalnya, *C:\myfolder\package.zip* bisa diganti dengan */myfolder/package.zip*.

Contoh masukan: salin objek Amazon S3 ke file lokal

Contoh berikut menunjukkan cara menyalin objek Amazon S3 ke file lokal.

```
name: DownloadMyFile
action: S3Download
inputs:
  - source: s3://mybucket/path/to/package.zip
    destination: C:\myfolder\package.zip
    expectedBucketOwner: 123456789022
    overwrite: false
  - source: s3://mybucket/path/to/package.zip
    destination: C:\myfolder\package.zip
    expectedBucketOwner: 123456789022
    overwrite: true
  - source: s3://mybucket/path/to/package.zip
    destination: C:\myfolder\package.zip
    expectedBucketOwner: 123456789022
```

Contoh masukan: salin semua objek Amazon S3 di bucket Amazon S3 dengan key prefix ke folder lokal

Contoh berikut menunjukkan cara menyalin semua objek Amazon S3 di bucket Amazon S3 dengan awalan key ke folder lokal. Amazon S3 tidak memiliki konsep folder, oleh karena itu semua objek yang cocok dengan key prefix disalin. Jumlah maksimum objek yang dapat diunduh adalah 1000.

```
name: MyS3DownloadKeyprefix
action: S3Download
```

```
maxAttempts: 3
inputs:
  - source: s3://mybucket/path/to/*
    destination: C:\myfolder\
    expectedBucketOwner: 123456789022
    overwrite: false
  - source: s3://mybucket/path/to/*
    destination: C:\myfolder\
    expectedBucketOwner: 123456789022
    overwrite: true
  - source: s3://mybucket/path/to/*
    destination: C:\myfolder\
    expectedBucketOwner: 123456789022
```

Output

Tidak ada.

S3Unggah

Dengan modul tindakan S3Upload, Anda dapat mengunggah file dari file sumber atau folder ke lokasi Amazon S3. Anda dapat menggunakan wildcard (*) di jalur yang ditentukan untuk lokasi sumber Anda untuk mengunggah semua file yang jalurnya cocok dengan pola wildcard.

Jika tindakan S3Upload rekursif gagal, file apa pun yang telah diunggah akan tetap berada di bucket Amazon S3 tujuan.

Kasus penggunaan yang didukung

- File lokal ke objek Amazon S3.
- File lokal dalam folder (dengan wildcard) ke awalan kunci Amazon S3.
- Salin folder lokal (harus `recurse` disetel ke `true`) ke Amazon S3 key prefix.

Persyaratan IAM

Peran IAM yang Anda kaitkan dengan profil instans Anda harus memiliki izin untuk menjalankan modul S3Upload tindakan. Kebijakan IAM berikut harus dilampirkan pada peran IAM yang terkait dengan profil instans. Kebijakan harus memberikan `s3:PutObject` izin ke bucket Amazon S3 target. Misalnya, `arn:aws:s3:::BucketName/*`).

Input

Primitif	Deskripsi	Tipe	Diperlukan	Default
<code>source</code>	Jalur lokal tempat file sumber/fo lder berasal. <code>source</code> mendukung wildcard asterisk (*)	Tali	Ya	T/A
<code>destination</code>	Jalur untuk bucket Amazon S3 tujuan tempat file sumber/folder diunggah.	Tali	Ya	T/A
<code>recurse</code>	Saat disetel ke <code>true</code> , lakukan <code>S3Upload</code> secara rekursif.	String	Tidak	<code>false</code>
<code>expectedBucketOwner</code>	ID akun pemilik yang diharapkan untuk bucket Amazon S3 yang ditentukan di jalur tujuan. Kami menyarankan Anda memverifikasi kepemilikan bucket Amazon S3 yang ditentukan di tujuan.	String	Tidak	T/A

Contoh masukan: salin file lokal ke objek Amazon S3

Contoh berikut menunjukkan cara menyalin file lokal ke objek Amazon S3.

```
name: MyS3UploadFile
action: S3Upload
onFailure: Abort
maxAttempts: 3
inputs:
  - source: C:\myfolder\package.zip
    destination: s3://mybucket/path/to/package.zip
    expectedBucketOwner: 123456789022
```

Contoh masukan: salin semua file dalam folder lokal ke bucket Amazon S3 dengan key prefix

Contoh berikut menunjukkan cara menyalin semua file di folder lokal ke bucket Amazon S3 dengan key prefix. Contoh ini tidak menyalin sub-folder atau isinya karena tidak recurse ditentukan, dan defaultnya. false

```
name: MyS3UploadMultipleFiles
action: S3Upload
onFailure: Abort
maxAttempts: 3
inputs:
  - source: C:\myfolder\*
    destination: s3://mybucket/path/to/
    expectedBucketOwner: 123456789022
```

Contoh masukan: salin semua file dan folder secara rekursif dari folder lokal ke bucket Amazon S3

Contoh berikut menunjukkan cara menyalin semua file dan folder secara rekursif dari folder lokal ke bucket Amazon S3 dengan key prefix.

```
name: MyS3UploadFolder
action: S3Upload
onFailure: Abort
maxAttempts: 3
inputs:
  - source: C:\myfolder\*
    destination: s3://mybucket/path/to/
    recurse: true
    expectedBucketOwner: 123456789022
```

Output

Tidak ada.

WebDownload

Modul WebDownload tindakan memungkinkan Anda mengunduh file dan sumber daya dari lokasi terpencil melalui protokol HTTP/HTTPS (disarankan HTTPS). Tidak ada batasan jumlah atau ukuran unduhan. Modul ini menangani coba lagi dan logika backoff eksponensial.

Setiap operasi pengunduhan dialokasikan maksimal 5 upaya untuk berhasil sesuai dengan input pengguna. Upaya ini berbeda dari yang ditentukan di `maxAttempts` bidang `documentSteps`, yang terkait dengan kegagalan modul tindakan.

Modul tindakan ini secara implisit menangani pengalihan. Semua kode status HTTP, kecuali 200, menghasilkan kesalahan.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Default
<code>source</code>	URL HTTP/HTTPS yang valid (disarankan HTTPS), yang mengikuti standar RFC 3986. Ekspresi rantai diizinkan.	Tali	Ya	T/A
<code>destination</code>	Sebuah file absolut atau relatif atau jalur folder pada sistem lokal. Jalur folder harus diakhiri dengan <code>/</code> . Jika mereka tidak berakhir	Tali	Ya	T/A

Primitif	Deskripsi	Tipe	Diperlukan	Default
	dengan/, mereka akan diperlakukan sebagai jalur file. Modul ini membuat file atau folder yang diperlukan untuk unduhan yang berhasil. Ekspresi rantai diizinkan.			

Primitif	Deskripsi	Tipe	Diperlukan	Default
<code>overwrite</code>	Saat diaktifkan, timpa file yang ada di sistem lokal dengan file atau sumber daya yang diunduh. Saat tidak diaktifkan, file apa pun yang ada di sistem lokal tidak ditimpa, dan modul tindakan gagal dengan kesalahan. Ketika timpa diaktifkan dan checksum dan algoritma ditentukan, maka modul tindakan mengunduh file hanya jika checksum dan hash dari file yang sudah ada sebelumnya tidak cocok.	Boolean	Tidak	<code>true</code>

Primitif	Deskripsi	Tipe	Diperlukan	Default
checksum	<p>Ketika Anda menentukan checksum, itu diperiksa terhadap hash dari file yang diunduh yang dihasilkan dengan algoritma yang disediakan. Agar verifikasi file diaktifkan, checksum dan algoritme harus disediakan. Ekspresi rantai diizinkan.</p>	String	Tidak	T/A
algorithm	<p>Algoritma yang digunakan untuk menghitung checksum. Pilihannya adalah MD5, SHA1, SHA256, dan SHA512. Agar verifikasi file diaktifkan, checksum dan algoritme harus disediakan. Ekspresi rantai diizinkan.</p>	String	Tidak	T/A

Primitif	Deskripsi	Tipe	Diperlukan	Default
<code>ignoreCertificateErrors</code>	Validasi sertifikat SSL diabaikan saat diaktifkan.	Boolean	Tidak	<code>false</code>

Output

Primitif	Deskripsi	Tipe				
<code>destination</code>	String berbatasan karakter baris baru yang menentukan jalur tujuan tempat file atau sumber daya yang diunduh disimpan.	String				

Contoh masukan: unduh file jarak jauh ke tujuan lokal

```
name: DownloadRemoteFile
action: WebDownload
maxAttempts: 3
inputs:
  - source: https://testdomain/path/to/java14.zip
    destination: C:\testfolder\package.zip
```

```
Output:
{
  "destination": "C:\\testfolder\\package.zip"
```

```
}
```

Contoh masukan: unduh lebih dari satu file jarak jauh ke lebih dari satu tujuan lokal

```
name: DownloadRemoteFiles
action: WebDownload
maxAttempts: 3
inputs:
  - source: https://testdomain/path/to/java14.zip
    destination: /tmp/java14_renamed.zip
  - source: https://testdomain/path/to/java14.zip
    destination: /tmp/create_new_folder_and_add_java14_as_zip/
```

Output:

```
{
  "destination": "/tmp/create_new_folder/java14_renamed.zip\n/tmp/
create_new_folder_and_add_java14_as_zip/java14.zip"
}
```

Contoh masukan: unduh satu file jarak jauh tanpa menimpa tujuan lokal, dan unduh file jarak jauh lainnya dengan verifikasi file

```
name: DownloadRemoteMultipleProperties
action: WebDownload
maxAttempts: 3
inputs:
  - source: https://testdomain/path/to/java14.zip
    destination: C:\create_new_folder\java14_renamed.zip
    overwrite: false
  - source: https://testdomain/path/to/java14.zip
    destination: C:\create_new_folder_and_add_java14_as_zip\
    checksum: ac68bbf921d953d1cfab916cb6120864
    algorithm: MD5
    overwrite: true
```

Output:

```
{
  "destination": "C:\\create_new_folder\\java14_renamed.zip\nC:\\
create_new_folder_and_add_java14_as_zip\\java14.zip"
}
```


Contoh masukan: unduh file jarak jauh dan abaikan validasi sertifikasi SSL

```
name: DownloadRemoteIgnoreValidation
action: WebDownload
maxAttempts: 3
inputs:
  - source: https://www.bad-ssl.com/resource
    destination: /tmp/downloads/
    ignoreCertificateErrors: true
```

```
Output:
{
  "destination": "/tmp/downloads/resource"
}
```

Modul operasi sistem file

Bagian berikut berisi rincian untuk modul tindakan yang melakukan perintah dan instruksi operasi sistem file.

Modul tindakan operasi sistem file

- [AppendFile](#)
- [CopyFile](#)
- [CopyFolder](#)
- [CreateFile](#)
- [CreateFolder](#)
- [CreateSymlink](#)
- [DeleteFile](#)
- [DeleteFolder](#)
- [ListFiles](#)
- [MoveFile](#)
- [MoveFolder](#)
- [ReadFile](#)
- [SetFileEncoding](#)
- [SetFileOwner](#)

- [SetFolderOwner](#)
- [SetFilePermissions](#)
- [SetFolderPermissions](#)

AppendFile

Modul AppendFile tindakan menambahkan konten tertentu ke konten file yang sudah ada sebelumnya.

Jika nilai encoding file berbeda dari nilai encoding (utf-8) default, maka Anda dapat menentukan nilai encoding file dengan menggunakan opsi. encoding Secara default, utf-16 dan utf-32 diasumsikan menggunakan pengkodean endian kecil.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- File yang ditentukan tidak ada saat runtime.
- Anda tidak memiliki izin menulis untuk mengubah konten file.
- Modul mengalami kesalahan selama operasi file.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
path	Jalur file.	Tali	Ya	T/A	N/A	Ya
content	Konten yang akan ditambahkan ke file.	String	Tidak	Empty string	T/A	Ya
encoding	Standar pengkodean.	String	Tidak	utf8	utf8,utf-8, utf16-LE, utf-16-LE, utf16-BE,utf-16-	Ya

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
					BE ,,utf32,utf-32-LE,utf-32-LE,utf32-BE, dan utf-32-BE . Nilai opsi pengkodean tidak peka huruf besar/kecil.	

Contoh masukan: tambahkan file tanpa pengkodean (Linux)

```
name: AppendingFileWithoutEncodingLinux
action: AppendFile
inputs:
  - path: ./Sample.txt
    content: "The string to be appended to the file"
```

Contoh masukan: tambahkan file tanpa pengkodean (Windows)

```
name: AppendingFileWithoutEncodingWindows
action: AppendFile
inputs:
  - path: C:\MyFolder\MyFile.txt
    content: "The string to be appended to the file"
```

Contoh masukan: tambahkan file dengan encoding (Linux)

```
name: AppendingFileWithEncodingLinux
action: AppendFile
```

```
inputs:
  - path: /FolderName/SampleFile.txt
    content: "The string to be appended to the file"
    encoding: UTF-32
```

Contoh masukan: tambahkan file dengan pengkodean (Windows)

```
name: AppendingFileWithEncodingWindows
action: AppendFile
inputs:
  - path: C:\MyFolderName\SampleFile.txt
    content: "The string to be appended to the file"
    encoding: UTF-32
```

Contoh masukan: tambahkan file dengan string kosong (Linux)

```
name: AppendingEmptyStringLinux
action: AppendFile
inputs:
  - path: /FolderName/SampleFile.txt
```

Contoh masukan: tambahkan file dengan string kosong (Windows)

```
name: AppendingEmptyStringWindows
action: AppendFile
inputs:
  - path: C:\MyFolderName\SampleFile.txt
```

Output

Tidak ada.

CopyFile

Modul CopyFile tindakan menyalin file dari sumber yang ditentukan ke tujuan yang ditentukan. Secara default, modul secara rekursif membuat folder tujuan jika tidak ada saat runtime.

Jika file dengan nama yang ditentukan sudah ada di folder yang ditentukan, modul tindakan, secara default, menimpa file yang ada. Anda dapat mengganti perilaku default ini dengan menyetel opsi overwrite ke `false` Ketika opsi overwrite diatur ke `false`, dan sudah ada file di lokasi yang

ditentukan dengan nama yang ditentukan, modul tindakan akan mengembalikan kesalahan. Opsi ini bekerja sama dengan `cp` perintah di Linux, yang menimpa secara default.

Nama file sumber dapat menyertakan wildcard (*). Karakter wildcard diterima hanya setelah pemisah jalur file terakhir (/atau\). Jika karakter wildcard disertakan dalam nama file sumber, semua file yang cocok dengan wildcard disalin ke folder tujuan. Jika Anda ingin memindahkan lebih dari satu file dengan menggunakan karakter wildcard, input ke `destination` opsi harus diakhiri dengan pemisah jalur file (/atau\), yang menunjukkan bahwa input tujuan adalah folder.

Jika nama file tujuan berbeda dari nama file sumber, Anda dapat menentukan nama file tujuan menggunakan `destination` opsi. Jika Anda tidak menentukan nama file tujuan, nama file sumber digunakan untuk membuat file tujuan. Setiap teks yang mengikuti pemisah jalur file terakhir (/atau\), diperlakukan sebagai nama file. Jika Anda ingin menggunakan nama file yang sama dengan file sumber, maka input `destination` opsi harus diakhiri dengan pemisah jalur file (/atau\).

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk membuat file di folder yang ditentukan.
- File sumber tidak ada saat runtime.
- Sudah ada folder dengan nama file yang ditentukan dan `overwrite` opsi diatur ke `false`.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
<code>source</code>	Jalur file sumber.	Tali	Ya	T/A	N/A	Ya
<code>destination</code>	Jalur file tujuan.	Tali	Ya	T/A	N/A	Ya
<code>overwrite</code>	Ketika diatur ke <code>false</code> , file tujuan	Boolean	Tidak	<code>true</code>	T/A	Ya

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
	tidak akan diganti ketika sudah ada file di lokasi yang ditentukan dengan nama yang ditentukan.					

Contoh masukan: salin file (Linux)

```
name: CopyingAFileLinux
action: CopyFile
inputs:
  - source: /Sample/MyFolder/Sample.txt
    destination: /MyFolder/destinationFile.txt
```

Contoh masukan: salin file (Windows)

```
name: CopyingAFileWindows
action: CopyFile
inputs:
  - source: C:\MyFolder\Sample.txt
    destination: C:\MyFolder\destinationFile.txt
```

Contoh masukan: salin file menggunakan nama file sumber (Linux)

```
name: CopyingFileWithSourceFileNameLinux
action: CopyFile
inputs:
  - source: /Sample/MyFolder/Sample.txt
    destination: /MyFolder/
```

Contoh masukan: salin file menggunakan nama file sumber (Windows)

```
name: CopyingFileWithSourceFileNameWindows
action: CopyFile
inputs:
  - source: C:\Sample\MyFolder\Sample.txt
    destination: C:\MyFolder\
```

Contoh masukan: salin file menggunakan karakter wildcard (Linux)

```
name: CopyingFilesWithWildCardLinux
action: CopyFile
inputs:
  - source: /Sample/MyFolder/Sample*
    destination: /MyFolder/
```

Contoh masukan: salin file menggunakan karakter wildcard (Windows)

```
name: CopyingFilesWithWildCardWindows
action: CopyFile
inputs:
  - source: C:\Sample\MyFolder\Sample*
    destination: C:\MyFolder\
```

Contoh masukan: salin file tanpa menimpa (Linux)

```
name: CopyingFilesWithoutOverwriteLinux
action: CopyFile
inputs:
  - source: /Sample/MyFolder/Sample.txt
    destination: /MyFolder/destinationFile.txt
    overwrite: false
```

Contoh masukan: salin file tanpa menimpa (Windows)

```
name: CopyingFilesWithoutOverwriteWindows
action: CopyFile
inputs:
  - source: C:\Sample\MyFolder\Sample.txt
    destination: C:\MyFolder\destinationFile.txt
```

```
overwrite: false
```

Output

Tidak ada.

CopyFolder

Modul CopyFolder tindakan menyalin folder dari sumber yang ditentukan ke tujuan yang ditentukan. Input untuk `source` opsi adalah folder yang akan disalin, dan input untuk `destination` opsi adalah folder tempat konten folder sumber disalin. Secara default, modul secara rekursif membuat folder tujuan jika tidak ada saat runtime.

Jika folder dengan nama yang ditentukan sudah ada di folder yang ditentukan, modul tindakan, secara default, menimpa folder yang ada. Anda dapat mengganti perilaku default ini dengan menyetel opsi `overwrite` ke `false`. Ketika opsi `overwrite` diatur ke `false`, dan sudah ada folder di lokasi yang ditentukan dengan nama yang ditentukan, modul tindakan akan mengembalikan kesalahan.

Nama folder sumber dapat menyertakan wildcard (*). Karakter wildcard diterima hanya setelah pemisah jalur file terakhir (/atau\). Jika karakter wildcard disertakan dalam nama folder sumber, semua folder yang cocok dengan wildcard akan disalin ke folder tujuan. Jika Anda ingin menyalin lebih dari satu folder dengan menggunakan karakter wildcard, input ke `destination` opsi harus diakhiri dengan pemisah jalur file (/atau\), yang menunjukkan bahwa input tujuan adalah folder.

Jika nama folder tujuan berbeda dari nama folder sumber, Anda dapat menentukan nama folder tujuan menggunakan `destination` opsi. Jika Anda tidak menentukan nama folder tujuan, nama folder sumber digunakan untuk membuat folder tujuan. Teks apa pun yang mengikuti pemisah jalur file terakhir (/atau\), diperlakukan sebagai nama folder. Jika Anda ingin menggunakan nama folder yang sama dengan folder sumber, maka input `destination` opsi harus diakhiri dengan pemisah jalur file (/atau\).

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk membuat folder di folder yang ditentukan.
- Folder sumber tidak ada saat runtime.
- Sudah ada folder dengan nama folder yang ditentukan dan `overwrite` opsi diatur ke `false`.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
<code>source</code>	Jalur folder sumber.	Tali	Ya	T/A	N/A	Ya
<code>destinasi</code>	Jalur folder tujuan.	Tali	Ya	T/A	N/A	Ya
<code>overwrite</code>	Ketika diatur ke false, folder tujuan tidak akan diganti ketika sudah ada folder di lokasi yang ditentukan dengan nama yang ditentukan.	Boolean	Tidak	<code>true</code>	T/A	Ya

Contoh masukan: salin folder (Linux)

```
name: CopyingAFolderLinux
action: CopyFolder
inputs:
  - source: /Sample/MyFolder/SampleFolder
    destination: /MyFolder/destinationFolder
```

Contoh masukan: salin folder (Windows)

```
name: CopyingAFolderWindows
action: CopyFolder
inputs:
  - source: C:\Sample\MyFolder\SampleFolder
    destination: C:\MyFolder\destinationFolder
```

Contoh masukan: salin folder menggunakan nama folder sumber (Linux)

```
name: CopyingFolderSourceFolderNameLinux
action: CopyFolder
inputs:
  - source: /Sample/MyFolder/SourceFolder
    destination: /MyFolder/
```

Contoh masukan: salin folder menggunakan nama folder sumber (Windows)

```
name: CopyingFolderSourceFolderNameWindows
action: CopyFolder
inputs:
  - source: C:\Sample\MyFolder\SampleFolder
    destination: C:\MyFolder\
```

Contoh masukan: salin folder menggunakan karakter wildcard (Linux)

```
name: CopyingFoldersWithWildCardLinux
action: CopyFolder
inputs:
  - source: /Sample/MyFolder/Sample*
    destination: /MyFolder/
```

Contoh masukan: salin folder menggunakan karakter wildcard (Windows)

```
name: CopyingFoldersWithWildCardWindows
action: CopyFolder
inputs:
  - source: C:\Sample\MyFolder\Sample*
    destination: C:\MyFolder\
```

Contoh masukan: salin folder tanpa menimpa (Linux)

```
name: CopyingFoldersWithoutOverwriteLinux
```

```
action: CopyFolder
inputs:
  - source: /Sample/MyFolder/SourceFolder
    destination: /MyFolder/destinationFolder
    overwrite: false
```

Contoh masukan: salin folder tanpa menimpa (Windows)

```
name: CopyingFoldersWithoutOverwrite
action: CopyFolder
inputs:
  - source: C:\Sample\MyFolder\SourceFolder
    destination: C:\MyFolder\destinationFolder
    overwrite: false
```

Output

Tidak ada.

CreateFile

Modul CreateFile tindakan membuat file di lokasi tertentu. Secara default, jika diperlukan, modul juga secara rekursif membuat folder induk.

Jika file sudah ada di folder yang ditentukan, modul tindakan, secara default, memotong atau menimpa file yang ada. Anda dapat mengganti perilaku default ini dengan menyetel opsi `overwrite` ke `false`. Ketika opsi `overwrite` diatur ke `false`, dan sudah ada file di lokasi yang ditentukan dengan nama yang ditentukan, modul tindakan akan mengembalikan kesalahan.

Jika nilai encoding file berbeda dari nilai encoding (`utf-8`) default, maka Anda dapat menentukan nilai encoding file dengan menggunakan opsi `encoding`. Secara default, `utf-16` dan `utf-32` diasumsikan menggunakan pengkodean endian kecil.

`owner`, `group`, dan `permissions` merupakan input opsional. Input untuk `permissions` harus berupa nilai string. File dibuat dengan nilai default bila tidak disediakan. Opsi ini tidak didukung pada platform Windows. Modul tindakan ini memvalidasi dan mengembalikan kesalahan jika `owner`, `group`, dan `permissions` opsi digunakan pada platform Windows.

Modul tindakan ini dapat membuat file dengan izin yang ditentukan oleh umask nilai default sistem operasi. Anda harus menetapkan umask nilai jika Anda ingin mengganti nilai default.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk membuat file atau folder di folder induk yang ditentukan.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
path	Jalur file.	Tali	Ya	T/A	N/A	Ya
content	Isi file.	String	Tidak	N/A	N/A	Ya
encoding	Standar pengkodean.	String	Tidak	utf8	utf8,utf-8-LE,utf-16-LE,utf16-be,utf-16-BE,utf32,utf-32-LE,utf-32-LE,utf32-be, dan utf-32-BE . Nilai opsi pengkodean tidak peka huruf besar/kecil.	Ya
owner	Nama pengguna atau ID.	String	Tidak	N/A	T/A	Tidak didukung di Windows.

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
group	Nama grup atau ID.	String	Tidak	Pengguna saat ini.	T/A	Tidak didukung di Windows.
permissions	Izin file.	String	Tidak	0666	T/A	Tidak didukung di Windows.
overwrite	Jika nama file yang ditentukan sudah ada, setelah nilai ini untuk false mencegah file terpotong atau ditimpa secara default.	Boolean	Tidak	true	T/A	Ya

Contoh masukan: buat file tanpa menimpa (Linux)

```
name: CreatingFileWithoutOverwriteLinux
action: CreateFile
inputs:
  - path: /home/UserName/Sample.txt
    content: ABCD\nRandom\tvalues
    overwrite: false
```

Contoh masukan: buat file tanpa menimpa (Windows)

```
name: CreatingFileWithoutOverwriteWindows
action: CreateFile
inputs:
  - path: C:\Temp\Sample.txt
    content: ABCD\nRandom\tvalues
    overwrite: false
```

Contoh masukan: buat file dengan properti file

```
name: CreatingFileWithFileProperties
action: CreateFile
inputs:
  - path: SampleFolder/Sample.txt
    content: ABCD\nRandom\tvalues
    encoding: UTF-16
    owner: Ubuntu
    group: UbuntuGroup
    permissions: 0777
  - path: SampleFolder/SampleFile.txt
    permissions: 755
  - path: SampleFolder/TextFile.txt
    encoding: UTF-16
    owner: root
    group: rootUserGroup
```

Contoh masukan: buat file tanpa properti file

```
name: CreatingFileWithoutFileProperties
action: CreateFile
inputs:
  - path: ./Sample.txt
  - path: Sample1.txt
```

Contoh masukan: buat file kosong untuk melewati bagian dalam skrip pembersihan Linux

```
name: CreateSkipCleanupfile
action: CreateFile
inputs:
  - path: <skip section file name>
```

Untuk informasi selengkapnya, silakan lihat [Ganti skrip pembersihan Linux](#)

Output

Tidak ada.

CreateFolder

Modul `CreateFolder` tindakan membuat folder di lokasi tertentu. Secara default, jika diperlukan, modul juga secara rekursif membuat folder induk.

Jika folder sudah ada di folder yang ditentukan, modul tindakan, secara default, memotong atau menimpa folder yang ada. Anda dapat mengganti perilaku default ini dengan menyetel opsi `overwrite` ke `false`. Ketika opsi `overwrite` diatur ke `false`, dan sudah ada folder di lokasi yang ditentukan dengan nama yang ditentukan, modul tindakan akan mengembalikan kesalahan.

`owner`, `group`, dan `permissions` merupakan input opsional. Input untuk `permissions` harus berupa nilai string. Opsi ini tidak didukung pada platform Windows. Modul tindakan ini memvalidasi dan mengembalikan kesalahan jika `owner`, `group`, dan `permissions` opsi digunakan pada platform Windows.

Modul tindakan ini dapat membuat folder dengan izin yang ditentukan oleh `umask` nilai default sistem operasi. Anda harus menetapkan `umask` nilai jika Anda ingin mengganti nilai default.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk membuat folder di lokasi yang ditentukan.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
<code>path</code>	Jalur folder.	Tali	Ya	T/A	N/A	Ya
<code>owner</code>	Nama pengguna atau ID.	String	Tidak	Pengguna saat ini.	T/A	Tidak didukung di Windows.

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
group	Nama grup atau ID.	String	Tidak	Kelompok pengguna saat ini.	T/A	Tidak didukung di Windows.
permissions	Izin folder.	String	Tidak	0777	T/A	Tidak didukung di Windows.
overwrite	Jika nama file yang ditentukan sudah ada, setelah nilai ini untuk false mencegah file terpotong atau ditimpa secara default.	Boolean	Tidak	true	T/A	Ya

Contoh masukan: buat folder (Linux)

```
name: CreatingFolderLinux
action: CreateFolder
inputs:
  - path: /Sample/MyFolder/
```

Contoh masukan: buat folder (Windows)

```
name: CreatingFolderWindows
action: CreateFolder
```



```
inputs:  
  - path: C:\MyFolder
```

Contoh masukan: buat folder yang menentukan properti folder

```
name: CreatingFolderWithFolderProperties  
action: CreateFolder  
inputs:  
  - path: /Sample/MyFolder/Sample/  
    owner: SampleOwnerName  
    group: SampleGroupName  
    permissions: 0777  
  - path: /Sample/MyFolder/SampleFolder/  
    permissions: 777
```

Contoh masukan: buat folder yang menimpa folder yang ada, jika ada.

```
name: CreatingFolderWithOverwrite  
action: CreateFolder  
inputs:  
  - path: /Sample/MyFolder/Sample/  
    overwrite: true
```

Output

Tidak ada.

CreateSymlink

Modul CreateSymlink tindakan membuat tautan simbolis, atau file yang berisi referensi ke file lain. Modul ini tidak didukung pada platform Windows.

Masukan untuk `target` opsi `path` dan dapat berupa jalur absolut atau relatif. Jika input untuk `path` opsi adalah jalur relatif, itu diganti dengan jalur absolut saat tautan dibuat.

Secara default, ketika tautan dengan nama yang ditentukan sudah ada di folder yang ditentukan, modul tindakan mengembalikan kesalahan. Anda dapat mengganti perilaku default ini dengan menyetel `force` opsi ke `true`. Ketika `force` opsi diatur ke `true`, modul akan menimpa tautan yang ada.

Jika folder induk tidak ada, modul tindakan akan membuat folder secara rekursif, secara default.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- File target tidak ada saat runtime.
- Sebuah file link nonsymbolic dengan nama yang ditentukan sudah ada.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
path	Jalur file.	Tali	Ya	T/A	T/A	Tidak didukung di Windows.
target	Jalur file target yang ditunjuk oleh tautan simbolis.	Tali	Ya	T/A	T/A	Tidak didukung di Windows.
force	Memaksa pembuatan tautan ketika tautan dengan nama yang sama sudah ada.	Boolean	Tidak	false	T/A	Tidak didukung di Windows.

Contoh masukan: buat tautan simbolis yang memaksa pembuatan tautan

```
name: CreatingSymbolicLinkWithForce
action: CreateSymlink
inputs:
```

```
- path: /Folder2/Symboliclink.txt
  target: /Folder/Sample.txt
  force: true
```

Contoh masukan: buat tautan simbolis yang tidak memaksa pembuatan tautan

```
name: CreatingSymbolicLinkWithOutForce
action: CreateSymlink
inputs:
  - path: Symboliclink.txt
    target: /Folder/Sample.txt
```

Output

Tidak ada.

DeleteFile

Modul DeleteFile tindakan menghapus file atau file di lokasi tertentu.

Input dari path harus berupa path file yang valid atau path file dengan karakter wild card (*) dalam nama file. Ketika karakter wildcard ditentukan dalam nama file, semua file dalam folder yang sama yang cocok dengan wildcard akan dihapus.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk melakukan operasi penghapusan.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
path	Jalur file.	Tali	Ya	T/A	N/A	Ya

Contoh masukan: hapus satu file (Linux)

```
name: DeletingSingleFileLinux
```

```
action: DeleteFile
inputs:
  - path: /SampleFolder/MyFolder/Sample.txt
```

Contoh masukan: hapus satu file (Windows)

```
name: DeletingSingleFileWindows
action: DeleteFile
inputs:
  - path: C:\SampleFolder\MyFolder\Sample.txt
```

Contoh masukan: hapus file yang diakhiri dengan “log” (Linux)

```
name: DeletingFileEndingWithLogLinux
action: DeleteFile
inputs:
  - path: /SampleFolder/MyFolder/*log
```

Contoh masukan: hapus file yang diakhiri dengan “log” (Windows)

```
name: DeletingFileEndingWithLogWindows
action: DeleteFile
inputs:
  - path: C:\SampleFolder\MyFolder\*log
```

Contoh masukan: hapus semua file dalam folder tertentu (Linux)

```
name: DeletingAllFilesInAFolderLinux
action: DeleteFile
inputs:
  - path: /SampleFolder/MyFolder/*
```

Contoh masukan: hapus semua file dalam folder tertentu (Windows)

```
name: DeletingAllFilesInAFolderWindows
action: DeleteFile
inputs:
  - path: C:\SampleFolder\MyFolder\*
```

Output

Tidak ada.

DeleteFolder

Modul DeleteFolder tindakan menghapus folder.

Jika folder tidak kosong, Anda harus mengatur `force` opsi `true` untuk menghapus folder dan isinya. Jika Anda tidak mengatur `force` opsi `true`, dan folder yang Anda coba hapus tidak kosong, modul tindakan mengembalikan kesalahan. Nilai default dari `force` opsi ini adalah `false`.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk melakukan operasi penghapusan.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
<code>path</code>	Jalur folder.	Tali	Ya	T/A	N/A	Ya
<code>force</code>	Menghapus folder apakah folder kosong atau tidak.	Boolean	Tidak	<code>false</code>	T/A	Ya

Contoh masukan: hapus folder yang tidak kosong menggunakan **force** opsi (Linux)

```
name: DeletingFolderWithForceOptionLinux
action: DeleteFolder
inputs:
  - path: /Sample/MyFolder/Sample/
    force: true
```

Contoh masukan: hapus folder yang tidak kosong menggunakan **force** opsi (Windows)

```
name: DeletingFolderWithForceOptionWindows
action: DeleteFolder
inputs:
  - path: C:\Sample\MyFolder\Sample\
    force: true
```

Contoh masukan: hapus folder (Linux)

```
name: DeletingFolderWithOutForceLinux
action: DeleteFolder
inputs:
  - path: /Sample/MyFolder/Sample/
```

Contoh masukan: hapus folder (Windows)

```
name: DeletingFolderWithOutForce
action: DeleteFolder
inputs:
  - path: C:\Sample\MyFolder\Sample\
```

Output

Tidak ada.

ListFiles

Modul ListFilestindakan mencantumkan file dalam folder tertentu. Ketika opsi rekursif diatur ke `true`, itu mencantumkan file dalam subfolder. Modul ini tidak mencantumkan file dalam subfolder secara default.

Untuk membuat daftar semua file dengan nama yang cocok dengan pola tertentu, gunakan `fileNamePattern` opsi untuk menyediakan pola. `fileNamePattern` opsi menerima nilai wildcard (*). Ketika `fileNamePattern` disediakan, semua file yang cocok dengan format nama file yang ditentukan dikembalikan.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Folder yang ditentukan tidak ada saat runtime.
- Anda tidak memiliki izin untuk membuat file atau folder di folder induk yang ditentukan.

- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
path	Jalur folder.	Tali	Ya	T/A	N/A	Ya
fileNamePattern	Pola yang cocok untuk mencantumkan semua file dengan nama yang cocok dengan pola.	String	Tidak	N/A	N/A	Ya
recursive	Daftar file dalam folder secara rekursif.	Boolean	Tidak	false	T/A	Ya

Contoh masukan: daftar file dalam folder tertentu (Linux)

```
name: ListingFilesInSampleFolderLinux
action: ListFiles
inputs:
  - path: /Sample/MyFolder/Sample
```

Contoh masukan: daftar file dalam folder tertentu (Windows)

```
name: ListingFilesInSampleFolderWindows
```

```
action: ListFiles
inputs:
  - path: C:\Sample\MyFolder\Sample
```

Contoh masukan: daftar file yang diakhiri dengan “log” (Linux)

```
name: ListingFilesWithEndingWithLogLinux
action: ListFiles
inputs:
  - path: /Sample/MyFolder/
    fileNamePattern: *log
```

Contoh masukan: daftar file yang diakhiri dengan “log” (Windows)

```
name: ListingFilesWithEndingWithLogWindows
action: ListFiles
inputs:
  - path: C:\Sample\MyFolder\
    fileNamePattern: *log
```

Contoh masukan: daftar file secara rekursif

```
name: ListingFilesRecursively
action: ListFiles
inputs:
  - path: /Sample/MyFolder/
    recursive: true
```

Output

Primitif	Deskripsi	Tipe				
files	Daftar file.	String				

Contoh keluaran

```
{
  "files": "/sample1.txt,/sample2.txt,/sample3.txt"
}
```


MoveFile

Modul MoveFile tindakan memindahkan file dari sumber yang ditentukan ke tujuan yang ditentukan.

Jika file sudah ada di folder yang ditentukan, modul tindakan, secara default, menimpa file yang ada. Anda dapat mengganti perilaku default ini dengan menyetel opsi `overwrite` ke `false`. Ketika opsi `overwrite` diatur ke `false`, dan sudah ada file di lokasi yang ditentukan dengan nama yang ditentukan, modul tindakan akan mengembalikan kesalahan. Opsi ini bekerja sama dengan `mv` perintah di Linux, yang menimpa secara default.

Nama file sumber dapat menyertakan wildcard (*). Karakter wildcard diterima hanya setelah pemisah jalur file terakhir (/atau\). Jika karakter wildcard disertakan dalam nama file sumber, semua file yang cocok dengan wildcard disalin ke folder tujuan. Jika Anda ingin memindahkan lebih dari satu file dengan menggunakan karakter wildcard, input ke `destination` opsi harus diakhiri dengan pemisah jalur file (/atau\), yang menunjukkan bahwa input tujuan adalah folder.

Jika nama file tujuan berbeda dari nama file sumber, Anda dapat menentukan nama file tujuan menggunakan `destination` opsi. Jika Anda tidak menentukan nama file tujuan, nama file sumber digunakan untuk membuat file tujuan. Setiap teks yang mengikuti pemisah jalur file terakhir (/atau\), diperlakukan sebagai nama file. Jika Anda ingin menggunakan nama file yang sama dengan file sumber, maka input `destination` opsi harus diakhiri dengan pemisah jalur file (/atau\).

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk membuat file di folder yang ditentukan.
- File sumber tidak ada saat runtime.
- Sudah ada folder dengan nama file yang ditentukan dan `overwrite` opsi diatur ke `false`.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
<code>source</code>	Jalur file sumber.	Tali	Ya	T/A	N/A	Ya

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
destination	Jalur file tujuan.	Tali	Ya	T/A	N/A	Ya
overwrite	Ketika diatur ke false, file tujuan tidak akan diganti ketika sudah ada file di lokasi yang ditentukan dengan nama yang ditentukan.	Boolean	Tidak	true	T/A	Ya

Contoh masukan: memindahkan file (Linux)

```
name: MovingAFileLinux
action: MoveFile
inputs:
  - source: /Sample/MyFolder/Sample.txt
    destination: /MyFolder/destinationFile.txt
```

Contoh masukan: memindahkan file (Windows)

```
name: MovingAFileWindows
action: MoveFile
inputs:
  - source: C:\Sample\MyFolder\Sample.txt
    destination: C:\MyFolder\destinationFile.txt
```

Contoh masukan: memindahkan file menggunakan nama file sumber (Linux)

```
name: MovingFileWithSourceFileNameLinux
action: MoveFile
inputs:
  - source: /Sample/MyFolder/Sample.txt
    destination: /MyFolder/
```

Contoh masukan: memindahkan file menggunakan nama file sumber (Windows)

```
name: MovingFileWithSourceFileNameWindows
action: MoveFile
inputs:
  - source: C:\Sample\MyFolder\Sample.txt
    destination: C:\MyFolder
```

Contoh masukan: memindahkan file menggunakan karakter wildcard (Linux)

```
name: MovingFilesWithWildCardLinux
action: MoveFile
inputs:
  - source: /Sample/MyFolder/Sample*
    destination: /MyFolder/
```

Contoh masukan: memindahkan file menggunakan karakter wildcard (Windows)

```
name: MovingFilesWithWildCardWindows
action: MoveFile
inputs:
  - source: C:\Sample\MyFolder\Sample*
    destination: C:\MyFolder
```

Contoh masukan: memindahkan file tanpa menimpa (Linux)

```
name: MovingFilesWithoutOverwriteLinux
action: MoveFile
inputs:
  - source: /Sample/MyFolder/Sample.txt
    destination: /MyFolder/destinationFile.txt
    overwrite: false
```

Contoh masukan: memindahkan file tanpa menimpa (Windows)

```
name: MovingFilesWithoutOverwrite
action: MoveFile
inputs:
  - source: C:\Sample\MyFolder\Sample.txt
    destination: C:\MyFolder\destinationFile.txt
    overwrite: false
```

Output

Tidak ada.

MoveFolder

Modul `MoveFolder` tindakan memindahkan folder dari sumber yang ditentukan ke tujuan yang ditentukan. Input untuk `source` opsi adalah folder untuk dipindahkan, dan input ke `destination` opsi adalah folder tempat konten folder sumber dipindahkan.

Jika folder induk tujuan atau input ke `destination` opsi tidak ada saat runtime, perilaku default modul adalah membuat folder secara rekursif di tujuan yang ditentukan.

Jika folder dengan folder yang sama dengan folder sumber sudah ada di folder tujuan, modul tindakan, secara default, menimpa folder yang ada. Anda dapat mengganti perilaku default ini dengan menyetel opsi `overwrite` ke `false`. Ketika opsi `overwrite` diatur ke `false`, dan sudah ada folder di lokasi yang ditentukan dengan nama yang ditentukan, modul tindakan akan mengembalikan kesalahan.

Nama folder sumber dapat menyertakan wildcard (*). Karakter wildcard diterima hanya setelah pemisah jalur file terakhir (/atau\). Jika karakter wildcard disertakan dalam nama folder sumber, semua folder yang cocok dengan wildcard akan disalin ke folder tujuan. Jika Anda ingin memindahkan lebih dari satu folder dengan menggunakan karakter wildcard, input ke `destination` opsi harus diakhiri dengan pemisah jalur file (/atau\), yang menunjukkan bahwa input tujuan adalah folder.

Jika nama folder tujuan berbeda dari nama folder sumber, Anda dapat menentukan nama folder tujuan menggunakan `destination` opsi. Jika Anda tidak menentukan nama folder tujuan, nama folder sumber digunakan untuk membuat folder tujuan. Teks apa pun yang mengikuti pemisah jalur file terakhir (/atau\), diperlakukan sebagai nama folder. Jika Anda ingin menggunakan nama folder

yang sama dengan folder sumber, maka input `destination` opsi harus diakhiri dengan pemisah jalur file (`/`atau`\`).

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk membuat folder di folder tujuan.
- Folder sumber tidak ada saat runtime.
- Sudah ada folder dengan nama yang ditentukan dan `overwrite` opsi diatur ke `false`.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
<code>source</code>	Jalur folder sumber.	Tali	Ya	T/A	N/A	Ya
<code>destination</code>	Jalur folder tujuan.	Tali	Ya	T/A	N/A	Ya
<code>overwrite</code>	Ketika diatur ke <code>false</code> , folder tujuan tidak akan diganti ketika sudah ada folder di lokasi yang ditentukan dengan nama yang ditentukan.	Boolean	Tidak	<code>true</code>	T/A	Ya

Contoh masukan: pindahkan folder (Linux)

```
name: MovingAFolderLinux
action: MoveFolder
inputs:
  - source: /Sample/MyFolder/SourceFolder
    destination: /MyFolder/destinationFolder
```

Contoh masukan: pindahkan folder (Windows)

```
name: MovingAFolderWindows
action: MoveFolder
inputs:
  - source: C:\Sample\MyFolder\SourceFolder
    destination: C:\MyFolder\destinationFolder
```

Contoh masukan: pindahkan folder menggunakan nama folder sumber (Linux)

```
name: MovingFolderWithSourceFolderNameLinux
action: MoveFolder
inputs:
  - source: /Sample/MyFolder/SampleFolder
    destination: /MyFolder/
```

Contoh masukan: pindahkan folder menggunakan nama folder sumber (Windows)

```
name: MovingFolderWithSourceFolderNameWindows
action: MoveFolder
inputs:
  - source: C:\Sample\MyFolder\SampleFolder
    destination: C:\MyFolder\
```

Contoh masukan: pindahkan folder menggunakan karakter wildcard (Linux)

```
name: MovingFoldersWithWildCardLinux
action: MoveFolder
inputs:
  - source: /Sample/MyFolder/Sample*
    destination: /MyFolder/
```

Contoh masukan: pindahkan folder menggunakan karakter wildcard (Windows)

```
name: MovingFoldersWithWildCardWindows
action: MoveFolder
inputs:
  - source: C:\Sample\MyFolder\Sample*
    destination: C:\MyFolder\
```

Contoh masukan: pindahkan folder tanpa menimpa (Linux)

```
name: MovingFoldersWithoutOverwriteLinux
action: MoveFolder
inputs:
  - source: /Sample/MyFolder/SampleFolder
    destination: /MyFolder/destinationFolder
    overwrite: false
```

Contoh masukan: pindahkan folder tanpa menimpa (Windows)

```
name: MovingFoldersWithoutOverwriteWindows
action: MoveFolder
inputs:
  - source: C:\Sample\MyFolder\SampleFolder
    destination: C:\MyFolder\destinationFolder
    overwrite: false
```

Output

Tidak ada.

ReadFile

Modul `ReadFile` tindakan membaca konten file teks dari jenis string. Modul ini dapat digunakan untuk membaca konten file untuk digunakan dalam langkah-langkah selanjutnya melalui rantai atau untuk membaca data ke `console.log` file. Jika jalur yang ditentukan adalah link simbolis, modul ini mengembalikan isi dari file target. Modul ini hanya mendukung file teks.

Jika nilai encoding file berbeda dari nilai encoding (`utf-8`) default, maka Anda dapat menentukan nilai encoding file dengan menggunakan opsi. encoding Secara default, `utf-16` dan `utf-32` diasumsikan menggunakan pengkodean endian kecil.

Secara default, modul ini tidak dapat mencetak konten file ke `console.log` file. Anda dapat mengganti setelan ini dengan menyetel `printFileContent` properti ke `true`.

Modul ini hanya dapat mengembalikan isi file. Itu tidak dapat mengurai file, seperti file Excel atau JSON.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- File tidak ada saat runtime.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
path	Jalur file.	Tali	Ya	T/A	N/A	Ya
encoding	Standar pengkodean.	String	Tidak	utf8	utf8,utf-8-LE,utf-16-LE,utf16-BE,utf-16-BE,,utf32,utf-32-LE,utf-32-LE,utf32-BE, dan utf-32-BE . Nilai opsi pengkodean tidak peka huruf besar/kecil.	Ya
printFileContent	Mencetak konten	Boolean	Tidak	salah	T/A	Ya.

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
	file ke console.log file.					

Contoh masukan: membaca file (Linux)

```
name: ReadingFileLinux
action: ReadFile
inputs:
  - path: /home/UserName/SampleFile.txt
```

Contoh masukan: membaca file (Windows)

```
name: ReadingFileWindows
action: ReadFile
inputs:
  - path: C:\Windows\WindowsUpdate.log
```

Contoh masukan: membaca file dan menentukan standar pengkodean

```
name: ReadingFileWithFileEncoding
action: ReadFile
inputs:
  - path: /FolderName/SampleFile.txt
    encoding: UTF-32
```

Contoh masukan: membaca file dan mencetak ke **console.log** file

```
name: ReadingFileToConsole
action: ReadFile
inputs:
  - path: /home/UserName/SampleFile.txt
    printFileContent: true
```

Output

Bidang	Deskripsi	Tipe
content	Konten file.	string

Contoh keluaran

```
{
  "content" : "The file content"
}
```

SetFileEncoding

Modul SetFileEncoding tindakan memodifikasi properti encoding dari file yang ada. Modul ini dapat mengonversi pengkodean file dari utf-8 ke standar pengkodean yang ditentukan. Secara default, utf-16 dan utf-32 diasumsikan sebagai pengkodean endian kecil.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk melakukan modifikasi yang ditentukan.
- File tidak ada saat runtime.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
path	Jalur file.	Tali	Ya	T/A	N/A	Ya
encoding	Standar pengkodean.	String	Tidak	utf8	utf8,utf-8,LE,utf-16-LE,utf16-BE,utf-16-	Ya

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
					BE, utf32, utf-32-LE, utf-32-LE, utf32-BE, dan utf-32-BE . Nilai opsi pengkodean tidak peka huruf besar/kecil.	

Contoh masukan: mengatur properti pengkodean file

```
name: SettingFileEncodingProperty
action: SetFileEncoding
inputs:
  - path: /home/UserName/SampleFile.txt
    encoding: UTF-16
```

Output

Tidak ada.

SetFileOwner

Modul SetFileOwnertindakan memodifikasi properti owner dan group pemilik file yang ada. Jika file yang ditentukan adalah tautan simbolis, modul memodifikasi owner properti file sumber. Modul ini tidak didukung pada platform Windows.

Modul ini menerima nama pengguna dan grup sebagai input. Jika nama grup tidak disediakan, modul akan menetapkan pemilik grup file ke grup tempat pengguna berada.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk melakukan modifikasi yang ditentukan.
- Nama pengguna atau grup yang ditentukan tidak ada saat runtime.
- File tidak ada saat runtime.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
path	Jalur file.	Tali	Ya	T/A	T/A	Tidak didukung di Windows.
owner	Nama pengguna.	string	Ya	T/A	T/A	Tidak didukung di Windows.
group	Nama grup pengguna.	String	Tidak	Nama grup tempat pengguna berada.	T/A	Tidak didukung di Windows.

Contoh masukan: mengatur properti pemilik file tanpa menentukan nama grup pengguna

```
name: SettingFileOwnerPropertyNoGroup
action: SetFileOwner
inputs:
  - path: /home/UserName/SampleText.txt
    owner: LinuxUser
```

Contoh masukan: mengatur properti pemilik file dengan menentukan pemilik dan grup pengguna

```
name: SettingFileOwnerProperty
action: SetFileOwner
```

```
inputs:
  - path: /home/UserName/SampleText.txt
    owner: LinuxUser
    group: LinuxUserGroup
```

Output

Tidak ada.

SetFolderOwner

Modul `SetFolderOwner` tindakan secara rekursif memodifikasi properti `owner` dan `group` pemilik folder yang ada. Secara default, modul dapat memodifikasi kepemilikan untuk semua konten dalam folder. Anda dapat mengatur `recursive` opsi `false` untuk mengganti perilaku ini. Modul ini tidak didukung pada platform Windows.

Modul ini menerima nama pengguna dan grup sebagai input. Jika nama grup tidak disediakan, modul akan menetapkan pemilik grup file ke grup tempat pengguna berada.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk melakukan modifikasi yang ditentukan.
- Nama pengguna atau grup yang ditentukan tidak ada saat runtime.
- Folder tidak ada saat runtime.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
<code>path</code>	Jalur folder.	Tali	Ya	T/A	T/A	Tidak didukung di Windows.
<code>owner</code>	Nama pengguna.	string	Ya	T/A	T/A	Tidak didukung di Windows.

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
<code>group</code>	Nama grup pengguna.	String	Tidak	Nama grup tempat pengguna berada.	T/A	Tidak didukung di Windows.
<code>recursive</code>	Mengganti perilaku default modifikasi kepemilikan untuk semua konten folder saat disetel ke <code>false</code>	Boolean	Tidak	<code>true</code>	T/A	Tidak didukung di Windows.

Contoh masukan: mengatur properti pemilik folder tanpa menentukan nama grup pengguna

```
name: SettingFolderPropertyWithoutGroup
action: SetFolderOwner
inputs:
  - path: /SampleFolder/
    owner: LinuxUser
```

Contoh masukan: setel properti pemilik folder tanpa mengesampingkan kepemilikan semua konten dalam folder

```
name: SettingFolderPropertyWithoutRecursively
action: SetFolderOwner
inputs:
  - path: /SampleFolder/
    owner: LinuxUser
```

```
recursive: false
```

Contoh masukan: mengatur properti kepemilikan file dengan menentukan nama grup pengguna

```
name: SettingFolderPropertyWithGroup
action: SetFolderOwner
inputs:
  - path: /SampleFolder/
    owner: LinuxUser
    group: LinuxUserGroup
```

Output

Tidak ada.

SetFilePermissions

Modul SetFilePermissionstindakan memodifikasi file permissions yang ada. Modul ini tidak didukung pada platform Windows.

Input untuk permissions harus berupa nilai string.

Modul tindakan ini dapat membuat file dengan izin yang ditentukan oleh nilai umask default dari sistem operasi. Anda harus menetapkan umask nilai jika Anda ingin mengganti nilai default.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk melakukan modifikasi yang ditentukan.
- File tidak ada saat runtime.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
path	Jalur file.	Tali	Ya	T/A	T/A	Tidak didukung di Windows.

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
permissions	Izin file.	Tali	Ya	T/A	T/A	Tidak didukung di Windows.

Contoh masukan: memodifikasi izin file

```
name: ModifyingFilePermissions
action: SetFilePermissions
inputs:
  - path: /home/UserName/SampleFile.txt
    permissions: 766
```

Output

Tidak ada.

SetFolderPermissions

Modul `SetFolderPermissions` tindakan secara rekursif memodifikasi folder `permissions` yang ada dan semua subfile dan subfoldernya. Secara default, modul ini dapat memodifikasi izin untuk semua isi folder yang ditentukan. Anda dapat mengatur `recursive` opsi `false` untuk mengganti perilaku ini. Modul ini tidak didukung pada platform Windows.

Input untuk `permissions` harus berupa nilai string.

Modul tindakan ini dapat memodifikasi izin sesuai dengan nilai `umask` default dari sistem operasi. Anda harus menetapkan `umask` nilai jika Anda ingin mengganti nilai default.

Modul tindakan mengembalikan kesalahan ketika hal berikut terjadi:

- Anda tidak memiliki izin untuk melakukan modifikasi yang ditentukan.
- Folder tidak ada saat runtime.
- Modul tindakan mengalami kesalahan saat melakukan operasi.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
<code>path</code>	Jalur folder.	Tali	Ya	T/A	T/A	Tidak didukung di Windows.
<code>permissions</code>	Izin folder.	Tali	Ya	T/A	T/A	Tidak didukung di Windows.
<code>recursive</code>	Mengganti perilaku default memodifikasi izin untuk semua konten folder saat disetel ke <code>false</code>	Boolean	Tidak	<code>true</code>	T/A	Tidak didukung di Windows.

Contoh masukan: atur izin folder

```
name: SettingFolderPermissions
action: SetFolderPermissions
inputs:
  - path: SampleFolder/
    permissions: 0777
```

Contoh masukan: atur izin folder tanpa memodifikasi izin untuk semua isi folder

```
name: SettingFolderPermissionsNoRecursive
action: SetFolderPermissions
inputs:
```

```
- path: /home/UserName/SampleFolder/
  permissions: 777
  recursive: false
```

Output

Tidak ada.

Tindakan instalasi perangkat lunak

Bagian ini menjelaskan modul tindakan yang melakukan perintah dan instruksi tindakan instalasi perangkat lunak.

Persyaratan IAM

Jika jalur unduhan instalasi Anda adalah URI S3, maka peran IAM yang Anda kaitkan dengan profil instans Anda harus memiliki izin untuk menjalankan modul `S3Download` tindakan. Untuk memberikan izin yang diperlukan, lampirkan kebijakan `S3:GetObject` IAM ke peran IAM yang terkait dengan profil instans Anda, dan tentukan jalur untuk bucket Anda. Misalnya, `arn:aws:s3:::BucketName/*`).

Input MSI Kompleks

Jika string masukan Anda berisi karakter kutipan ganda ("), Anda harus menggunakan salah satu metode berikut untuk memastikan bahwa string tersebut ditafsirkan dengan benar:

- Anda dapat menggunakan tanda kutip tunggal (') di bagian luar string Anda, untuk menampungnya, dan tanda kutip ganda (") di dalam string Anda, seperti yang ditunjukkan pada contoh berikut.

```
properties:
  COMPANYNAME: '"Acme "'Widgets'" and "'Gizmos.'"'
```

Dalam hal ini, jika Anda perlu menggunakan tanda kutip di dalam string Anda, Anda harus menghindarinya. Ini berarti menggunakan kutipan tunggal lain (') sebelum apostrof.

- Anda dapat menggunakan tanda kutip ganda (") di bagian luar string Anda, untuk menampungnya. Dan Anda dapat melarikan diri dari tanda kutip ganda di dalam string Anda, menggunakan karakter garis miring terbalik (\), seperti yang ditunjukkan pada contoh berikut.

```
properties:
  COMPANYNAME: "\"Acme \"\"Widgets\"\"" and "\"Gizmos.\"\"\""
```

Kedua metode ini meneruskan nilai `COMPANYNAME="Acme ""Widgets"" and ""Gizmos.""` ke `msiexec` perintah.

Modul tindakan instalasi perangkat lunak

- [InstalMSI](#)
- [UninstallMSI](#)

InstalMSI

Modul `InstalMSI` tindakan menginstal aplikasi Windows menggunakan file MSI. Anda dapat menentukan file MSI menggunakan jalur lokal, URI objek S3, atau URL web. Opsi reboot mengkonfigurasi perilaku reboot sistem.

`AWSTOEmenghasilkan` `msiexec` perintah berdasarkan parameter input untuk modul tindakan. Nilai untuk parameter input `path` (lokasi file MSI) dan `LogFile` (lokasi file log) harus diapit tanda kutip (“).

Kode keluar MSI berikut dianggap berhasil:

- 0 (Sukses)
- 1614 (`ERROR_PRODUCT_UNINSTALLED`)
- 1641 (Reboot Dimulai)
- 3010 (Diperlukan Reboot)

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
<code>path</code>	Tentukan lokasi file MSI menggunakan salah satu dari berikut ini: <ul style="list-style-type: none"> • 	Tali	Ya	T/A	T/A

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
	<p>Jalur file lokal. Jalannya bisa absolut atau relatif</p> <ul style="list-style-type: none"> • URI objek S3 yang valid. • URL HTTP/HTTPS web yang valid (disarankan HTTPS) yang mengikuti standar RFC 3986. <p>Ekspresi rantai diperbolehkan.</p>				

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
reboot	<p>Konfigurasi perilaku reboot sistem yang mengikuti keberhasilan menjalankan modul tindakan.</p> <p>Pengaturan:</p> <ul style="list-style-type: none"> • Force— Memulai reboot sistem setelah <code>msiexec</code> perintah berjalan dengan sukses. • Allow— Memulai reboot sistem jika <code>msiexec</code> perintah mengembalikan kode keluar yang 	String	Tidak	Allow	Allow, Force, Skip

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
	<p>menunjukkan an reboot diperlukan.</p> <ul style="list-style-type: none"> • Skip— Log pesan informasi ke <code>console.log</code> file yang menunjukkan bahwa reboot dilewati. Opsi ini mencegah reboot, bahkan jika <code>msiexec</code> perintah mengembalikan kode keluar yang menunjukkan an reboot diperlukan. 				

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
logOptions	<p>Tentukan opsi yang akan digunakan untuk logging instalasi MSI. Bendera yang ditentukan diteruskan ke penginstal MSI, bersama dengan parameter baris /L perintah untuk mengaktifkan logging. Jika tidak ada flag yang ditentukan, AWSTOE gunakan nilai default.</p> <p>Untuk informasi selengkapnya tentang opsi log untuk</p>	String	Tidak	*VX	i,w,e,a,r ,u,c,m,o, p,v,x,+ ,! ,*

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
	MSI, lihat Opsis Baris Perintah dalam dokumentasi produk Penginstal Microsoft Windows.				
logFile	Jalur absolut atau relatif ke lokasi file log. Jika jalur file log tidak ada, itu dibuat. Jika jalur file log tidak disediakan, AWSTOE tidak menyimpan log instalasi MSI.	String	Tidak	N/A	T/A

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
properties	<p>Pasangan nilai kunci properti logging MSI, misalnya: TARGETDIR : "C:\target\location"</p> <p>Catatan: Modifikasi properti berikut tidak diperbolehkan:</p> <ul style="list-style-type: none"> • REBOOT="ReallySuppress" • REINSTALLMODE="ecmus" • REINSTALL="ALL" 	Peta [String] String	Tidak	N/A	T/A

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
<code>ignoreAuthenticodeSignatureErrors</code>	<p>Tandai untuk mengabaikan kesalahan validasi tanda tangan authenticode untuk penginstall yang ditentukan di jalur. <code>Get-AuthenticodeSignaturePerintah</code> ini digunakan untuk memvalidasi installer.</p> <p>Pengaturan:</p> <ul style="list-style-type: none"> <code>true</code>— Kesalahan validasi diabaikan dan penginstall berjalan. <code>false</code>— Kesalahan validasi 	Boolean	Tidak	<code>false</code>	<code>true</code> , <code>false</code>

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
	tidak diabaikan. Penginstallasi berjalan hanya ketika validasi berhasil. Ini adalah perilaku default.				

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
<code>allowUnsignedInstaller</code>	<p>Tandai untuk memungkinkan menjalankan penginstalan yang tidak ditandatangani yang ditentukan di jalur. Get-AuthenticodeSignaturePerintah ini digunakan untuk memvalidasi installer.</p> <p>Pengaturan:</p> <ul style="list-style-type: none"> • <code>true</code>—Mengabaikan <code>NotSigned</code> status yang dikembalikan oleh Get-AuthenticodeSignature perintah 	Boolean	Tidak	<code>false</code>	<code>true</code> , <code>false</code>

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
	<p>dan menjalankan installer.</p> <ul style="list-style-type: none"> • <code>false</code>—Membutuhkan installer untuk ditandatangani. Penginstal yang tidak ditandatangani tidak akan berjalan. Ini adalah perilaku default. 				

Contoh

Contoh berikut menunjukkan variasi bagian input untuk dokumen komponen Anda, tergantung pada jalur instalasi Anda.

Contoh masukan: instalasi jalur dokumen lokal

```
- name: local-path-install
  steps:
    - name: LocalPathInstaller
      action: InstallMSI
      inputs:
        path: C:\sample.msi
        logFile: C:\msilogs\local-path-install.log
        logOptions: '*VX'
```

```
reboot: Allow
properties:
  COMPANYNAME: "Amazon Web Services"
ignoreAuthenticodeSignatureErrors: true
allowUnsignedInstaller: true
```

Contoh masukan: Instalasi jalur Amazon S3

```
- name: s3-path-install
steps:
  - name: S3PathInstaller
    action: InstallMSI
    inputs:
      path: s3://<bucket-name>/sample.msi
      logFile: s3-path-install.log
      reboot: Force
      ignoreAuthenticodeSignatureErrors: false
      allowUnsignedInstaller: true
```

Contoh masukan: instalasi jalur web

```
- name: web-path-install
steps:
  - name: WebPathInstaller
    action: InstallMSI
    inputs:
      path: https://<some-path>/sample.msi
      logFile: web-path-install.log
      reboot: Skip
      ignoreAuthenticodeSignatureErrors: true
      allowUnsignedInstaller: false
```

Output

Berikut ini adalah contoh output dari modul `InstallMSI` tindakan.

```
{
  "logFile": "web-path-install.log",
  "msiExitCode": 0,
  "stdout": ""
}
```

UninstallMSI

Modul `UninstallMSI` tindakan memungkinkan Anda untuk menghapus aplikasi Windows menggunakan file MSI. Anda dapat menentukan lokasi file MSI menggunakan jalur file lokal, URI objek S3, atau URL web. Opsi `reboot` mengkonfigurasi perilaku reboot sistem.

`AWSTOEm` menghasilkan `msiexec` perintah berdasarkan parameter input untuk modul tindakan. Lokasi file MSI (`path`) dan lokasi file log (`logfile`) secara eksplisit diapit tanda kutip ganda (") saat menghasilkan perintah. `msiexec`

Kode keluar MSI berikut dianggap berhasil:

- 0 (Sukses)
- 1605 (ERROR_UNKNOWN_PRODUCT)
- 1614 (ERROR_PRODUCT_UNINSTALLED)
- 1641 (Reboot Dimulai)
- 3010 (Diperlukan Reboot)

Input

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
<code>path</code>	Tentukan lokasi file MSI menggunakan salah satu dari berikut ini: <ul style="list-style-type: none"> • Jalur file lokal. Jalannya bisa absolut atau relatif. 	Tali	Ya	T/A	T/A

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
	<ul style="list-style-type: none">• URI objek S3 yang valid.• URL HTTP/HTTPS web yang valid (disarankan HTTPS) yang mengikuti standar RFC 3986. <p>Ekspresi rantai diperbolehkan.</p>				

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
reboot	<p>Mengkonfigurasi perilaku reboot sistem yang mengikuti keberhasilan menjalankan modul tindakan.</p> <p>Pengaturan:</p> <ul style="list-style-type: none"> • Force—Memulai reboot sistem setelah <code>msiexec</code> perintah berjalan dengan sukses. • Allow—Memulai reboot sistem jika <code>msiexec</code> perintah mengembalikan kode 	String	Tidak	Allow	Allow, Force, Skip

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
	<p>keluar yang menunjukkan an reboot diperlukan.</p> <ul style="list-style-type: none"> • Skip— Log pesan informasi ke <code>console.log</code> file yang menunjukkan an bahwa reboot dilewati. Opsi ini mencegah reboot, bahkan jika <code>msiexec</code> perintah mengembalikan kode keluar yang menunjukkan an reboot diperlukan. 				

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
logOptions	<p>Tentukan opsi yang akan digunakan untuk logging instalasi MSI. Bendera yang ditentukan diteruskan ke penginstal MSI, bersama dengan parameter baris /L perintah untuk mengaktifkan logging. Jika tidak ada flag yang ditentukan, AWSTOE gunakan nilai default.</p> <p>Untuk informasi selengkapnya tentang opsi log untuk</p>	String	Tidak	*VX	i,w,e,a,r ,u,c,m,o, p,v,x,+!, ,*

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
	MSI, lihat Opsis Baris Perintah dalam dokumentasi produk Penginstal Microsoft Windows.				
logFile	Jalur absolut atau relatif ke lokasi file log. Jika jalur file log tidak ada, itu dibuat. Jika jalur file log tidak disediakan, AWSTOE tidak menyimpan log instalasi MSI.	String	Tidak	N/A	T/A

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
properties	<p>Pasangan nilai kunci properti logging MSI, misalnya: TARGETDIR : "C:\target\location"</p> <p>Catatan: Modifikasi properti berikut tidak diperbolehkan:</p> <ul style="list-style-type: none"> • REBOOT="ReallySuppress" • REINSTALLMODE="ecmus" • REINSTALL="ALL" 	Peta [String] String	Tidak	N/A	T/A

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
<code>ignoreAuthenticodeSignatureErrors</code>	<p>Tandai untuk mengabaikan kesalahan validasi tanda tangan authenticode untuk penginstall yang ditentukan di jalur. <code>Get-AuthenticodeSignaturePerintah</code> ini digunakan untuk memvalidasi installer.</p> <p>Pengaturan:</p> <ul style="list-style-type: none"> <code>true</code>— Kesalahan validasi diabaikan dan penginstall berjalan. <code>false</code>— Kesalahan validasi 	Boolean	Tidak	<code>false</code>	<code>true</code> , <code>false</code>

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
	tidak diabaikan. Penginstall berjalan hanya ketika validasi berhasil. Ini adalah perilaku default.				

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
<code>allowUnsignedInstaller</code>	<p>Tandai untuk memungkinkan menjalankan penginstalan yang tidak ditandatangani yang ditentukan di jalur. <code>Get-AuthenticodeSignaturePerintah</code> ini digunakan untuk memvalidasi installer.</p> <p>Pengaturan:</p> <ul style="list-style-type: none"> <code>true</code>—Mengabaikan <code>NotSigned</code> status yang dikembalikan oleh <code>Get-AuthenticodeSignature</code> perintah 	Boolean	Tidak	<code>false</code>	<code>true</code> , <code>false</code>

Primitif	Deskripsi	Tipe	Diperlukan	Nilai default	Nilai yang dapat diterima
	<p>dan menjalankan installer.</p> <ul style="list-style-type: none"> • <code>false</code>—Membutuhkan installer untuk ditandatangani. Penginstal yang tidak ditandatangani tidak akan berjalan. Ini adalah perilaku default. 				

Contoh

Contoh berikut menunjukkan variasi bagian input untuk dokumen komponen Anda, tergantung pada jalur instalasi Anda.

Contoh masukan: hapus instalasi jalur dokumen lokal

```
- name: local-path-uninstall
  steps:
    - name: LocalPathUninstaller
      action: UninstallMSI
      inputs:
        path: C:\sample.msi
        logFile: C:\msilog\local-path-uninstall.log
        logOptions: '*VX'
```

```
reboot: Allow
properties:
  COMPANYNAME: "Amazon Web Services"
ignoreAuthenticodeSignatureErrors: true
allowUnsignedInstaller: true
```

Contoh masukan: hapus instalasi jalur Amazon S3

```
- name: s3-path-uninstall
steps:
  - name: S3PathUninstaller
    action: UninstallMSI
    inputs:
      path: s3://<bucket-name>/sample.msi
      logFile: s3-path-uninstall.log
      reboot: Force
      ignoreAuthenticodeSignatureErrors: false
      allowUnsignedInstaller: true
```

Contoh masukan: hapus instalasi jalur web

```
- name: web-path-uninstall
steps:
  - name: WebPathUninstaller
    action: UninstallMSI
    inputs:
      path: https://<some-path>/sample.msi
      logFile: web-path-uninstall.log
      reboot: Skip
      ignoreAuthenticodeSignatureErrors: true
      allowUnsignedInstaller: false
```

Output

Berikut ini adalah contoh output dari modul `UninstallMSI` tindakan.

```
{
  "logFile": "web-path-uninstall.log",
  "msiExitCode": 0,
  "stdout": ""
}
```

Modul tindakan sistem

Bagian berikut menjelaskan modul tindakan yang melakukan perintah dan instruksi tindakan sistem file.

Modul tindakan sistem

- [Mulai ulang](#)
- [SetRegistry](#)
- [UpdateOS](#)

Mulai ulang

Modul aksi Reboot me-reboot instance. Ini memiliki opsi yang dapat dikonfigurasi untuk menunda dimulainya reboot. Secara default, `delaySeconds` diatur ke 0, yang berarti bahwa tidak ada penundaan. Batas waktu langkah tidak didukung untuk modul tindakan Reboot, karena tidak berlaku saat instance di-boot ulang.

Jika aplikasi dipanggil oleh Agen Systems Manager, itu menyerahkan kode keluar (3010 untuk Windows, 194 untuk Linux) ke Agen Systems Manager. Agen Systems Manager menangani reboot sistem seperti yang dijelaskan dalam [Memulai Ulang Instans Terkelola dari Skrip](#).

Jika aplikasi dipanggil pada host sebagai proses mandiri, ia menyimpan status eksekusi saat ini, mengonfigurasi pemicu auto-run pasca-reboot untuk menjalankan kembali aplikasi setelah reboot, dan kemudian reboot sistem.

Pemicu auto-run pasca-reboot:

- **Jendela.** AWSTOEmembuat entri Penjadwal Tugas Windows dengan pemicu yang berjalan secara otomatis di `SystemStartup`
- **Linux.** AWSTOEmenambahkan pekerjaan di crontab yang berjalan secara otomatis setelah sistem reboot.

```
@reboot /download/path/awstoe run --document s3://bucket/key/doc.yaml
```

Pemicu ini dibersihkan saat aplikasi dimulai.

Percobaan ulang

Secara default, jumlah maksimum percobaan ulang diatur ke `Systems ManagerCommandRetryLimit`. Jika jumlah reboot melebihi batas coba lagi, otomatisasi gagal. Anda dapat mengubah batas dengan mengedit file konfigurasi agen Systems Manager (`Mds.CommandRetryLimit`). Lihat [Konfigurasi Runtime](#) di agen Systems Manager open source.

Untuk menggunakan modul tindakan Reboot, untuk langkah-langkah yang berisi reboot exitcode (misalnya, 3010), Anda harus menjalankan biner aplikasi sebagai `sudo user`.

Input

Primitif	Deskripsi	Tipe	Diperlukan	Default
<code>delaySeconds</code>	Menunda jumlah waktu tertentu sebelum memulai reboot.	Bulat	Tidak	0

Contoh masukan: langkah reboot

```
name: RebootStep
action: Reboot
onFailure: Abort
maxAttempts: 2
inputs:
  delaySeconds: 60
```

Keluaran

Tidak ada.

Ketika modul Reboot selesai, Image Builder melanjutkan ke langkah berikutnya dalam build.

SetRegistry

Modul SetRegistry tindakan menerima daftar input dan memungkinkan Anda untuk mengatur nilai untuk kunci registri yang ditentukan. Jika kunci registri tidak ada, itu dibuat di jalur yang ditentukan. Fitur ini hanya berlaku untuk Windows.

Input

Primitif	Deskripsi	Tipe	Diperlukan
path	Jalur kunci registri.	Tali	Ya
name	Nama kunci registri.	Tali	Ya
value	Nilai kunci registri.	String/Nomor/Array	Ya
type	Jenis nilai kunci registri.	Tali	Ya

Awalan jalur yang didukung

- HKEY_CLASSES_ROOT / HKCR:
- HKEY_USERS / HKU:
- HKEY_LOCAL_MACHINE / HKLM:
- HKEY_CURRENT_CONFIG / HKCC:
- HKEY_CURRENT_USER / HKCU:

Jenis yang didukung

- BINARY
- DWORD
- QWORD
- SZ
- EXPAND_SZ
- MULTI_SZ

Contoh masukan: atur nilai kunci registri

```
name: SetRegistryKeyValues
action: SetRegistry
maxAttempts: 3
inputs:
  - path: HKLM:\SOFTWARE\MySoftWare
    name: MyName
    value: FirstVersionSoftware
    type: SZ
  - path: HKEY_CURRENT_USER\Software\Test
    name: Version
    value: 1.1
    type: DWORD
```

Keluaran

Tidak ada.

UpdateOS

Modul tindakan updateOS menambahkan dukungan untuk menginstal pembaruan Windows dan Linux. Ini menginstal semua pembaruan yang tersedia secara default. Atau, Anda dapat mengonfigurasi daftar satu atau lebih pembaruan spesifik untuk modul tindakan yang akan diinstal. Anda juga dapat menentukan pembaruan untuk dikecualikan dari instalasi.

Jika daftar “sertakan” dan “kecualikan” disediakan, daftar pembaruan yang dihasilkan hanya dapat mencakup yang tercantum dalam daftar “sertakan” yang tidak tercantum dalam daftar “kecualikan”.

Note

UpdateOS tidak mendukung Amazon Linux 2023 (AL2023). Kami menyarankan Anda memperbarui AMI dasar Anda ke versi baru yang disertakan dengan setiap rilis. Untuk alternatif lain, lihat [Mengontrol pembaruan yang diterima dari rilis mayor dan minor](#) di Panduan Pengguna Amazon Linux 2023.

- Jendela. Pembaruan diinstal dari sumber pembaruan yang dikonfigurasi pada mesin target.
- Linux. Aplikasi memeriksa manajer paket yang didukung di platform Linux dan menggunakan salah satu yum atau manajer apt-get paket. Jika tidak ada yang didukung, kesalahan dikembalikan.

Anda harus memiliki sudo izin untuk menjalankan modul tindakan updateOS. Jika Anda tidak memiliki sudo izin, maka `error`. Input dikembalikan.

Input

Primitif	Deskripsi	Tipe	Diperlukan
<code>include</code>	<p>Untuk Windows, Anda dapat menentukan yang berikut:</p> <ul style="list-style-type: none"> Satu atau beberapa ID artikel Pangkalan Pengetahuan Microsoft (KB) untuk disertakan dalam daftar pembaruan yang mungkin diinstal. Format yang valid adalah <code>KB1234567</code> atau <code>1234567</code>. Nama pembaruan menggunakan nilai wildcard (*). Format yang valid adalah <code>Security*</code> atau <code>*Security*</code>. <p>Untuk Linux, Anda dapat menentukan satu atau lebih paket untuk dimasukkan dalam daftar</p>	Daftar String	Tidak

Primitif	Deskripsi	Tipe	Diperlukan
	pembaruan untuk instalasi.		
exclude	<p>Untuk Windows, Anda dapat menentukan yang berikut:</p> <ul style="list-style-type: none"> • Satu atau beberapa ID artikel Microsoft Knowledge Base (KB) untuk disertakan dalam daftar pembaruan yang akan dikecualikan dari penginstalan. Format yang valid adalah KB1234567 atau1234567. • Nama pembaruan menggunakan nilai wildcard (*). Format yang valid adalah: Security* atau*Security* . <p>Untuk Linux, Anda dapat menentukan satu atau lebih paket yang akan dikecualikan dari daftar pembaruan untuk instalasi.</p>	Daftar String	Tidak

Contoh masukan: tambahkan dukungan untuk menginstal pembaruan Linux

```
name: UpdateMyLinux
action: UpdateOS
onFailure: Abort
maxAttempts: 3
inputs:
  exclude:
    - ec2-hibinit-agent
```

Contoh masukan: tambahkan dukungan untuk menginstal pembaruan Windows

```
name: UpdateWindowsOperatingSystem
action: UpdateOS
onFailure: Abort
maxAttempts: 3
inputs:
  include:
    - KB1234567
    - '*Security*'
```

Keluaran

Tidak ada.

Konfigurasi input untuk perintah AWSTOE run

Untuk merampingkan input baris perintah untuk AWSTOE run perintah Anda, Anda dapat menyertakan pengaturan untuk parameter perintah dan opsi dalam file konfigurasi input format JSON dengan ekstensi `.json` file. AWSTOE dapat membaca file Anda dari salah satu lokasi berikut:

- Jalur file lokal (`./config.json`).
- <bucket-name>Bucket S3 (`s3:///config.json<bucket-path>`).

Saat Anda memasukkan run perintah, Anda dapat menentukan file konfigurasi input menggunakan `--config` parameter. Sebagai contoh:

```
awstoe run --config <file-path>/config.json
```

File konfigurasi masukan

File JSON konfigurasi input mencakup pasangan nilai kunci untuk semua pengaturan yang dapat Anda berikan secara langsung melalui parameter dan opsi run perintah. Jika Anda menentukan pengaturan di file konfigurasi input dan run perintah, sebagai parameter atau opsi, aturan prioritas berikut berlaku:

Aturan prioritas

1. Pengaturan yang diberikan langsung ke run perintah diAWS CLI, melalui parameter atau opsi, mengesampingkan nilai apa pun yang ditentukan dalam file konfigurasi input untuk pengaturan yang sama.
2. Pengaturan dalam file konfigurasi input mengesampingkan nilai default komponen.
3. Jika tidak ada pengaturan lain yang diteruskan ke dalam dokumen komponen, itu dapat menerapkan nilai default, jika ada.

Ada dua pengecualian untuk aturan ini - dokumen dan parameter. Pengaturan ini bekerja secara berbeda dalam konfigurasi input dan sebagai parameter perintah. Jika Anda menggunakan file konfigurasi input, Anda tidak boleh menentukan parameter ini langsung ke run perintah. Melakukannya akan menghasilkan kesalahan.

Pengaturan komponen

File konfigurasi input berisi pengaturan berikut. Untuk merampingkan file Anda, Anda dapat meninggalkan pengaturan opsional apa pun yang tidak diperlukan. Semua pengaturan adalah opsional kecuali dinyatakan lain.

- `cwIgnoreFailures(Boolean)` - Abaikan kegagalan logging dari CloudWatch Log.
- `cwLogGroup(String)` — `LogGroup` Nama untuk CloudWatch Log.
- `cwLogRegion(String)` - AWS Wilayah yang berlaku untuk CloudWatch Log.
- `cwLogStream(String)` - `LogStream` Nama untuk CloudWatch Log, yang mengarahkan ke AWSTOE mana untuk streaming `console.log` file.
- `Documents3 BucketOwner (String)` — ID akun pemilik bucket untuk dokumen berbasis S3 URI.
- `documents (array of objects, required)` — Sebuah array objek JSON yang mewakili dokumen komponen YAMG yang AWSTOE run menjalankan perintah. Setidaknya satu dokumen komponen harus ditentukan.

Setiap objek terdiri dari bidang-bidang berikut:

- `path` (String, required) - Lokasi file dari dokumen komponen YAMAL. Ini harus salah satu dari yang berikut:
 - Jalur file lokal (`./component-doc-example.yaml`).
 - URI S3 (`s3://bucket/key`).
 - *Versi build komponen Image Builder ARN (`arn:aws:imagebuilder:us-west-2:123456789012:component/ /2021.12.02/1`). `my-example-component`*
- `parameter` (array objek) — Sebuah array objek pasangan kunci-nilai, masing-masing mewakili parameter spesifik komponen yang diteruskan run perintah ketika menjalankan dokumen komponen. Parameter adalah opsional untuk komponen. Dokumen komponen mungkin atau mungkin tidak memiliki parameter yang ditentukan.

Setiap objek terdiri dari bidang-bidang berikut:

- `name` (String, required) — Nama parameter komponen.
- `value` (String, required) — Nilai untuk diteruskan ke dokumen komponen untuk parameter bernama.

Untuk mempelajari lebih lanjut tentang parameter komponen, lihat bagian [Parameter di Mendefinisikan dan mereferensikan variabel dalam AWSTOE](#) halaman.

- `ExecutionId` (String) — Ini adalah ID unik yang berlaku untuk eksekusi perintah saat ini. run ID ini disertakan dalam nama file keluaran dan log, untuk mengidentifikasi file-file tersebut secara unik, dan menautkannya ke eksekusi perintah saat ini. Jika pengaturan ini ditinggalkan, buat AWSTOE GUID.
- `LogDirectory` (String) - Direktori tujuan tempat AWSTOE menyimpan semua file log dari eksekusi perintah ini. Secara default, direktori ini terletak di dalam direktori induk berikut: `TOE_<DATETIME>_<EXECUTIONID>`. Jika Anda tidak menentukan direktori log, AWSTOE gunakan direktori kerja saat ini (`.`).
- `LogS3 BucketName` (String) - Jika log komponen disimpan di Amazon S3 (disarankan) AWSTOE, unggah log aplikasi komponen ke bucket S3 yang disebutkan dalam parameter ini.
- `LogS3 BucketOwner` (String) - Jika log komponen disimpan di Amazon S3 (disarankan), ini adalah ID akun pemilik untuk bucket AWSTOE tempat menulis file log.
- `LogS3 KeyPrefix` (String) - Jika log komponen disimpan di Amazon S3 (disarankan), ini adalah awalan kunci objek S3 untuk lokasi log di bucket.
- `parameter` (array objek) — Sebuah array objek pasangan kunci-nilai yang mewakili parameter yang berlaku secara global untuk semua komponen yang termasuk dalam eksekusi run perintah saat ini.

- `name` (String, required) — Nama parameter global.
- `value` (String, required) — Nilai untuk diteruskan ke semua dokumen komponen untuk parameter bernama.
- `phase` (String) - Daftar dipisahkan koma yang menentukan fase mana yang akan dijalankan dari dokumen komponen YAMG. Jika dokumen komponen menyertakan fase tambahan, itu tidak akan berjalan.
- `StateDirectory` (String) - Jalur file tempat file pelacakan negara disimpan.
- `trace` (Boolean) - Mengaktifkan logging verbose ke konsol.

Contoh

Contoh berikut menunjukkan file konfigurasi input yang menjalankan `build` dan `test` fase untuk dua dokumen komponen: `sampledoc.yaml`, dan `conversation-intro.yaml`. Setiap dokumen komponen memiliki parameter yang hanya berlaku untuk dirinya sendiri, dan keduanya menggunakan satu parameter bersama. `projectParameter` berlaku untuk kedua dokumen komponen.

```
{
  "documents": [
    {
      "path": "<file path>/awstoe/sampledoc.yaml",
      "parameters": [
        {
          "name": "dayofweek",
          "value": "Monday"
        }
      ]
    },
    {
      "path": "<file path>/awstoe/conversation-intro.yaml",
      "parameters": [
        {
          "name": "greeting",
          "value": "Hello, HAL."
        }
      ]
    }
  ],
  "phases": "build,test",
```

```
"parameters": [
  {
    "name": "project",
    "value": "examples"
  }
],
"cwLogGroup": "<log_group_name>",
"cwLogStream": "<log_stream_name>",
"documentS3BucketOwner": "<owner_aws_account_number>",
"executionId": "<id_number>",
"logDirectory": "<local_directory_path>",
"logS3BucketName": "<bucket_name_for_log_files>",
"logS3KeyPrefix": "<key_prefix_for_log_files>",
"logS3BucketOwner": "<owner_aws_account_number>"
}
```

Komponen terkelola paket distributor untuk Windows

AWS Systems Manager Distributor membantu Anda mengemas dan mempublikasikan perangkat lunak ke node yang AWS Systems Manager dikelola. Anda dapat mengemas dan mempublikasikan perangkat lunak Anda sendiri atau menggunakan Distributor untuk menemukan dan menerbitkan paket perangkat lunak agen yang AWS disediakan. Untuk informasi selengkapnya tentang Distributor Systems Manager, lihat [AWS Systems Manager Distributor](#) di Panduan AWS Systems Manager Pengguna.

Komponen terkelola untuk Distributor

Komponen terkelola Image Builder berikut menggunakan AWS Systems Manager Distributor untuk menginstal paket aplikasi pada instance Windows.

- Komponen `distributor-package-windows` terkelola menggunakan AWS Systems Manager Distributor untuk menginstal paket aplikasi yang Anda tentukan pada instance build image Windows Anda. Untuk mengonfigurasi parameter saat Anda menyertakan komponen ini dalam resep Anda, lihat [Konfigurasi distributor-package-windows sebagai komponen mandiri](#).
- `aws-vss-components-windows` Komponen menggunakan AWS Systems Manager Distributor untuk menginstal `AwsVssComponents` paket pada instance build image Windows Anda. Untuk mengonfigurasi parameter saat Anda menyertakan komponen ini dalam resep Anda, lihat [Konfigurasi aws-vss-components-windows sebagai komponen mandiri](#).

Untuk informasi selengkapnya tentang cara menggunakan komponen terkelola dalam resep Image Builder, lihat [Buat versi baru dari resep gambar](#) resep gambar atau [Buat versi baru resep wadah](#) resep kontainer. Untuk informasi selengkapnya tentang `AwsVssComponents` paket, lihat [Membuat snapshot yang konsisten dengan aplikasi VSS di Panduan Pengguna Amazon EC2 untuk Instans Windows](#).

Prasyarat

Sebelum Anda menggunakan komponen Image Builder yang mengandalkan Systems Manager Distributor untuk menginstal paket aplikasi, Anda harus memastikan bahwa prasyarat berikut terpenuhi.

- Komponen Image Builder yang menggunakan Systems Manager Distributor untuk menginstal paket aplikasi pada instans Anda memerlukan izin untuk memanggil Systems Manager API. Sebelum menggunakan komponen dalam resep Image Builder, Anda harus membuat kebijakan dan peran IAM yang memberikan izin. Untuk mengonfigurasi izin, lihat [Konfigurasi izin Distributor Systems Manager](#).

Note

Image Builder saat ini tidak mendukung paket Distributor Systems Manager yang me-reboot instance. Misalnya, paket, `AWSNVMeAWSPVDrivers`, dan `AwsEnaNetworkDriver` Distributor me-reboot instance, dan karenanya tidak diperbolehkan.

Konfigurasi izin Distributor Systems Manager

`distributor-package-windows` Komponen dan komponen lain yang menggunakannya, seperti `aws-vss-components-windows`, memerlukan izin tambahan pada instance build untuk dijalankan. Instans build harus dapat memanggil Systems Manager API untuk memulai instalasi Distributor dan polling untuk hasilnya.

Ikuti prosedur berikut AWS Management Console untuk membuat kebijakan dan peran IAM kustom yang memberikan izin bagi komponen Image Builder untuk menginstal paket Distributor Systems Manager dari instance build.

Langkah 1: Buat kebijakan

Buat kebijakan IAM untuk izin Distributor.

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Di panel navigasi, pilih Kebijakan dan kemudian pilih Buat kebijakan.
3. Pada halaman Buat kebijakan, pilih tab JSON, lalu ganti konten default dengan kebijakan JSON berikut, ganti partisi, Wilayah, dan ID akun seperlunya, atau menggunakan wildcard.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDistributorSendCommand",
      "Effect": "Allow",
      "Action": [
        "ssm:SendCommand"
      ],
      "Resource": [
        "arn:${AWS::Partition}:ssm:${AWS::Region}::document/AWS-ConfigureAWSPackage",
        "arn:${AWS::Partition}:ec2:${AWS::Region}:${AWS::AccountId}:instance/*"
      ]
    },
    {
      "Sid": "AllowGetCommandInvocation",
      "Effect": "Allow",
      "Action": [
        "ssm:GetCommandInvocation"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

4. Pilih Tinjau kebijakan.
5. Untuk Nama, masukkan nama untuk mengidentifikasi kebijakan, seperti *InvokeDistributor* atau nama lain yang Anda inginkan.
6. (Opsional) Untuk Deskripsi, masukkan deskripsi tujuan peran.
7. Pilih Buat kebijakan.

Langkah 2: Buat peran

Buat peran IAM untuk izin Distributor.

1. Dari panel navigasi konsol IAM, pilih Peran, lalu pilih Buat peran.
2. Di bawah Pilih jenis entitas tepercaya, pilih Layanan AWS.
3. Segera di bawah Pilih layanan yang akan menggunakan peran ini, pilih EC2, lalu pilih Berikutnya: Izin.
4. Di bawah Pilih kasus penggunaan Anda, pilih EC2, lalu pilih Berikutnya: Izin.
5. Dalam daftar kebijakan, pilih kotak centang di samping AmazonSSM ManagedInstanceCore. (Jenis SSM di kotak pencarian jika Anda perlu mempersempit daftar.)
6. Dalam daftar kebijakan ini, pilih kotak di sebelah EC2 InstanceProfileForImageBuilder. (Jenis ImageBuilder di kotak pencarian jika Anda perlu mempersempit daftar.)
7. Pilih Berikutnya: Tag.
8. (Opsional) Tambahkan satu atau beberapa pasangan nilai kunci tag untuk mengatur, melacak, atau mengontrol akses untuk peran ini, lalu pilih Berikutnya: Tinjau.
9. Untuk Nama peran, masukkan nama untuk peran, seperti *InvokeDistributor* atau nama lain yang Anda inginkan.
10. (Opsional) Untuk Deskripsi peran, ganti teks default dengan deskripsi tujuan peran ini.
11. Pilih Buat peran. Sistem mengembalikan Anda ke halaman Peran.

Langkah 3: Lampirkan kebijakan ke peran

Langkah terakhir untuk mengatur izin Distributor Anda adalah melampirkan kebijakan IAM ke peran IAM.

1. Dari halaman Peran di konsol IAM, pilih peran yang baru saja Anda buat. Peran Halaman ringkasan terbuka.
2. Pilih Lampirkan kebijakan.
3. Cari kebijakan yang Anda buat di prosedur sebelumnya dan pilih kotak centang di sebelah nama.
4. Pilih Lampirkan kebijakan.

Gunakan peran ini di sumber daya Konfigurasi Infrastruktur Image Builder untuk gambar apa pun yang menyertakan komponen yang menggunakan Distributor Systems Manager. Untuk informasi selengkapnya, lihat [Membuat konfigurasi infrastruktur](#).

Konfigurasi **distributor-package-windows** sebagai komponen mandiri

Untuk menggunakan `distributor-package-windows` komponen dalam resep, atur parameter berikut yang mengkonfigurasi paket yang akan diinstal.

Note

Sebelum Anda menggunakan `distributor-package-windows` komponen dalam resep, Anda harus memastikan bahwa semua [Prasyarat](#) terpenuhi.

- **Tindakan (Wajib)** - Tentukan apakah akan menginstal atau menghapus paket. Nilai yang valid mencakup `Install` dan `Uninstall`. Nilai default ke `Install`
- **PackageName(Wajib)** - Nama paket Distributor untuk menginstal atau menghapus instalasi. Untuk daftar nama paket yang valid, lihat [Temukan paket Distributor](#).
- **PackageVersion(Opsional)** — Versi paket Distributor yang akan diinstal. `PackageVersion` default ke versi yang direkomendasikan.
- **AdditionalArguments(Opsional)** - String JSON yang berisi parameter tambahan untuk diberikan ke skrip Anda untuk menginstal, menghapus, atau memperbarui paket. Untuk informasi selengkapnya, lihat `additionalArguments` di bagian [AWS:ConfigurePackage Inputs](#) dari halaman referensi plugin dokumen `Systems Manager Command`.

Konfigurasi **aws-vss-components-windows** sebagai komponen mandiri

Saat Anda menggunakan `aws-vss-components-windows` komponen dalam resep, Anda dapat mengatur `PackageVersion` parameter secara opsional untuk menggunakan versi `AwsVssComponents` paket tertentu. Ketika Anda meninggalkan parameter ini, komponen default untuk menggunakan versi paket yang direkomendasikan. `AwsVssComponents`

Note

Sebelum Anda menggunakan `aws-vss-components-windows` komponen dalam resep, Anda harus memastikan bahwa semua [Prasyarat](#) terpenuhi.

Temukan paket Distributor

Amazon dan pihak ketiga menyediakan paket publik yang dapat Anda instal dengan Systems Manager Distributor.

Untuk melihat paket yang tersedia diAWS Management Console, masuk ke [AWS Systems Managerkonsol](#) dan pilih Distributor dari panel navigasi. Halaman Distributor menunjukkan semua paket yang tersedia untuk Anda. Untuk informasi selengkapnya tentang daftar paket yang tersedia denganAWS CLI, lihat [Lihat paket \(baris perintah\)](#) di Panduan AWS Systems Manager Pengguna.

Anda juga dapat membuat paket Distributor Systems Manager pribadi Anda sendiri. Untuk informasi selengkapnya, lihat [Membuat paket](#) di Panduan AWS Systems Manager Pengguna.

Komponen pengerasan CIS

Center for Internet Security (CIS) adalah organisasi nirlaba berbasis komunitas. Pakar keamanan siber mereka bekerja sama untuk mengembangkan pedoman keamanan TI yang melindungi organisasi publik dan swasta dari ancaman cyber. Serangkaian praktik terbaik mereka yang diakui secara global, yang dikenal sebagai Tolok Ukur CIS, membantu organisasi TI di seluruh dunia untuk mengkonfigurasi sistem mereka dengan aman. Untuk artikel yang sedang tren, posting blog, podcast, webinar, dan whitepaper, lihat [CIS Insights di situs web Center for Internet Security](#).

Patokan CIS

CIS menciptakan dan memelihara seperangkat pedoman konfigurasi, yang dikenal sebagai Tolok Ukur CIS, yang menyediakan praktik terbaik konfigurasi untuk teknologi tertentu, termasuk sistem operasi, platform cloud, aplikasi, database, dan banyak lagi. Tolok Ukur CIS diakui sebagai standar industri oleh organisasi dan standar seperti PCI DSS, HIPAA, DoD Cloud Computing SRG, FISMA, DFARS, dan FEDRAMP. Untuk mempelajari lebih lanjut, lihat [Tolok Ukur CIS di situs](#) web Center for Internet Security.

Komponen pengerasan CIS

Saat Anda berlangganan CIS Hardened Image diAWS Marketplace, Anda juga mendapatkan akses ke komponen pengerasan terkait yang menjalankan skrip untuk menerapkan pedoman CIS Benchmarks Level 1 untuk konfigurasi Anda. Organisasi CIS memiliki dan memelihara komponen pengerasan CIS untuk memastikan bahwa mereka mencerminkan pedoman terbaru.

Note

Komponen pengerasan CIS tidak mengikuti aturan pemesanan komponen standar dalam resep Image Builder. Komponen pengerasan CIS selalu berjalan terakhir untuk memastikan bahwa tes benchmark berjalan terhadap gambar keluaran Anda.

Amazon mengelola komponen pengerasan STIG untuk EC2 Image Builder

Security Technical Implementation Guides (STIGs) adalah standar pengerasan konfigurasi yang dibuat oleh Badan Sistem Informasi Pertahanan (DISA) untuk mengamankan sistem informasi dan perangkat lunak. Untuk membuat sistem Anda sesuai dengan standar STIG, Anda harus menginstal, mengonfigurasi, dan menguji berbagai pengaturan keamanan.

Image Builder menyediakan komponen pengerasan STIG untuk membantu Anda membuat gambar yang sesuai dengan standar STIG dasar secara lebih efisien. Komponen STIG ini memindai kesalahan konfigurasi dan menjalankan skrip remediasi. Tidak ada biaya tambahan untuk menggunakan komponen yang sesuai dengan STIG.

Important

Dengan beberapa pengecualian, komponen pengerasan STIG tidak menginstal paket pihak ketiga. Jika paket pihak ketiga sudah diinstal pada instance, dan jika ada STIG terkait yang didukung Image Builder untuk paket itu, komponen pengerasan menerapkannya.

Halaman ini mencantumkan semua STIG yang didukung Image Builder yang diterapkan ke instans EC2 yang diluncurkan Image Builder saat Anda membuat dan menguji gambar baru. Jika Anda ingin menerapkan pengaturan STIG tambahan ke gambar Anda, Anda dapat membuat komponen khusus untuk mengonfigurasinya. Untuk informasi selengkapnya tentang komponen kustom dan cara membuatnya, lihat [Mengelola komponen dengan Image Builder](#).

Saat Anda membuat gambar, komponen pengerasan STIG mencatat apakah STIG yang didukung diterapkan atau dilewati. Kami menyarankan Anda meninjau log Image Builder untuk gambar Anda yang menggunakan komponen pengerasan STIG. Untuk informasi selengkapnya tentang cara mengakses dan meninjau log Image Builder, lihat [Memecahkan masalah pembangunan pipa](#).

Tingkat kepatuhan

- Tinggi (Kategori I)

Risiko paling parah. Termasuk kerentanan apa pun yang dapat mengakibatkan hilangnya kerahasiaan, ketersediaan, atau integritas.

- Sedang (Kategori II)

Termasuk kerentanan apa pun yang dapat mengakibatkan hilangnya kerahasiaan, ketersediaan, atau integritas, tetapi risikonya dapat dikurangi.

- Rendah (Kategori III)

Setiap kerentanan yang menurunkan langkah untuk melindungi hilangnya kerahasiaan, ketersediaan, atau integritas.

Topik

- [Komponen pengerasan STIG Windows](#)
- [Log riwayat versi STIG untuk Windows](#)
- [Komponen pengerasan STIG Linux](#)
- [Log riwayat versi STIG untuk Linux](#)
- [Komponen validator kepatuhan SCAP](#)

Komponen pengerasan STIG Windows

AWSTOEKomponen pengerasan STIG Windows dirancang untuk server mandiri dan menerapkan Kebijakan Grup Lokal. Komponen pengerasan yang sesuai dengan STIG diinstal InstallRoot dari Departemen Pertahanan (DoD) pada infrastruktur Windows untuk mengunduh, menginstal, dan memperbarui sertifikat DoD. Mereka juga menghapus sertifikat yang tidak perlu untuk mempertahankan kepatuhan STIG. Saat ini, garis dasar STIG didukung untuk versi Windows Server berikut: 2012 R2, 2016, 2019, dan 2022.

Bagian ini mencantumkan pengaturan saat ini untuk masing-masing komponen pengerasan STIG Windows, diikuti oleh log riwayat versi.

STIG-Build-Windows-versi rendah 2022.4.x

Daftar berikut berisi pengaturan STIG yang diterapkan komponen pengerasan untuk infrastruktur Anda. Jika setelan yang didukung tidak berlaku untuk infrastruktur Anda, komponen pengerasan akan melewati pengaturan itu, dan melanjutkan. Misalnya, beberapa pengaturan STIG mungkin tidak berlaku untuk server mandiri. Kebijakan khusus organisasi juga dapat memengaruhi pengaturan mana yang diterapkan komponen pengerasan, seperti persyaratan bagi administrator untuk meninjau pengaturan dokumen.

Untuk daftar lengkap STIG Windows, lihat [Perpustakaan Dokumen STIG](#). Untuk informasi tentang cara melihat daftar lengkap, lihat [Alat Melihat STIG](#).

- Windows Server 2022 STIG Versi 1 Rilis 1

V-254335, V-254336, V-254337, V-254338, V-254351, V-254357, V-254363, dan V-254481

- Windows Server 2019 STIG Versi 2 Rilis 5

V-205691, V-205819, V-205858, V-205859, V-205860, V-205870, V-205871, dan V-205923

- Windows Server 2016 STIG Versi 2 Rilis 5

V-224916, V-224917, V-224918, V-224919, V-224931, V-224942, dan V-225060

- Windows Server 2012 R2 MS STIG Versi 3 Rilis 5

V-225537, V-225536, V-225526, V-225525, V-225514, V-225511, V-225490, V-225489, V-225488, V-225487, V-225485, V-225483, V-225482, V-225481, V-225425 V-225479, V-225476, V-225473, V-225468, V-225462, V-225460, V-225459, V-225412, V-225394, V-225392, V-225376, V-225363, V-225362, V-225360, V-225359, V-225359 358, V-225357, V-225355, V-225343, V-225342, V-225336, V-225335, V-225334, V-225333, V-225332, V-225331, V-225328, V-225327, V-225324, V-225319, V-225319, V-225319, V-225318, dan V-225250

- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 2

Tidak ada pengaturan STIG yang diterapkan ke kerentanan Microsoft .NET Framework untuk Kategori III.

- Windows Firewall STIG Versi 2 Rilis 1

V-241994, V-241995, V-241996, V-241999, V-242000, V-242001, V-242006, V-242007, dan V-242008

- Internet Explorer 11 STIG Versi 2 Rilis 3

V-46477, V-46629, dan V-97527

- Microsoft Edge STIG Versi 1 Rilis 6 (hanya Windows Server 2022)

V-235727, V-235731, V-235751, V-235752, dan V-235765

STIG-build-windows-versi menengah 2022.4.x

Daftar berikut berisi pengaturan STIG yang diterapkan komponen pengerasan untuk infrastruktur Anda. Jika setelan yang didukung tidak berlaku untuk infrastruktur Anda, komponen pengerasan akan melewati pengaturan itu, dan melanjutkan. Misalnya, beberapa pengaturan STIG mungkin tidak berlaku untuk server mandiri. Kebijakan khusus organisasi juga dapat memengaruhi pengaturan mana yang diterapkan komponen pengerasan, seperti persyaratan bagi administrator untuk meninjau pengaturan dokumen.

Untuk daftar lengkap STIG Windows, lihat [Perpustakaan Dokumen STIG](#). Untuk informasi tentang cara melihat daftar lengkap, lihat [Alat Melihat STIG](#).

Note

Komponen pengerasan STIG-Build-Windows-medium mencakup semua pengaturan STIG yang terdaftar yang AWSTOE berlaku untuk komponen pengerasan rendah STIG-Build-Windows-rendah, selain pengaturan STIG yang terdaftar secara khusus untuk kerentanan Kategori II.

- Windows Server 2022 STIG Versi 1 Rilis 1

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah), ditambah:

V-254247, V-254265, V-254269, V-254270, V-254271, V-254272, V-254273, V-254274, V-254276, V-254277, V-254278, V-254285, V-254286, V-254287, V-254288, V-254288, V-254288, V-254288, V-254288, V-254290, V-254291, V-254292, V-254300, V-254301, V-254302, V-254303, V-254304, V-254305, V-254306, V-254307, V-254308, V-254309, V-254310, V-254311, V-254312, V-254312, V-254312, V-254312, V-254312, V-254313, V-254314, V-254315, V-254316, V-254317, V-254318, V-254319, V-254320, V-254321, V-254322, V-254323, V-254324, V-254325, V-254326, V-254327, V-254328, V-254329, V-254330, V-254331, V-254332, V-254333, V-254334, V-254339, V-254341, V-254342,

V-254344, V-254345, V-254346, V-254347, V-254348, V-254349, V-254350, V-254355, V-254356, V-254358, V-254359, V-254360, V-254361, V-254362, V-254364, V-254365, V-254366, V-254366, V-254366, V-254366 367, V-254368, V-254369, V-254370, V-254371, V-254372, V-254373, V-254375, V-254376, V-254377, V-254379, V-254380, V-254382, V-254383, V-254431, V-254432, V-254433, V-254433, V-254433, V-254433, V-254433, V-254433, V-254434, V-254435, V-254436, V-254438, V-254439, V-254442, V-254443, V-254444, V-254445, V-254449, V-254450, V-254451, V-254452, V-254453, V-254454, V-254455, V-254456, V-254459, V-254460, V-254461, V-254462, V-254463, V-254464, V-254468, V-254470, V-254471, V-254472, V-254473, V-254476, V-254477, V-254478, V-254479, V-254480, V-254482, V-254483, V-254484, V-254486, V-254487, V-254488, V-254488, V-254488, V-254488 4489, V-254490, V-254493, V-254494, V-254495, V-254497, V-254499, V-254501, V-254502, V-254503, V-254504, V-254505, V-254507, V-254508, V-254509, V-254510, V-254511, dan V-254512

- Windows Server 2019 STIG Versi 2 Rilis 5

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah), ditambah:

V-205625, V-205626, V-205627, V-205629, V-205630, V-205633, V-205634, V-205635, V-205636, V-205637, V-205638, V-205639, V-205643, V-205648, V-205649, V-205650, V-205651, V-205651, V-205652, V-205655, V-205656, V-205659, V-205660, V-205662, V-205671, V-205672, V-205673, V-205675, V-205676, V-205678, V-205679, V-205680, V-205681, V-205683, V-205684, V-205685, V-205685, V-205685, V-205687, V-205688, V-205689, V-205690, V-205692, V-205693, V-205694, V-205697, V-205698, V-205708, V-205709, V-205712, V-205714, V-205716, V-205717, V-205718, V-205719, V-205720, V-205720, V-205722, V-205729, V-205730, V-205733, V-205747, V-205751, V-205752, V-205754, V-205756, V-205758, V-205759, V-205760, V-205761, V-205762, V-205764, V-205766, V-205767, V-205768, V-205768, V-205769, V-205770, V-205771, V-205772, V-205773, V-205774, V-205775, V-205776, V-205777, V-205778, V-205779, V-205780, V-205781, V-205782, V-205783, V-205795, V-205796, V-205797, V-205797, V-205797, V-205797, V-205797, V-205797 98, V-205801, V-205808, V-205809, V-205810, V-205811, V-205812, V-205813, V-205814, V-205815, V-205816, V-205817, V-205821, V-205822, V-205823, V-205824, V-205825, V-205826, V-205827, V-205828, V-205830, V-205832, V-205833, V-205834, V-205835, V-205836, V-205837, V-205838, V-205839, V-205840, V-205841, V-205861, V-205863, V-205866, V-205867, V-205868, V-205869, V-205869, V-205872, V-205873, V-205874, V-205911, V-205912, V-205915, V-205916, V-205917, V-205918, V-205920, V-205921, V-205922, V-205924, V-205925, dan V-236001

- Windows Server 2016 STIG Versi 2 Rilis 5

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah), ditambah:

V-224850, V-224852, V-224853, V-224854, V-224855, V-224856, V-224857, V-224858, V-224859, V-224866, V-224867, V-224868, V-224870, V-224871, V-224871, V-224871, V-224871, V-224871, V-224872, V-224873, V-224881, V-224882, V-224883, V-224884, V-224885, V-224886, V-224887, V-224888, V-224889, V-224890, V-224891, V-224892, V-224893, V-224894, V-224894, V-224894, V-224894, V-224894 24895, V-224896, V-224897, V-224898, V-224899, V-224901, V-224902, V-224903, V-224904, V-224905, V-224906, V-224907, V-224908, V-224909, V-224910, V-224910, V-224910, V-224910 224911, V-224912, V-224913, V-224914, V-224915, V-224920, V-224922, V-224924, V-224925, V-224926, V-224927, V-224928, V-224929, V-224930, V-224935, V-224936, V-224937, V-224938, V-224939, V-224940, V-224941, V-224943, V-224944, V-224945, V-224946, V-224946, V-224946, V-224946, V-224946 47, V-224948, V-224949, V-224951, V-224952, V-224953, V-224955, V-224956, V-224957, V-224959, V-224960, V-224962, V-224963, V-225010, V-225013, V-225014, V-225015, V-225016, V-225017, V-225018, V-225019, V-225021, V-225022, V-225023, V-225024, V-225028, V-225029, V-225030, V-225031, V-225032, V-225033, V-225034, V-225035, V-225038, V-225039, V-225040, V-225041, V-225042, V-225043, V-225047, V-225049, V-225050, V-225051, V-225052, V-225055, V-225056, V-225057, V-225061, V-225062, V-225063, V-225064, V-225065, V-225066, V-225067, V-225068, V-225069, V-225072, V-225072, V-225072 25073, V-225074, V-225076, V-225078, V-225080, V-225081, V-225082, V-225083, V-225084, V-225086, V-225087, V-225088, V-225089, V-225092, V-225093 dan V-236000

- Windows Server 2012 R2 MS STIG Versi 3 Rilis 5

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah), ditambah:

V-225574, V-225573, V-225572, V-225571, V-225570, V-225569, V-225568, V-225567, V-225566, V-225565, V-225564, V-225563, V-225562, V-225561, V-225560, V-225560 9, V-225558, V-225557, V-225555, V-225554, V-225553, V-225551, V-225550, V-225549, V-225548, V-225546, V-225545, V-225544, V-225543, V-225542, V-225541, V-225541, V-225541, V-225541, V-225541, V-225541 540, V-225539, V-225538, V-225535, V-225534, V-225533, V-225532, V-225531, V-225530, V-225529, V-225528, V-225527, V-225524, V-225523, V-225522, V-225521, V-225521, V-225521, V-225520, V-225519, V-225518, V-225517, V-225516, V-225515, V-225513, V-225510, V-225509, V-225508, V-225506, V-225504, V-225503, V-225502, V-225501, V-225500, V-225494, V-225486, V-225478, V-225477, V-225475, V-225474, V-225472, V-225471, V-225470, V-225469, V-225464, V-225463, V-225461, V-225458, V-225457, V-225456, V-225455, V-225454, V-225453, V-225452, V-225448, V-225443, V-225442, V-225441, V-225415, V-225415, V-225415,

V-225415 414, V-225413, V-225411, V-225410, V-225409, V-225408, V-225407, V-225406, V-225405, V-225404, V-225402, V-225401, V-225398, V-225397, V-225395, V-225395, V-225395, V-225395, V-225395 25393, V-225391, V-225389, V-225386, V-225385, V-225384, V-225383, V-225382, V-225381, V-225380, V-225379, V-225378, V-225377, V-225375, V-225374, V-225373, V-225372, V-225371, V-225370, V-225369, V-225368, V-225367, V-225356, V-225353, V-225352, V-235252, V-23525252, V-23525252 1, V-225350, V-225349, V-225348, V-225347, V-225346, V-225345, V-225344, V-225341, V-225340, V-225339, V-225338, V-225337, V-225329, V-225326, V-225325, V-225325, V-225325, V-225325, V-225325 317, V-225316, V-225315, V-225314, V-225305, V-225304, V-225303, V-225302, V-225301, V-225300, V-225299, V-225298, V-225297, V-225296, V-225295, V-225294, V-225294, V-225294, V-225294 293, V-225292, V-225291, V-225290, V-225289, V-225288, V-225287, V-225286, V-225285, V-225284, V-225283, V-225282, V-225281, V-225280, V-225279, V-225278, V-225277, V-225276, V-225275, V-225273, V-225272, V-225271, V-225270, V-225269, V-225268, V-225267, V-225266, V-225265, V-225264, V-225263, V-225261, V-225260, V-225259, dan V-225239

- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 2

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen penguatan untuk kerentanan Kategori III (Rendah), ditambah V-225238

- Windows Firewall STIG Versi 2 Rilis 1

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen penguatan untuk kerentanan Kategori III (Rendah), ditambah:

V-241989, V-241990, V-241991, V-241993, V-241998, dan V-242003

- Internet Explorer 11 STIG Versi 2 Rilis 3

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen penguatan untuk kerentanan Kategori III (Rendah), ditambah:

V-46473, V-46475, V-46481, V-46483, V-46501, V-46507, V-46509, V-46511, V-46513, V-46515, V-46517, V-46521, V-46523, V-46525, V-46543, V-46545, V-46547, V-46549, V-46553, V-46555, V-46573, V-46575, V-46577, V-46579, V-46581, V-46583, V-46587, V-46589, V-46591, V-46593, V-46597, V-46599, V-46601, V-46603, V-46605, V-46607, V-46609, V-46660 V-46617, V-46619, V-46621, V-46625, V-46633, V-46635, V-46637, V-46639, V-46641, V-46643, V-46645, V-46647, V-46649, V-46653, V-46663, V-46665, V-46669, V-46681, V-46681, V-46681, V-46681, V-46681 685, V-46689, V-46691, V-46693, V-46695, V-46701, V-46705, V- 46709, V-46711, V-46713, V-46715, V-46717, V-46719, V-46721, V-46723, V-46725, V-46727, V-46729, V-46731, V-46733, V-46779, V-46787, V-46789, V-46791, V-46797, V-46799, V-46801, V-46807, V-46811, V-46815,

V-46819, V-46829, V-46841, V-46847, V-46849, V-46853, V-46857, V-46859, V-46861, V-46865, V-46879, V-46883, V-46885, V-46889, V-46893, V-46895, V-46897, V-46903, V-46907, V-46921, V-46927, V-46939, V-46975, V-46981, V-46987, V-46995, V-46997, V-46999, V-47003, V-47005, V-47009, V-647009, V-647009 11, V-64713, V-64715, V-64717, V-64719, V-64721, V-64723, V-64725, V-64729, V-72757, V-72759, V-72761, V-72763, V-75169, dan V-75171

- Microsoft Edge STIG Versi 1 Rilis 6 (hanya Windows Server 2022)

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen penguatan untuk kerentanan Kategori III (Rendah), ditambah:

V-235720, V-235721, V-235723, V-235724, V-235725, V-235726, V-235728, V-235729, V-235730, V-235732, V-235733, V-235734, V-235735, V-235736, V-235738, V-235739, V-235740, V-235741, V-235742, V-235743, V-235744, V-235745, V-235746, V-235747, V-235748, V-235749, V-235750, V-235754, V-235756, V-235761, V-235763, V-235764, V-235766, V-235767, V-235768, V-235768, V-235768 69, V-235770, V-235771, V-235772, V-235773, V-235774, dan V-246736

- Defender STIG Versi 2 Rilis 4 (hanya Windows Server 2022)

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen penguatan untuk kerentanan Kategori III (Rendah), ditambah:

V-213427, V-213429, V-213430, V-213431, V-213432, V-213433, V-213434, V-213435, V-213436, V-213437, V-213439, V-213440, V-213441, V-213442, V-213442, V-213442 43, V-213444, V-213445, V-213446, V-213447, V-213448, V-213449, V-213450, V-213451, V-213455, V-213464, V-213465, dan V-213466

STIG-Build-Windows-versi tinggi 2022.4.x

Daftar berikut berisi pengaturan STIG yang diterapkan komponen penguatan untuk infrastruktur Anda. Jika setelan yang didukung tidak berlaku untuk infrastruktur Anda, komponen penguatan akan melewati pengaturan itu, dan melanjutkan. Misalnya, beberapa pengaturan STIG mungkin tidak berlaku untuk server mandiri. Kebijakan khusus organisasi juga dapat memengaruhi pengaturan mana yang diterapkan komponen penguatan, seperti persyaratan bagi administrator untuk meninjau pengaturan dokumen.

Untuk daftar lengkap STIG Windows, lihat [Perpustakaan Dokumen STIG](#). Untuk informasi tentang cara melihat daftar lengkap, lihat [Alat Melihat STIG](#).

 Note

Komponen pengerasan STIG-Build-Windows-High mencakup semua pengaturan STIG yang terdaftar yang AWSTOE berlaku untuk komponen pengerasan STIG-Build-Windows-Low dan STIG-build-Windows-medium, selain pengaturan STIG yang terdaftar khusus untuk kerentanan Kategori I.

- Windows Server 2022 STIG Versi 1 Rilis 1

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori II dan III (Sedang dan Rendah), ditambah:

V-254293, V-254352, V-254353, V-254354, V-254374, V-254378, V-254381, V-254446, V-254465, V-254466, V-254467, V-254469, V-254474, V-254475, dan V-254500

- Windows Server 2019 STIG Versi 2 Rilis 5

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori II dan III (Sedang dan Rendah), ditambah:

V-205653, V-205654, V-205711, V-205713, V-205724, V-205725, V-205757, V-205802, V-205804, V-205805, V-205806, V-205849, V-205908, V-205913, V-205914, dan V-205919

- Windows Server 2016 STIG Versi 2 Rilis 5

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori II dan III (Sedang dan Rendah), ditambah:

V-224874, V-224932, V-224933, V-224934, V-224954, V-224958, V-224961, V-225025, V-225044, V-225045, V-225046, V-225048, V-225053, V-225054, dan V-225079

- Windows Server 2012 R2 MS STIG Versi 3 Rilis 5

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori II dan III (Sedang dan Rendah), ditambah:

V-225556, V-225552, V-225547, V-225507, V-225505, V-225498, V-225497, V-225496, V-225493, V-225492, V-225491, V-225449, V-225444, V-225399, V-225396, V-23256 90, V-225366, V-225365, V-225364, V-225354, dan V-225274

- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 2

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen penguatan untuk kerentanan Kategori II dan III (Sedang dan Rendah) untuk Microsoft .NET Framework. Tidak ada pengaturan STIG tambahan yang berlaku untuk kerentanan Kategori I.

- Windows Firewall STIG Versi 2 Rilis 1

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen penguatan untuk kerentanan Kategori II dan III (Sedang dan Rendah), ditambah:

V-241992, V-241997, dan V-242002

- Internet Explorer 11 STIG Versi 2 Rilis 3

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen penguatan untuk kerentanan Kategori II dan III (Sedang dan Rendah) untuk Internet Explorer 11. Tidak ada pengaturan STIG tambahan yang berlaku untuk kerentanan Kategori I.

- Microsoft Edge STIG Versi 1 Rilis 6 (hanya Windows Server 2022)

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen penguatan untuk kerentanan Kategori II dan III (Sedang dan Rendah), ditambah:

V-235758 dan V-235759

- Defender STIG Versi 2 Rilis 4 (hanya Windows Server 2022)

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen penguatan untuk kerentanan Kategori II dan III (Sedang dan Rendah), ditambah:

V-213426, V-213452, dan V-213453

Log riwayat versi STIG untuk Windows

Bagian ini mencatat riwayat versi komponen penguatan Windows untuk pembaruan STIG triwulanan. Untuk melihat perubahan dan versi yang diterbitkan selama seperempat, pilih judul untuk memperluas informasi.

Perubahan Q3 2023 - 10/04/2023 (tidak ada perubahan):

Tidak ada perubahan untuk komponen Windows STIGS untuk rilis kuartal ketiga 2023.

Perubahan Q2 2023 - 05/03/2023 (tidak ada perubahan):

Tidak ada perubahan untuk komponen Windows STIGS untuk rilis kuartal kedua 2023.

Perubahan Q1 2023 - 27/03/2023 (tidak ada perubahan):

Tidak ada perubahan untuk komponen Windows STIGS untuk rilis kuartal pertama 2023.

2022 Perubahan Q4 - 02/01/2023:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis Q4 2022 sebagai berikut:

STIG-Build-Windows-versi rendah 2022.4.x

- Windows Server 2022 STIG Versi 1 Rilis 1
- Windows Server 2019 STIG Versi 2 Rilis 5
- Windows Server 2016 STIG Versi 2 Rilis 5
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 5
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 2
- Windows Firewall STIG Versi 2 Rilis 1
- Internet Explorer 11 STIG Versi 2 Rilis 3
- Microsoft Edge STIG Versi 1 Rilis 6 (hanya Windows Server 2022)

STIG-build-windows-versi menengah 2022.4.x

- Windows Server 2022 STIG Versi 1 Rilis 1
- Windows Server 2019 STIG Versi 2 Rilis 5
- Windows Server 2016 STIG Versi 2 Rilis 5
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 5
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 2
- Windows Firewall STIG Versi 2 Rilis 1
- Internet Explorer 11 STIG Versi 2 Rilis 3
- Microsoft Edge STIG Versi 1 Rilis 6 (hanya Windows Server 2022)
- Defender STIG Versi 2 Rilis 4 (hanya Windows Server 2022)

STIG-Build-Windows-versi tinggi 2022.4.x

- Windows Server 2022 STIG Versi 1 Rilis 1
- Windows Server 2019 STIG Versi 2 Rilis 5
- Windows Server 2016 STIG Versi 2 Rilis 5
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 5
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 2
- Windows Firewall STIG Versi 2 Rilis 1
- Internet Explorer 11 STIG Versi 2 Rilis 3
- Microsoft Edge STIG Versi 1 Rilis 6 (hanya Windows Server 2022)
- Defender STIG Versi 2 Rilis 4 (hanya Windows Server 2022)

2022 Perubahan Q3 - 30/09/2022 (tidak ada perubahan):

Tidak ada perubahan untuk komponen Windows STIGS untuk rilis kuartal ketiga 2022.

2022 Perubahan Q2 - 08/02/2022:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis Q2 2022.

STIG-Build-Windows-Low versi 1.5.x

- Windows Server 2019 STIG Versi 2 Rilis 4
- Windows Server 2016 STIG Versi 2 Rilis 4
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 3
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 1
- Windows Firewall STIG Versi 2 Rilis 1
- Internet Explorer 11 STIG Versi 1 Rilis 19

STIG-Build-Windows-Medium versi 1.5.x

- Windows Server 2019 STIG Versi 2 Rilis 4
- Windows Server 2016 STIG Versi 2 Rilis 4
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 3

- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 1
- Windows Firewall STIG Versi 2 Rilis 1
- Internet Explorer 11 STIG Versi 1 Rilis 19

STIG-Build-Windows-versi tinggi 1.5.x

- Windows Server 2019 STIG Versi 2 Rilis 4
- Windows Server 2016 STIG Versi 2 Rilis 4
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 3
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 1
- Windows Firewall STIG Versi 2 Rilis 1
- Internet Explorer 11 STIG Versi 1 Rilis 19

2022 Perubahan Q1 - 08/02/2022 (tidak ada perubahan):

Tidak ada perubahan untuk komponen Windows STIGS untuk rilis kuartal pertama 2022.

2021 Perubahan Q4 - 12/20/2021:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal keempat 2021.

STIG-Build-Windows-Low versi 1.5.x

- Windows Server 2019 STIG Versi 2 Rilis 3
- Windows Server 2016 STIG Versi 2 Rilis 3
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 3
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 1
- Windows Firewall STIG Versi 2 Rilis 1
- Internet Explorer 11 STIG Versi 1 Rilis 19

STIG-Build-Windows-Medium versi 1.5.x

- Windows Server 2019 STIG Versi 2 Rilis 3
- Windows Server 2016 STIG Versi 2 Rilis 3

- Windows Server 2012 R2 MS STIG Versi 3 Rilis 3
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 1
- Windows Firewall STIG Versi 2 Rilis 1
- Internet Explorer 11 STIG Versi 1 Rilis 19

STIG-Build-Windows-versi tinggi 1.5.x

- Windows Server 2019 STIG Versi 2 Rilis 3
- Windows Server 2016 STIG Versi 2 Rilis 3
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 3
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 1
- Windows Firewall STIG Versi 2 Rilis 1
- Internet Explorer 11 STIG Versi 1 Rilis 19

2021 Perubahan Q3 - 09/30/2021:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal ketiga 2021.

STIG-Build-Windows-Low versi 1.4.x

- Windows Server 2019 STIG Versi 2 Rilis 2
- Windows Server 2016 STIG Versi 2 Rilis 2
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 2
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 1
- Windows Firewall STIG Versi 1 Rilis 7
- Internet Explorer 11 STIG Versi 1 Rilis 19

STIG-Build-Windows-Medium versi 1.4.x

- Windows Server 2019 STIG Versi 2 Rilis 2
- Windows Server 2016 STIG Versi 2 Rilis 2
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 2
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 1

- Windows Firewall STIG Versi 1 Rilis 7
- Internet Explorer 11 STIG Versi 1 Rilis 19

STIG-Build-Windows-versi tinggi 1.4.x

- Windows Server 2019 STIG Versi 2 Rilis 2
- Windows Server 2016 STIG Versi 2 Rilis 2
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 2
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 1
- Windows Firewall STIG Versi 1 Rilis 7
- Internet Explorer 11 STIG Versi 1 Rilis 19

Komponen pengerasan STIG Linux

Bagian ini berisi informasi tentang komponen pengerasan STIG Linux, diikuti oleh log riwayat versi. Jika distribusi Linux tidak memiliki pengaturan STIG sendiri, komponen pengerasan menerapkan pengaturan RHEL. Komponen pengerasan menerapkan pengaturan STIG yang didukung ke infrastruktur berdasarkan distribusi Linux, sebagai berikut:

Red Hat Enterprise Linux (RHEL) 7 pengaturan STIG

- RHEL 7
- CentOS 7
- Amazon Linux 2 (AL2)

Pengaturan RHEL 8 STIG

- RHEL 8
- CentOS 8
- Amazon Linux 2023 (AL 2023)

STIG-Build-Linux-versi rendah 2023.3.x

Daftar berikut berisi pengaturan STIG yang diterapkan komponen pengerasan untuk infrastruktur Anda. Jika setelan yang didukung tidak berlaku untuk infrastruktur Anda, komponen pengerasan

akan melewatkan pengaturan itu, dan melanjutkan. Misalnya, beberapa pengaturan STIG mungkin tidak berlaku untuk server mandiri. Kebijakan khusus organisasi juga dapat memengaruhi pengaturan mana yang diterapkan komponen pengerasan, seperti persyaratan bagi administrator untuk meninjau pengaturan dokumen.

Untuk daftar lengkapnya, lihat [Pustaka Dokumen STIGs](#). Untuk informasi tentang cara melihat daftar lengkap, lihat [Alat Melihat STIG](#).

RHEL 7 STIG Versi 3 Rilis 12

- RHEL 7/CentOS 7

V-204452, V-204576, dan V-204605

- AL2

V-204452, V-204576, dan V-204605

RHEL 8 STIG Versi 1 Rilis 11

- RHEL 8/CentOS 8/AL 2023

V-230241, V-230253, V-230269, V-230270, V-230281, V-230285, V-230346, V-230381, V-230395, V-230468, V-230469, V-230485, V-230486, V-230491, V-230494, V-230469 95, V-230496, V-230497, V-230498, V-230499, dan V-244527

Ubuntu 18.04 STIG Versi 2 Rilis 11

V-219163, V-219164, V-219165, V-219172, V-219173, V-219174, V-219175, V-219178, V-219180, V-219210, V-219301, V-219327, V-219332, dan V-219333

Ubuntu 20.04 STIG Versi 1 Rilis 9

V-238202, V-238221, V-238222, V-238223, V-238224, V-238226, V-238234, V-238235, V-238237, V-238308, V-238323, V-238357, V-238362, dan V-238373

STIG-build-linux-versi menengah 2023.3.x

Daftar berikut berisi pengaturan STIG yang diterapkan komponen pengerasan untuk infrastruktur Anda. Jika setelan yang didukung tidak berlaku untuk infrastruktur Anda, komponen pengerasan akan melewatkan pengaturan itu, dan melanjutkan. Misalnya, beberapa pengaturan STIG mungkin

tidak berlaku untuk server mandiri. Kebijakan khusus organisasi juga dapat memengaruhi pengaturan mana yang diterapkan komponen pengerasan, seperti persyaratan bagi administrator untuk meninjau pengaturan dokumen.

Untuk daftar lengkapnya, lihat [Pustaka Dokumen STIGs](#). Untuk informasi tentang cara melihat daftar lengkap, lihat [Alat Melihat STIG](#).

Note

Komponen pengerasan STIG-Build-Linux-medium mencakup semua pengaturan STIG yang terdaftar yang AWSTOE berlaku untuk komponen pengerasan rendah STIG-Build-Linux, selain pengaturan STIG yang terdaftar khusus untuk kerentanan Kategori II.

RHEL 7 STIG Versi 3 Rilis 12

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah) untuk distribusi Linux ini, ditambah:

- RHEL 7/CentOS 7

V-204405, V-204406, V-204407, V-204408, V-204409, V-204410, V-204411, V-204412, V-204413, V-204414, V-204415, V-204416, V-204417, V-204422, V-204423, V-204426, V-204427, V-204428, V-204431, V-204435, V-204437, V-204449, V-204450, V-204451, V-204457, V-204466, V-204503, V-204516, V-204517, V-204521, V-204524, V-204531, V-204536, V-204537, V-204538, V-204539, V-204540, V-204541, V-204542, V-204543, V-204544, V-204545, V-204546, V-204547, V-204548, V-204549, V-204550, V-204551, V-204552, V-204553, V-204554, V-204555, V-204556, V-204557, V-204558, V-204559, V-204560, V-204562, V-204563, V-204564, V-204565, V-204566, V-204567, V-204568, V-204572, V-204579, V-204584, V-204585, V-204586, V-204589, V-204590, V-204591, V-204592, V-204593, V-204598, V-204599, V-204600, V-204601, V-204602, V-204609, V-204610, V-204611, V-204612, V-204613, V-204614, V-204615, V-204617, V-204619, V-204622, V-204624, V-204625, V-204630, V-204631, V-204633, V-233307, V-237634, V-237635, V-251703, V-255925, V-255927, V-255928, dan V-256970

- AL2:

V-204405, V-204406, V-204407, V-204408, V-204409, V-204410, V-204411, V-204412, V-204413, V-204414, V-204415, V-204416, V-204417, V-204422, V-204423, V-204426, V-204427, V-204428, V-204431, V-204435, V-204437, V-204449, V-204450, V-204451, V-204457, V-204466, V-204503, V-204516, V-204517, V-204521, V-204524, V-204531, V-204536, V-204537, V-204538, V-204539,

V-204540, V-204541, V-204542, V-204543, V-204544, V-204545, V-204546, V-204547, V-204548, V-204549, V-204550, V-204551, V-204552, V-204553, V-204554, V-204555, V-204556, V-204556, V-204557, V-204558, V-204559, V-204560, V-204562, V-204563, V-204564, V-204565, V-204566, V-204567, V-204568, V-204572, V-204578, V-204579, V-204584, V-204585, V-204586, V-204589, V-204590, V-204591, V-204592, V-204593, V-204595, V-204598, V-204599, V-204600, V-204601, V-204602, V-204609, V-204610, V-204611, V-204612, V-204613, V-204615, V-204616, V-204617, V-204619, V-204622, V-204624, V-204625, V-204630, V-204631, V-204633, V-233307, V-237634, V-237635, V-251703, V-255925, V-255927, V-255928, dan V-256970

RHEL 8 STIG Versi 1 Rilis 11

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah) untuk distribusi Linux ini, ditambah:

- RHEL 8/CentOS 8/AL 2023

V-230228, V-230231, V-230233, V-230236, V-230237, V-230239, V-230240, V-230244, V-230255, V-230266, V-230267, V-230268, V-230273, V-230275, V-230277, V-230278, V-230280, V-230282, V-230288, V-230290, V-230291, V-230296, V-230298, V-230310, V-230311, V-230312, V-230313, V-230314, V-230315, V-230324, V-230330, V-230330, V-230330 32, V-230333, V-230334, V-230335, V-230336, V-230337, V-230338, V-230339, V-230340, V-230341, V-230342, V-230343, V-230344, V-230345, V-230348, V-230349, V-230349, V-230349, V-230349 30356, V-230357, V-230358, V-230359, V-230360, V-230361, V-230362, V-230363, V-230365, V-230368, V-230369, V-230370, V-230375, V-230377, V-230378, V-230382, V-230383, V-230386, V-230387, V-230390, V-230392, V-230402, V-230403, V-230404, V-230405, V-230405, V-230405 6, V-230407, V-230408, V-230409, V-230410, V-230411, V-230412, V-230413, V-230418, V-230419, V-230421, V-230422, V-230423, V-230424, V-230425, V-230426, V-230426, V-230426 30427, V-230428, V-230429, V-230430, V-230431, V-230432, V-230433, V-230434, V-230435, V-230436, V-230437, V-230438, V-230439, V-230444, V-230446, V-230447, V-230448, V-230449, V-230455, V-230456, V-230462, V-230463, V-230464, V-230465, V-230466, V-230467, V-230478, V-230480, V-230488, V-230489, V-230502, V-230503, V-230526, V-230527, V-230532, V-230535, V-230536, V-230537, V-230538, V-230539, V-230540, V-230540, V-230540 41, V-230542, V-230543, V-230544, V-230545, V-230546, V-230547, V-230548, V-230549, V-230555, V-230556, V-230560, V-230561, V-237640, V-237642, V-237643, V-244523, V-244524, V-244525, V-244526, V-244528, V-244534, V-244537, V-244542, V-244542, V-244549, V-244550, V-244551, V-244552, V-244553, V-244554, V-250317, V-251711, V-251713, V-251713 14, V-251715, V-251716, V-251717, V-251718, dan V-256974

Ubuntu 18.04 STIG Versi 2 Rilis 11

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah) untuk distribusi Linux ini, ditambah:

V-219149, V-219155, V-219156, V-219160, V-219166, V-219176, V-219181, V-219184, V-219186, V-219187, V-219188, V-219189, V-219190, V-219191, V-219192, V-219193, V-219194, V-219195, V-219196, V-219197, V-219198, V-219199, V-219200, V-219201, V-219202, V-219203, V-219204, V-219205, V-219206, V-219207, V-219208, V-219209, V-219213, V-219214, V-219215, V-219216, V-219217, V-219218, V-219219, V-219220, V-219221, V-219222, V-219223, V-219224, V-219227, V-219228, V-219229, V-219230, V-219231, V-219232, V-219233, V-219234, V-219235, V-219236, V-219238, V-219239, V-219240, V-219241, V-219242, V-219243, V-219244, V-219250, V-219254, V-219257, V-219263, V-219264, V-219265, V-219266, V-219267, V-219268, V-219269, V-219270, V-219271, V-219272, V-219273, V-219274, V-219275, V-219276, V-219277, V-219279, V-219281, V-219287, V-219291, V-219297, V-219298, V-219299, V-219300, V-219303, V-219303, V-219303, V-219303 6, V-219309, V-219310, V-219311, V-219312, V-219315, V-219326, V-219328, V-219330, V-219331, V-219334, V-219335, V-219336, V-219337, V-219338, V-219339, V-219339, V-219339, V-219339, V-219339, V-219339 19342, V-233779, V-233780, V-237768, V-237769, V-237770, dan V-255906

Ubuntu 20.04 STIG Versi 1 Rilis 9

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah) untuk distribusi Linux ini, ditambah:

V-238200, V-238205, V-238207, V-238209, V-238211, V-238212, V-238213, V-238216, V-238220, V-238225, V-238227, V-238228, V-238230, V-238231, V-238236, V-238238, V-238239, V-238223 40, V-238241, V-238242, V-238244, V-238245, V-238246, V-238247, V-238248, V-238249, V-238250, V-238251, V-238252, V-238253, V-238254, V-238255, V-238256, V-238257, V-238258, V-238264, V-238268, V-238271, V-238277, V-238278, V-238279, V-238280, V-238281, V-238282, V-238283, V-238284, V-238285, V-238286, V-238287, V-238287, V-238287, V-238287 8, V-238289, V-238290, V-238291, V-238292, V-238293, V-238294, V-238295, V-238297, V-238299, V-238300, V-238301, V-238302, V-238303, V-238304, V-238309, V-238310, V-238315, V-238316, V-238317, V-238319, V-238320, V-238324, V-238324, V-238324, V-238324 8325, V-238329, V-238330, V-238332, V-238333, V-238334, V-238335, V-238337, V-238338, V-238339, V-238340, V-238342, V-238343, V-238344, V-238345, V-238346, V-238347, V-238348, V-238349, V-238350, V-238351, V-238352, V-238353, V-238356, V-238358, V-238359, V-238369, V-238370, V-238376, V-238377, V-238378, V-251505, dan V-255912

STIG-Build-Linux-versi tinggi 2023.3.x

Daftar berikut berisi pengaturan STIG yang diterapkan komponen pengerasan untuk infrastruktur Anda. Jika setelan yang didukung tidak berlaku untuk infrastruktur Anda, komponen pengerasan akan melewati pengaturan itu, dan melanjutkan. Misalnya, beberapa pengaturan STIG mungkin tidak berlaku untuk server mandiri. Kebijakan khusus organisasi juga dapat memengaruhi pengaturan mana yang diterapkan komponen pengerasan, seperti persyaratan bagi administrator untuk meninjau pengaturan dokumen.

Untuk daftar lengkapnya, lihat [Pustaka Dokumen STIGs](#). Untuk informasi tentang cara melihat daftar lengkap, lihat [Alat Melihat STIG](#).

Note

Komponen pengerasan STIG-Build-Linux-High mencakup semua pengaturan STIG terdaftar yang berlaku untuk komponen pengerasan STIG-Build-Linux-Low dan STIG-Build-Linux-medium, selain pengaturan STIG terdaftar yang AWSTOE berlaku khusus untuk kerentanan Kategori I.

RHEL 7 STIG Versi 3 Rilis 12

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori II dan III (Sedang dan Rendah) untuk distribusi Linux ini, ditambah:

- RHEL 7/CentOS 7

V-204425, V-204442, V-204443, V-204447, V-204448, V-204455, V-204502, V-204620, dan V-204621

- AL2:

V-204425, V-204442, V-204443, V-204447, V-204448, V-204455, V-204502, V-204620, dan V-204621

RHEL 8 STIG Versi 1 Rilis 11

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori II dan III (Sedang dan Rendah) untuk distribusi Linux ini, ditambah:

- RHEL 8/CentOS 8/AL 2023

V-230264, V-230265, V-230487, V-230492, V-230529, V-230531, V-230533, dan V-230558

Ubuntu 18.04 STIG Versi 2 Rilis 11

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori II dan III (Sedang dan Rendah) untuk distribusi Linux ini, ditambah:

V-219157, V-219158, V-219177, V-219212, V-219308, V-219313, V-219314, V-219316, V-251506, dan V-251507

Ubuntu 20.04 STIG Versi 1 Rilis 9

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori II dan III (Sedang dan Rendah) untuk distribusi Linux ini, ditambah:

V-238201, V-238215, V-238218, V-238219, V-238326, V-238327, V-238380, V-251503, dan V-251504

Log riwayat versi STIG untuk Linux

Bagian ini mencatat riwayat versi komponen Linux. Untuk melihat perubahan dan versi yang diterbitkan selama seperempat, pilih judul untuk memperluas informasi.

Perubahan Q3 2023 - 10/04/2023:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal ketiga 2023 sebagai berikut:

STIG-Build-Linux-versi rendah 2023.3.x

- RHEL 7 STIG Versi 3 Rilis 12
- RHEL 8 STIG Versi 1 Rilis 11
- Ubuntu 18.04 STIG Versi 2 Rilis 11
- Ubuntu 20.04 STIG Versi 1 Rilis 9

STIG-build-linux-versi menengah 2023.3.x

- RHEL 7 STIG Versi 3 Rilis 12
- RHEL 8 STIG Versi 1 Rilis 11

- Ubuntu 18.04 STIG Versi 2 Rilis 11
- Ubuntu 20.04 STIG Versi 1 Rilis 9

STIG-Build-Linux-versi tinggi 2023.3.x

- RHEL 7 STIG Versi 3 Rilis 12
- RHEL 8 STIG Versi 1 Rilis 11
- Ubuntu 18.04 STIG Versi 2 Rilis 11
- Ubuntu 20.04 STIG Versi 1 Rilis 9

Perubahan Q2 2023 - 05/03/2023:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal kedua 2023 sebagai berikut:

STIG-Build-Linux-versi rendah 2023.2.x

- RHEL 7 STIG Versi 3 Rilis 11
- RHEL 8 STIG Versi 1 Rilis 10
- Ubuntu 18.04 STIG Versi 2 Rilis 11
- Ubuntu 20.04 STIG Versi 1 Rilis 8

STIG-build-linux-versi menengah 2023.2.x

- RHEL 7 STIG Versi 3 Rilis 11
- RHEL 8 STIG Versi 1 Rilis 10
- Ubuntu 18.04 STIG Versi 2 Rilis 11
- Ubuntu 20.04 STIG Versi 1 Rilis 8

STIG-Build-Linux-versi tinggi 2023.2.x

- RHEL 7 STIG Versi 3 Rilis 11
- RHEL 8 STIG Versi 1 Rilis 10
- Ubuntu 18.04 STIG Versi 2 Rilis 11
- Ubuntu 20.04 STIG Versi 1 Rilis 8

Perubahan Q1 2023 - 27/03/2023:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal pertama 2023 sebagai berikut:

STIG-Build-Linux-versi rendah 2023.1.x

- RHEL 7 STIG Versi 3 Rilis 10
- RHEL 8 STIG Versi 1 Rilis 9
- Ubuntu 18.04 STIG Versi 2 Rilis 10
- Ubuntu 20.04 STIG Versi 1 Rilis 7

STIG-build-linux-versi menengah 2023.1.x

- RHEL 7 STIG Versi 3 Rilis 10
- RHEL 8 STIG Versi 1 Rilis 9
- Ubuntu 18.04 STIG Versi 2 Rilis 10
- Ubuntu 20.04 STIG Versi 1 Rilis 7

STIG-Build-Linux-versi tinggi 2023.1.x

- RHEL 7 STIG Versi 3 Rilis 10
- RHEL 8 STIG Versi 1 Rilis 9
- Ubuntu 18.04 STIG Versi 2 Rilis 10
- Ubuntu 20.04 STIG Versi 1 Rilis 7

2022 Perubahan Q4 - 02/01/2023:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal keempat 2022 sebagai berikut:

STIG-Build-Linux-versi rendah 2022.4.x

- RHEL 7 STIG Versi 3 Rilis 9
- RHEL 8 STIG Versi 1 Rilis 8
- Ubuntu 18.04 STIG Versi 2 Rilis 9
- Ubuntu 20.04 STIG Versi 1 Rilis 6

STIG-build-linux-versi menengah 2022.4.x

- RHEL 7 STIG Versi 3 Rilis 9
- RHEL 8 STIG Versi 1 Rilis 8
- Ubuntu 18.04 STIG Versi 2 Rilis 9
- Ubuntu 20.04 STIG Versi 1 Rilis 6

STIG-Build-Linux-versi tinggi 2022.4.x

- RHEL 7 STIG Versi 3 Rilis 9
- RHEL 8 STIG Versi 1 Rilis 8
- Ubuntu 18.04 STIG Versi 2 Rilis 9
- Ubuntu 20.04 STIG Versi 1 Rilis 6

2022 Perubahan Q3 - 30/09/2022 (tidak ada perubahan):

Tidak ada perubahan untuk komponen Linux STIGS untuk rilis kuartal ketiga 2022.

2022 Perubahan Q2 - 08/02/2022:

Memperkenalkan dukungan Ubuntu, versi STIG yang diperbarui, dan menerapkan STIGS untuk rilis kuartal kedua 2022 sebagai berikut:

STIG-Build-Linux-versi rendah 2022.2.x

- RHEL 7 STIG Versi 3 Rilis 7
- RHEL 8 STIG Versi 1 Rilis 6
- Ubuntu 18.04 STIG Versi 2 Rilis 6 (baru)
- Ubuntu 20.04 STIG Versi 1 Rilis 4 (baru)

STIG-build-linux-versi menengah 2022.2.x

- RHEL 7 STIG Versi 3 Rilis 7
- RHEL 8 STIG Versi 1 Rilis 6
- Ubuntu 18.04 STIG Versi 2 Rilis 6 (baru)
- Ubuntu 20.04 STIG Versi 1 Rilis 4 (baru)

STIG-Build-Linux-versi tinggi 2022.2.x

- RHEL 7 STIG Versi 3 Rilis 7
- RHEL 8 STIG Versi 1 Rilis 6
- Ubuntu 18.04 STIG Versi 2 Rilis 6 (baru)
- Ubuntu 20.04 STIG Versi 1 Rilis 4 (baru)

2022 Perubahan Q1 - 26/04/2022:

Difaktorkan ulang untuk menyertakan dukungan yang lebih baik untuk wadah. Menggabungkan skrip AL2 sebelumnya dengan RHEL 7. Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal pertama 2022 sebagai berikut:

STIG-Build-Linux-versi rendah 3.6.x

- RHEL 7 STIG Versi 3 Rilis 6
- RHEL 8 STIG Versi 1 Rilis 5

STIG-Build-Linux-Medium versi 3.6.x

- RHEL 7 STIG Versi 3 Rilis 6
- RHEL 8 STIG Versi 1 Rilis 5

STIG-Build-Linux-versi tinggi 3.6.x

- RHEL 7 STIG Versi 3 Rilis 6
- RHEL 8 STIG Versi 1 Rilis 5

2021 Perubahan Q4 - 12/20/2021:

Versi STIG yang diperbarui, dan menerapkan STIGS untuk rilis kuartal keempat 2021 sebagai berikut:

STIG-Build-Linux-versi rendah 3.5.x

- RHEL 7 STIG Versi 3 Rilis 5
- RHEL 8 STIG Versi 1 Rilis 4

STIG-Build-Linux-Medium versi 3.5.x

- RHEL 7 STIG Versi 3 Rilis 5
- RHEL 8 STIG Versi 1 Rilis 4

STIG-Build-Linux-versi tinggi 3.5.x

- RHEL 7 STIG Versi 3 Rilis 5
- RHEL 8 STIG Versi 1 Rilis 4

2021 Perubahan Q3 - 09/30/2021:

Versi STIG yang diperbarui, dan menerapkan STIGS untuk rilis kuartal ketiga 2021 sebagai berikut:

STIG-Build-Linux-versi rendah 3.4.x

- RHEL 7 STIG Versi 3 Rilis 4
- RHEL 8 STIG Versi 1 Rilis 3

STIG-Build-Linux-Medium versi 3.4.x

- RHEL 7 STIG Versi 3 Rilis 4
- RHEL 8 STIG Versi 1 Rilis 3

STIG-Build-Linux-versi tinggi 3.4.x

- RHEL 7 STIG Versi 3 Rilis 4
- RHEL 8 STIG Versi 1 Rilis 3

Komponen validator kepatuhan SCAP

Security Content Automation Protocol (SCAP) adalah seperangkat standar yang dapat digunakan oleh para profesional TI untuk mengidentifikasi kerentanan keamanan aplikasi untuk kepatuhan. SCAP Compliance Checker (SCC) adalah alat pemindaian yang divalidasi SCAP-yang dirilis oleh Naval Information Warfare Center (NIWC) Atlantic. Untuk informasi selengkapnya, lihat [Security Content Automation Protocol \(SCAP\) Compliance Checker \(SCC\)](#) di situs web NIWC Atlantic.

`scap-compliance-checker-linux` Komponen AWSTOE `scap-compliance-checker-windows` dan mengunduh dan menginstal pemindai SCC pada instance pembuatan dan pengujian pipa. Saat pemindai berjalan, ia melakukan pemindaian konfigurasi yang diautentikasi menggunakan Benchmark SCAP DISA, dan memberikan laporan yang menyertakan informasi berikut. AWSTOE juga menulis informasi ke log aplikasi Anda.

- Pengaturan STIG yang diterapkan pada instance.
- Skor kepatuhan keseluruhan untuk contoh tersebut.

Kami menyarankan Anda menjalankan validasi SCAP sebagai langkah terakhir dalam proses pembuatan, untuk memastikan bahwa Anda melaporkan hasil validasi kepatuhan yang akurat.

Note

Anda dapat meninjau laporan dengan salah satu [STIG Viewing Tools](#). Alat-alat ini tersedia secara online melalui DoD Cyber Exchange.

Bagian berikut menjelaskan tolok ukur yang termasuk dalam komponen validasi SCAP.

`scap-compliance-checker-linux` versi 2021.04.0

`scap-compliance-checker-linux` Komponen berjalan pada instance build dan test pipeline Image Builder. AWSTOE mencatat laporan dan skor yang dihasilkan aplikasi SCC.

Komponen melakukan langkah-langkah alur kerja berikut:

1. Mengunduh dan menginstal aplikasi SCC.
2. Mengimpor tolok ukur kepatuhan.
3. Menjalankan validasi menggunakan aplikasi SCC.
4. Menyimpan laporan kepatuhan dan skor secara lokal di desktop instance build.
5. Mencatat skor kepatuhan dari laporan lokal ke file log AWSTOE aplikasi.

Note

AWSTOE saat ini mendukung validasi kepatuhan SCAP untuk Windows Server 2012 R2, 2016, dan 2019.

Komponen pemeriksa kepatuhan SCAP untuk Windows mencakup tolok ukur berikut:

Versi SCC: 5.4.2

2021 Q4 Tolok Ukur:

- U_MS__Framework_4-0_v2r1_stig_scap_1-2_benchmark DotNet
- u_ms_ie11_v2r1_stig_scap_1-2_patokan
- u_ms_windows_2012_dan_2012_r2_ms_v3r2_stig_scap_1-2_patokan
- u_ms_windows_defender_av_v2r2_stig_scap_1-2_patokan
- u_ms_windows_server_2016_v2r1_stig_scap_1-2_patokan
- u_ms_windows_server_2019_v2r1_stig_scap_1-2_patokan
- u_ms_windows_firewall_v2r1_stig_scap_1-2_patokan
- u_can_ubuntu_18-04_v2r4_stig_scap_1-2_patokan
- u_rhel_7_v3r5_stig_scap_1-2_patokan
- u_rhel_8_v1r3_stig_scap_1-2_patokan

scap-compliance-checker-linux versi 2021.04.0

scap-compliance-checker-linuxKomponen berjalan pada instance build dan test pipeline Image Builder. AWSTOEmencatat laporan dan skor yang dihasilkan aplikasi SCC.

Komponen melakukan langkah-langkah alur kerja berikut:

1. Mengunduh dan menginstal aplikasi SCC.
2. Mengimpor tolok ukur kepatuhan.
3. Menjalankan validasi menggunakan aplikasi SCC.
4. Menyimpan laporan kepatuhan dan skor secara lokal, di lokasi berikut pada instance build:/opt/scc/SCCResults.
5. Mencatat skor kepatuhan dari laporan lokal ke file log AWSTOE aplikasi.

Note

AWSTOEsaat ini mendukung validasi kepatuhan SCAP untuk RHEL 7/8 dan Ubuntu 18. Aplikasi SCC saat ini mendukung arsitektur x86 untuk validasi.

Komponen pemeriksa kepatuhan SCAP untuk Linux mencakup tolok ukur berikut:

Versi SCC: 5.4.2

2021 Q4 Tolok Ukur:

- u_can_ubuntu_18-04_v2r4_stig_scap_1-2_patokan
- u_rhel_7_v3r5_stig_scap_1-2_patokan
- u_rhel_8_v1r3_stig_scap_1-2_patokan
- U_MS_ _Framework_4-0_v2r1_stig_scap_1-2_benchmark DotNet
- u_ms_ie11_v2r1_stig_scap_1-2_patokan
- u_ms_windows_2012_dan_2012_r2_ms_v3r2_stig_scap_1-2_patokan
- u_ms_windows_defender_av_v2r2_stig_scap_1-2_patokan
- u_ms_windows_server_2016_v2r1_stig_scap_1-2_patokan
- u_ms_windows_server_2019_v2r1_stig_scap_1-2_patokan
- u_ms_windows_firewall_v2r1_stig_scap_1-2_patokan

riwayat versi SCAP

Tabel berikut menjelaskan perubahan penting pada lingkungan dan pengaturan SCAP yang dijelaskan dalam dokumen ini.

Perubahan	Deskripsi	Tanggal
Ditambahkan komponen SCAP	<p>Memperkenalkan komponen SCAP berikut:</p> <ul style="list-style-type: none"> • scap-compliance-checker-linux Versi yang dibuat 2021.04.0 (SCC Version: 5.4.2) • scap-compliance-checker-linux Versi yang dibuat 2021.04.0 (SCC Version: 5.4.2) 	Desember 20, 2021

AWSTOEREferensi perintah

AWSTOE adalah aplikasi manajemen komponen yang berjalan di AWS CLI.

Note

Beberapa modul AWSTOE tindakan memerlukan izin tinggi untuk berjalan di server Linux. Untuk menggunakan izin yang ditinggikan, awali sintaks perintah dengan `sudo`, atau jalankan `sudo su` perintah satu kali ketika Anda masuk sebelum menjalankan perintah yang ditautkan di bawah ini. Untuk informasi selengkapnya tentang modul AWSTOE tindakan, lihat [Modul aksi didukung oleh manajer AWSTOE komponen](#).

run

Gunakan `run` perintah untuk menjalankan skrip dokumen YAMAL untuk satu atau beberapa dokumen komponen.

validasi

Jalankan `validate` perintah untuk memvalidasi sintaks dokumen YAMAL untuk satu atau beberapa dokumen komponen.

awstoe jalankan perintah

Perintah ini menjalankan skrip dokumen komponen YAMAL dalam urutan di mana mereka disertakan dalam file konfigurasi yang ditentukan oleh `--config` parameter, atau daftar dokumen komponen yang ditentukan oleh parameter. `--documents`

Note

Anda harus menentukan dengan tepat salah satu parameter berikut, tidak pernah keduanya:

`--config`

`--dokumen`

Sintaksis

```
awstoe run [--config <file path>] [--cw-ignore-failures <?>]
```



```

[--cw-log-group <?>] [--cw-log-region us-west-2] [--cw-log-stream <?>]
[--document-s3-bucket-owner <owner>] [--documents <file path,file path,...>]
[--execution-id <?>] [--log-directory <file path>]
[--log-s3-bucket-name <name>] [--log-s3-bucket-owner <owner>]
[--log-s3-key-prefix <?>] [--parameters name1=value1,name2=value2...]
[--phases <phase name>] [--state-directory <directory path>] [--version <?>]
[--help] [--trace]

```

Parameter dan opsi

Parameter

`--config ./config-example.json`

Bentuk pendek: `-c ./config-example.json`

File konfigurasi (kondisional). Parameter ini berisi lokasi file untuk file JSON yang berisi pengaturan konfigurasi untuk komponen yang sedang dijalankan perintah ini. Jika Anda menentukan pengaturan run perintah dalam file konfigurasi, Anda tidak boleh menentukan `--documents` parameter. Untuk informasi selengkapnya tentang konfigurasi input, lihat [Konfigurasi input untuk perintah AWSTOE run](#).

Lokasi yang valid meliputi:

- Jalur file lokal (*./config-example.json*)
- URI S3 () *s3://bucket/key*

`--cw-ignore-failures`

Bentuk pendek: N/A

Abaikan kegagalan logging dari CloudWatch Log.

`--cw-log-group`

Bentuk pendek: N/A

LogGroupName untuk CloudWatch Log.

`--cw-log-region`

Bentuk pendek: N/A

AWSWilayah yang berlaku untuk CloudWatch Log.

--cw-log-stream

Bentuk pendek: N/A

LogStreamNama untuk CloudWatch Log, yang mengarahkan ke AWSTOE mana untuk streaming console.log file.

--dokumen-s3-bucket-pemilik

Bentuk pendek: N/A

ID akun pemilik bucket untuk dokumen berbasis S3 URI.

--dokumen./doc-1.yaml, ./doc-n.yaml

Bentuk pendek: -d *./doc-1.yaml, ./doc-n*

Dokumen komponen (kondisional). Parameter ini berisi daftar lokasi file yang dipisahkan koma untuk menjalankan dokumen komponen YAMAL. Jika Anda menentukan dokumen YAMAL untuk run perintah menggunakan --documents parameter, Anda tidak boleh menentukan --config parameternya.

Lokasi yang valid meliputi:

- jalur file lokal (*./component-doc-example.yaml*).
- URI S3 (*s3://bucket/key*).
- *Versi pembuatan komponen Image Builder ARN (arn:aws:imagebuilder:us-west-2:123456789012:component/ /2021.12.02/1). my-example-component*

Note

Tidak ada spasi di antara item dalam daftar, hanya koma.

--eksekusi-id

Bentuk pendek: -i

Ini adalah ID unik yang berlaku untuk eksekusi run perintah saat ini. ID ini disertakan dalam nama file keluaran dan log, untuk mengidentifikasi file-file tersebut secara unik, dan menautkannya ke eksekusi perintah saat ini. Jika pengaturan ini ditinggalkan, buat AWSTOE GUID.

--log-direktori

Bentuk pendek: -l

Direktori tujuan tempat AWSTOE menyimpan semua file log dari eksekusi perintah ini. Secara default, direktori ini terletak di dalam direktori induk berikut: `TOE_<DATETIME>_<EXECUTIONID>`.

Jika Anda tidak menentukan direktori log, AWSTOE gunakan direktori kerja saat ini (.).

--log-s3-nama-ember

Bentuk pendek: -b

Jika log komponen disimpan di Amazon S3 (disarankan), AWSTOE unggah log aplikasi komponen ke bucket S3 yang disebutkan dalam parameter ini.

--log-s3-bucket-pemilik

Bentuk pendek: N/A

Jika log komponen disimpan di Amazon S3 (disarankan), ini adalah ID akun pemilik untuk bucket tempat AWSTOE menulis file log.

--log-s3-key-awalan

Bentuk pendek: -k

Jika log komponen disimpan di Amazon S3 (disarankan), ini adalah awalan kunci objek S3 untuk lokasi log di bucket.

--parameter name1 = value1, name2 = value2...

Bentuk pendek: N/A

Parameter adalah variabel yang dapat berubah yang didefinisikan dalam dokumen komponen, dengan pengaturan yang dapat disediakan oleh aplikasi pemanggilan saat runtime.

--fase

Bentuk pendek: -p

Daftar dipisahkan koma yang menentukan fase mana yang akan dijalankan dari dokumen komponen YAMAL. Jika dokumen komponen menyertakan fase tambahan, itu tidak akan berjalan.

--state-direktori

Bentuk pendek: -s

Jalur file tempat file pelacakan negara disimpan.

--versi

Bentuk pendek: -v

Menentukan versi aplikasi komponen.

Opsi

--help

Bentuk pendek: -h

Menampilkan manual bantuan untuk menggunakan opsi aplikasi manajemen komponen.

--jejak

Bentuk pendek: -t

Mengaktifkan pencatatan verbose ke konsol.

awstoe memvalidasi perintah

Ketika Anda menjalankan perintah ini, itu memvalidasi sintaks dokumen YAMAL untuk setiap dokumen komponen yang ditentukan oleh parameter. --documents

Sintaksis

```
awstoe validate [--document-s3-bucket-owner <owner>]
  --documents <file path,file path,...> [--help] [--trace]
```

Parameter dan opsi

Parameter

--dokumen-s3-bucket-pemilik

Bentuk pendek: N/A


ID akun sumber dokumen berbasis S3 URI disediakan.

`--dokumen./doc-1.yaml, ./doc-n.yaml`

Bentuk pendek: `-d ./doc-1.yaml, ./doc-n`

Dokumen komponen (wajib). Parameter ini berisi daftar lokasi file yang dipisahkan koma untuk menjalankan dokumen komponen YAMAL. Lokasi yang valid meliputi:

- jalur file lokal (`./component-doc-example.yaml`)
- URI S3 (`s3://bucket/key`)
- *Versi pembuatan komponen Image Builder ARN (`arn:aws:imagebuilder:us-west-2:123456789012:component/ /2021.12.02/1`) `my-example-component`*

 Note

Tidak ada spasi di antara item dalam daftar, hanya koma.

Ops

`--help`

Bentuk pendek: `-h`

Menampilkan manual bantuan untuk menggunakan opsi aplikasi manajemen komponen.

`--jejak`

Bentuk pendek: `-t`

Mengaktifkan pencatatan verbose ke konsol.

Kelola sumber daya EC2 Image Builder

Sumber daya adalah blok bangunan yang membentuk pipa gambar, serta gambar yang dihasilkan pipa tersebut. Bab ini mencakup pembuatan, pemeliharaan, dan berbagi sumber daya Image Builder, termasuk komponen, resep, dan gambar, bersama dengan konfigurasi infrastruktur dan pengaturan distribusi.

Note

Untuk membantu mengelola sumber daya Image Builder, Anda dapat menetapkan metadata Anda sendiri ke setiap sumber daya dalam bentuk tag. Anda menggunakan tag untuk mengkategorikan AWS sumber daya Anda dengan cara yang berbeda; misalnya, berdasarkan tujuan, pemilik, atau lingkungan. Hal ini berguna jika Anda memiliki banyak sumber daya dengan jenis yang sama. Anda dapat lebih mudah mengidentifikasi sumber daya tertentu berdasarkan tag yang telah Anda tetapkan padanya.

Untuk informasi selengkapnya tentang menandai sumber daya Anda menggunakan perintah Image Builder diAWS CLI, lihat [Memberi tanda pada sumber daya](#) bagian panduan ini.

Konten

- [Mengelola komponen dengan Image Builder](#)
- [Kelola resep](#)
- [Mengelola gambar EC2 Image Builder](#)
- [Kelola konfigurasi infrastruktur EC2 Image Builder](#)
- [Mengelola pengaturan distribusi EC2 Image Builder](#)
- [Mengelola kebijakan siklus hidup untuk gambar EC2 Image Builder](#)
- [Mengelola alur kerja build dan test untuk image EC2 Image Builder](#)
- [Impor dan ekspor gambar mesin virtual \(VM\) dengan EC2 Image Builder](#)
- [Bagikan sumber daya EC2 Image Builder](#)
- [Menandai sumber daya EC2 Image Builder](#)
- [Hapus sumber EC2 Image Builder](#)

Mengelola komponen dengan Image Builder

Image Builder menggunakan aplikasi manajemen komponen AWS Task Orchestrator and Executor (AWSTOE) untuk mengatur alur kerja yang kompleks. Membangun dan menguji komponen yang bekerja dengan AWSTOE aplikasi didasarkan pada dokumen YAMAL yang menentukan skrip untuk menyesuaikan atau menguji gambar Anda. Untuk image AMI, Image Builder menginstal komponen dan aplikasi manajemen AWSTOE komponen pada instans build dan pengujian Amazon EC2. Untuk gambar kontainer, AWSTOE komponen dan aplikasi manajemen komponen dipasang di dalam wadah yang sedang berjalan.

Image Builder menggunakan AWSTOE untuk melakukan semua aktivitas on-instance. Tidak ada pengaturan tambahan yang diperlukan untuk berinteraksi AWSTOE saat Anda menjalankan perintah Image Builder atau menggunakan konsol Image Builder.

Note

Ketika komponen yang dikelola oleh Amazon mencapai akhir masa dukungannya, komponen tersebut tidak lagi dipertahankan. Sekitar empat minggu sebelum ini terjadi, setiap akun yang menggunakan komponen menerima pemberitahuan, dan daftar resep yang terpengaruh di akun mereka dari mereka AWS Health Dashboard. Untuk mempelajari selengkapnya tentang AWS Health, lihat [Panduan Pengguna AWS Health](#).

Tahapan alur kerja untuk membangun gambar baru

Alur kerja Image Builder untuk membuat gambar baru mencakup dua tahapan berbeda berikut.

1. Tahap build (pre-snapshot) — Selama tahap build, Anda membuat perubahan pada instance build Amazon EC2 yang menjalankan image dasar, untuk membuat baseline untuk image baru Anda. Misalnya, resep Anda dapat menyertakan komponen yang menginstal aplikasi atau memodifikasi pengaturan firewall sistem operasi.

Fase komponen berikut berjalan selama tahap build:

- build
- validasi

Setelah tahap ini selesai dengan sukses, Image Builder membuat snapshot atau image kontainer yang digunakan untuk tahap pengujian dan seterusnya.

2. Tahap pengujian (post-snapshot) — Selama tahap pengujian, ada beberapa perbedaan antara gambar yang membuat AMI dan gambar kontainer. Untuk alur kerja AMI, Image Builder meluncurkan instans EC2 dari snapshot yang dibuatnya sebagai langkah terakhir dari tahap pembuatan. Pengujian dijalankan pada instance baru untuk memvalidasi pengaturan dan memastikan bahwa instance berfungsi seperti yang diharapkan. Untuk alur kerja kontainer, pengujian dijalankan pada instance yang sama yang digunakan untuk membangun.

Fase komponen berikut berjalan untuk setiap komponen yang termasuk dalam resep selama tahap pengujian:

- pengujian

Fase komponen ini berlaku untuk tipe komponen Build dan Test. Setelah tahap ini selesai dengan sukses, Image Builder dapat membuat dan mendistribusikan gambar akhir Anda dari snapshot atau gambar kontainer.

Note

Meskipun AWSTOE memungkinkan Anda untuk menentukan banyak fase dalam dokumen komponen, Image Builder memiliki aturan ketat tentang fase apa yang dijalankannya, dan selama tahapan mana ia menjalankannya. Agar komponen dapat berjalan selama tahap pembuatan, dokumen komponen harus mendefinisikan setidaknya satu dari fase ini: `build` atau `validate`. Agar komponen berjalan selama tahap pengujian, dokumen komponen harus menentukan `test` fase, dan tidak ada fase lainnya.

Karena Image Builder menjalankan tahapan secara independen, referensi rantai dalam dokumen komponen tidak dapat melewati batas tahap. Anda tidak dapat merantai nilai dari fase yang berjalan di tahap build ke fase yang berjalan di tahap pengujian. Namun, Anda dapat menentukan parameter input ke target yang dimaksud, dan meneruskan nilai melalui baris perintah. Untuk informasi selengkapnya tentang menyetel parameter komponen dalam resep Image Builder, lihat [Kelola parameter AWSTOE komponen dengan EC2 Image Builder](#).

Untuk membantu pemecahan masalah pada AWSTOE instance build atau pengujian Anda, buat folder log yang berisi dokumen input dan file log untuk melacak apa yang terjadi setiap kali komponen berjalan. Jika Anda mengonfigurasi bucket Amazon S3 dalam konfigurasi pipeline, log juga ditulis di sana. Untuk informasi selengkapnya tentang dokumen YAMAL dan keluaran log, lihat [Gunakan dokumen komponen di AWSTOE](#).

Tip

Bila Anda memiliki banyak komponen untuk dilacak, penandaan membantu Anda mengidentifikasi komponen atau versi tertentu berdasarkan tag yang telah Anda tetapkan padanya. Untuk informasi selengkapnya tentang menandai sumber daya Anda menggunakan perintah Image Builder diAWS CLI, lihat [Memberi tanda pada sumber daya](#) bagian panduan ini.

Bagian ini mencakup cara membuat daftar, melihat, membuat, dan mengimpor komponen, menggunakan konsol Image Builder atau perintah diAWS CLI.

Daftar Isi

- [Membuat dokumen komponen YAMAL](#)
- [Kelola parameter AWSTOE komponen dengan EC2 Image Builder](#)
- [Daftar dan lihat detail komponen](#)
- [Membuat komponen menggunakan konsol Image Builder](#)
- [Buat komponen dengan AWS CLI](#)
- [Impor komponen \(AWS CLI\)](#)
- [Pembersihan sumber daya](#)

Membuat dokumen komponen YAMAL

Untuk membangun komponen, sediakan dokumen komponen aplikasi YAMAL. Ini mewakili fase dan langkah yang Anda butuhkan untuk membuat komponen.

Contoh di bagian ini membuat komponen build yang memanggil modul UpdateOS tindakan dalam aplikasi manajemen AWSTOE komponen. Modul memperbarui sistem operasi. Untuk informasi selengkapnya tentang modul UpdateOS tindakan, lihat [UpdateOS](#). Untuk informasi selengkapnya tentang fase, langkah, dan sintaks untuk dokumen komponen AWSTOE aplikasi YAMAL, lihat [Menggunakan dokumen](#) di AWSTOE

Note

Image Builder menentukan jenis komponen dalam alur kerja pipeline. Alur kerja ini sesuai dengan tahap Build dan tahap Test dalam proses build. Image Builder menentukan jenis komponen sebagai berikut:

- Build — Ini adalah tipe komponen default. Apa pun yang tidak diklasifikasikan sebagai komponen uji, adalah komponen build. Jenis komponen ini berjalan selama tahap Build. Jika komponen build ini memiliki test fase yang ditentukan, fase tersebut berjalan selama tahap Test.
- Test — Untuk memenuhi syarat sebagai komponen pengujian, dokumen komponen harus menyertakan hanya satu fase, bernama `test`. Untuk pengujian yang terkait dengan konfigurasi komponen build, sebaiknya Anda tidak menggunakan komponen pengujian mandiri. Sebaliknya, gunakan test fase dalam komponen build terkait.

Untuk informasi selengkapnya tentang cara Image Builder menggunakan tahapan dan fase untuk mengelola alur kerja komponen dalam proses pembuatannya, lihat [Mengelola komponen dengan Image Builder](#).

Untuk membuat dokumen komponen aplikasi YAMAL untuk contoh aplikasi, ikuti langkah-langkah pada tab yang cocok dengan sistem operasi gambar Anda.

Linux

Buat file komponen YAMAL

Gunakan alat pengeditan file untuk membuat file bernama `update-linux-os.yaml`. Sertakan konten berikut:

```
# Copyright 2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
#
# Permission is hereby granted, free of charge, to any person obtaining a copy of
# this
# software and associated documentation files (the "Software"), to deal in the
# Software
# without restriction, including without limitation the rights to use, copy, modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
```

```
# permit persons to whom the Software is furnished to do so.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
name: update-linux-os
description: Updates Linux with the latest security updates.
schemaVersion: 1
phases:
  - name: build
    steps:
      - name: UpdateOS
        action: UpdateOS
# Document End
```

 Tip

Gunakan alat seperti [Validator YAMAL online](#) ini, atau ekstensi lint YAMAL di lingkungan kode Anda untuk memverifikasi bahwa YAMAL Anda terbentuk dengan baik.

Windows

Buat file komponen YAMAL

Gunakan alat pengeditan file untuk membuat file bernama *update-windows-os.yaml*. Sertakan konten berikut:

```
# Copyright 2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: MIT-0
#
# Permission is hereby granted, free of charge, to any person obtaining a copy of
# this
# software and associated documentation files (the "Software"), to deal in the
# Software
# without restriction, including without limitation the rights to use, copy, modify,
# merge, publish, distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so.
```

```
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
# IMPLIED,
# INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
# PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
# HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
# OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
name: update-windows-os
description: Updates Windows with the latest security updates.
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: UpdateOS
        action: UpdateOS
# Document End
```

Tip

Gunakan alat seperti [Validator YAMAL online](#) ini, atau ekstensi lint YAMAL di lingkungan kode Anda untuk memverifikasi bahwa YAMAL Anda terbentuk dengan baik.

Kelola parameter AWSTOE komponen dengan EC2 Image Builder

Anda dapat mengelola AWSTOE komponen, termasuk membuat dan menyetel parameter komponen, langsung dari konsol EC2 Image Builder, atau dengan AWS CLI menggunakan perintah, atau salah satu SDK Image Builder. Di bagian ini, kita akan membahas pembuatan dan penggunaan parameter di komponen Anda, dan mengatur parameter komponen melalui konsol dan AWS CLI perintah Image Builder.

Important

Parameter komponen adalah nilai teks biasa, dan masuk AWS CloudTrail. Kami menyarankan Anda menggunakan AWS Secrets Manager atau AWS Systems Manager Parameter Store untuk menyimpan rahasia Anda. Untuk informasi selengkapnya tentang Secrets Manager, lihat [Apa itu Secrets Manager?](#) dalam AWS Secrets Manager User Guide. Untuk

informasi selengkapnya tentang AWS Systems Manager Parameter Store, lihat [AWS Systems Manager Parameter Menyimpan](#) di Panduan AWS Systems Manager Pengguna.

Gunakan parameter dalam dokumen komponen YAMAL Anda

Untuk membangun komponen, sediakan dokumen komponen aplikasi YAMG. Ini mewakili fase dan langkah yang Anda butuhkan untuk membuat komponen. Resep yang mereferensikan komponen dapat mengatur parameter untuk menyesuaikan nilai saat runtime, dengan nilai default yang berlaku jika parameter tidak disetel ke nilai tertentu.

Buat dokumen komponen dengan parameter input

Bagian ini menunjukkan kepada Anda cara menentukan dan menggunakan parameter input dalam dokumen komponen YAMAL Anda.

Untuk membuat dokumen komponen aplikasi YAMG yang menggunakan parameter dan menjalankan perintah di build Image Builder atau instance pengujian, ikuti langkah-langkah yang sesuai dengan sistem operasi image Anda:

Linux

Membuat dokumen komponen YAMG

Gunakan alat pengeditan file untuk membuat file bernama *hello-world-test.yaml*. Sertakan konten berikut:

```
# Document Start
#
name: "HelloWorldTestingDocument-Linux"
description: "Hello world document to demonstrate parameters."
schemaVersion: 1.0
parameters:
  - MyInputParameter:
      type: string
      default: "It's me!"
      description: This is an input parameter.
phases:
  - name: build
    steps:
      - name: HelloWorldStep
        action: ExecuteBash
```

```
    inputs:
      commands:
        - echo "Hello World! Build phase. My input parameter value is
{{ MyInputParameter }}"

- name: validate
  steps:
    - name: HelloWorldStep
      action: ExecuteBash
      inputs:
        commands:
          - echo "Hello World! Validate phase. My input parameter value is
{{ MyInputParameter }}"

- name: test
  steps:
    - name: HelloWorldStep
      action: ExecuteBash
      inputs:
        commands:
          - echo "Hello World! Test phase. My input parameter value is
{{ MyInputParameter }}"
# Document End
```

Tip

Gunakan alat seperti [Validator YAMG online](#) ini, atau ekstensi lint YAMG di lingkungan kode Anda untuk memverifikasi bahwa YAMAL Anda terbentuk dengan baik.

Windows

Membuat dokumen komponen YAMG

Gunakan alat pengeditan file untuk membuat file bernama *hello-world-test.yaml*. Sertakan konten berikut:

```
# Document Start
#
name: "HelloWorldTestingDocument-Windows"
description: "Hello world document to demonstrate parameters."
schemaVersion: 1.0
```

```
parameters:
  - MyInputParameter:
      type: string
      default: "It's me!"
      description: This is an input parameter.
phases:
  - name: build
    steps:
      - name: HelloWorldStep
        action: ExecutePowerShell
        inputs:
          commands:
            - Write-Host "Hello World! Build phase. My input parameter value is
              {{ MyInputParameter }}"

  - name: validate
    steps:
      - name: HelloWorldStep
        action: ExecutePowerShell
        inputs:
          commands:
            - Write-Host "Hello World! Validate phase. My input parameter value is
              {{ MyInputParameter }}"

  - name: test
    steps:
      - name: HelloWorldStep
        action: ExecutePowerShell
        inputs:
          commands:
            - Write-Host "Hello World! Test phase. My input parameter value is
              {{ MyInputParameter }}"
# Document End
```

 Tip

Gunakan alat seperti [Validator YAMG online](#) ini, atau ekstensi lint YAMG di lingkungan kode Anda untuk memverifikasi bahwa YAMAL Anda terbentuk dengan baik.

Untuk informasi selengkapnya tentang fase, langkah, dan sintaks untuk dokumen komponen AWSTOE aplikasi YAMAL, lihat [Menggunakan dokumen](#) di. AWSTOE Untuk informasi selengkapnya

tentang parameter dan persyaratannya, lihat [Parameter](#) bagian Tentukan dan referensi variabel di AWSTOE halaman.

Buat komponen dari dokumen komponen YAMG

Metode apa pun yang Anda gunakan untuk membuat AWSTOE komponen, dokumen komponen aplikasi YAMAL selalu diperlukan sebagai baseline.

- Untuk menggunakan konsol Image Builder untuk membuat komponen langsung dari dokumen YAMAL Anda, lihat [Membuat komponen menggunakan konsol Image Builder](#).
- Untuk menggunakan perintah Image Builder di dalam AWS CLI untuk membuat komponen Anda, lihat [Buat AWSTOE komponen dengan Image Builder dengan AWS CLI](#). Ganti nama dokumen YAMAL dalam contoh tersebut dengan nama dokumen YAMAL Hello World Anda (*hello-world-test.yaml*).

Mengatur parameter komponen dalam resep Image Builder (konsol)

Pengaturan parameter komponen bekerja sama untuk resep gambar dan resep wadah. Saat membuat resep baru, atau resep versi baru, Anda memilih komponen mana yang akan disertakan dari komponen Build dan daftar komponen Uji. Daftar komponen mencakup komponen yang berlaku untuk sistem operasi dasar yang Anda pilih untuk gambar Anda.

Setelah Anda memilih komponen, itu ditampilkan di bagian Komponen yang dipilih, langsung di bawah daftar komponen. Opsi konfigurasi ditampilkan untuk setiap komponen yang dipilih. Jika komponen Anda memiliki parameter input yang ditentukan, parameter tersebut ditampilkan sebagai bagian yang dapat diperluas yang disebut Parameter input.

Pengaturan parameter berikut ditampilkan untuk setiap parameter yang ditentukan untuk komponen Anda:

- Nama parameter (tidak dapat diedit) — Nama parameter.
- Deskripsi (tidak dapat diedit) — Deskripsi parameter
- Jenis (tidak dapat diedit) - Tipe data untuk nilai parameter.
- Nilai — Nilai untuk parameter. Jika Anda menggunakan komponen ini untuk pertama kalinya dalam resep ini, dan nilai default ditentukan untuk komponen, nilai default muncul di kotak Nilai dengan teks berwarna abu-abu. Jika tidak ada nilai lain yang dimasukkan, AWSTOE gunakan nilai default.

Daftar dan lihat detail komponen

Bagian ini menjelaskan bagaimana Anda dapat menemukan informasi dan melihat detail untuk komponen AWS Task Orchestrator and Executor (AWSTOE) yang Anda gunakan dalam resep EC2 Image Builder.

Detail komponen

- [Daftar AWSTOE komponen](#)
- [Daftar versi build komponen \(AWS CLI\)](#)
- [Dapatkan detail komponen \(AWS CLI\)](#)
- [Dapatkan detail kebijakan komponen \(AWS CLI\)](#)

Daftar AWSTOE komponen

Anda dapat menggunakan salah satu metode berikut untuk membuat daftar dan memfilter AWSTOE komponen.

AWS Management Console

Untuk menampilkan daftar komponen di AWS Management Console, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Komponen dari panel navigasi. Secara default, Image Builder menampilkan daftar komponen yang dimiliki akun Anda.
3. Anda dapat memfilter kepemilikan komponen secara opsional. Untuk melihat komponen yang tidak Anda miliki, tetapi memiliki akses, perluas daftar dropdown tipe pemilik dan pilih salah satu nilainya. Daftar jenis pemilik terletak di bilah pencarian, di sebelah kotak teks pencarian. Anda dapat memilih dari nilai-nilai berikut:
 - Mulai cepat (dikelola Amazon) - Komponen yang tersedia untuk umum yang dibuat dan dipelihara Amazon.
 - Dimiliki oleh saya — Komponen yang Anda buat. Ini adalah pilihan default.
 - Berbagi dengan saya — Komponen yang dibuat dan dibagikan orang lain dengan Anda dari akun mereka.
 - Pihak ketiga dikelola — Komponen yang dimiliki pihak ketiga yang Anda langgani. AWS Marketplace

AWS CLI

Contoh berikut menunjukkan cara menggunakan [list-components](#) perintah untuk mengembalikan daftar AWSTOE komponen yang dimiliki akun Anda.

```
aws imagebuilder list-components
```

Anda dapat memfilter kepemilikan komponen secara opsional. Atribut pemilik mendefinisikan siapa yang memiliki komponen yang ingin Anda daftar. Secara default, permintaan ini mengembalikan daftar komponen yang dimiliki akun Anda. Untuk memfilter hasil menurut pemilik komponen, tentukan salah satu nilai berikut dengan `--owner` parameter saat Anda menjalankan `list-components` perintah.

Nilai pemilik komponen

- Mandiri
- Amazon
- ThirdParty
- Bersama

Contoh berikut menunjukkan `list-components` perintah dengan `--owner` parameter untuk memfilter hasil.

```
aws imagebuilder list-components --owner Self
{
  "requestId": "012a3456-b789-01cd-e234-fa5678b9012b",
  "componentVersionList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/sample-component01/1.0.0",
      "name": "sample-component01",
      "version": "1.0.0",
      "platform": "Linux",
      "type": "BUILD",
      "owner": "123456789012",
      "dateCreated": "2020-09-24T16:58:24.444Z"
    },
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/sample-component01/1.0.1",
```

```

        "name": "sample-component01",
        "version": "1.0.1",
        "platform": "Linux",
        "type": "BUILD",
        "owner": "123456789012",
        "dateCreated": "2021-07-10T03:38:46.091Z"
      }
    ]
  }

```

```
aws imagebuilder list-components --owner Amazon
```

```
aws imagebuilder list-components --owner Shared
```

```
aws imagebuilder list-components --owner ThirdParty
```

Daftar versi build komponen (AWS CLI)

Contoh berikut menunjukkan cara menggunakan [list-component-build-versions](#) perintah untuk membuat daftar versi build komponen yang memiliki versi semantik tertentu. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. [Versi semantik](#)

```

aws imagebuilder list-component-build-versions --component-version-arn
arn:aws:imagebuilder:us-west-2:123456789012:component/example-component/1.0.1
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "componentSummaryList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
examplecomponent/1.0.1/1",
      "name": "examplecomponent",
      "version": "1.0.1",
      "platform": "Linux",
      "type": "BUILD",
      "owner": "123456789012",
      "description": "An example component that builds, validates and tests an
image",
      "changeDescription": "Updated version.",
      "dateCreated": "2020-02-19T18:53:45.940Z",
      "tags": {

```

```

        "KeyName": "KeyValue"
      }
    }
  ]
}

```

Dapatkan detail komponen (AWS CLI)

Contoh berikut menunjukkan cara menggunakan [get-component](#) perintah untuk mendapatkan detail komponen saat Anda menentukan Amazon Resource Name (ARN) komponen.

```

aws imagebuilder get-component --component-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/example-component/1.0.1/1
{
  "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11112",
  "component": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
examplecomponent/1.0.1/1",
    "name": "examplecomponent",
    "version": "1.0.1",
    "type": "BUILD",
    "platform": "Linux",
    "owner": "123456789012",
    "data": "name: HelloWorldTestingDocument\ndescription: This is hello world
testing document... etc.\n",
    "encrypted": true,
    "dateCreated": "2020-09-24T16:58:24.444Z",
    "tags": {}
  }
}

```

Dapatkan detail kebijakan komponen (AWS CLI)

Contoh berikut menunjukkan cara menggunakan [get-component-policy](#) perintah untuk mendapatkan rincian kebijakan komponen saat Anda menentukan ARN komponen.

```

aws imagebuilder get-component-policy --component-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/example-component/1.0.1

```

Membuat komponen menggunakan konsol Image Builder

Untuk membuat komponen AWSTOE aplikasi dari konsol Image Builder, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Komponen dari panel navigasi. Kemudian pilih Buat komponen.
3. Pada halaman Create component, di bawah Component details, masukkan yang berikut ini:
 - a. Gambar Sistem operasi (OS). Tentukan sistem operasi yang kompatibel dengan komponen tersebut.
 - b. Kategori komponen. Dari dropdown, pilih jenis komponen build atau test yang Anda buat.
 - c. Nama komponen. Masukkan nama untuk komponen.
 - d. Versi komponen. Masukkan nomor versi komponen.
 - e. Deskripsi. Berikan deskripsi opsional untuk membantu Anda mengidentifikasi komponen.
 - f. Ubah deskripsi. Berikan deskripsi opsional untuk membantu Anda memahami perubahan yang dibuat pada versi komponen ini.
4. Di bagian Dokumen definisi, opsi defaultnya adalah Tentukan konten dokumen. Dokumen komponen mendefinisikan tindakan yang dilakukan Image Builder pada instance build dan test untuk membuat image Anda.

Di kotak Konten, masukkan konten dokumen komponen YAMAL Anda. Untuk memulai dengan contoh Hello World untuk Linux, pilih opsi Use example. Untuk mempelajari lebih lanjut tentang cara membuat dokumen komponen YAMAL, atau menyalin dan menempelkan contoh UpdateOS dari halaman tersebut, lihat. [Membuat dokumen komponen YAMAL](#)

5. Setelah Anda memasukkan detail komponen, pilih Buat komponen.

Note

Untuk melihat komponen baru Anda saat membuat atau memperbarui resep, terapkan filter Owned by me ke daftar komponen build atau test. Filter terletak di bagian atas daftar komponen, di sebelah kotak pencarian.

6. Untuk menghapus komponen, dari halaman Komponen, pilih kotak centang di sebelah komponen yang ingin Anda hapus. Dari dropdown Tindakan, pilih Hapus komponen.

Untuk membuat versi komponen baru, ikuti langkah-langkah berikut:

1. Tergantung di mana Anda memulai:
 - Dari halaman daftar Komponen - Pilih kotak centang di sebelah nama komponen, lalu pilih Buat versi baru dari menu Tindakan.
 - Dari halaman detail komponen - Pilih tombol Buat versi baru di sudut kanan atas judul.
2. Informasi komponen sudah diisi dengan nilai saat ini saat halaman Create Component ditampilkan. Ikuti langkah-langkah membuat komponen untuk memperbarui komponen. Ini memastikan bahwa Anda memasukkan versi semantik unik dalam versi Komponen. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat [Versi semantik](#)

Buat komponen dengan AWS CLI

Bagian ini menjelaskan cara menggunakan perintah Image Builder untuk membuat AWS Task Orchestrator and Executor (AWSTOE) komponen dari AWS Command Line Interface. Untuk membangun komponen, sediakan dokumen komponen aplikasi YAMB. Ini mewakili fase dan langkah yang Anda butuhkan untuk membuat komponen. Untuk membuat dokumen komponen YAMB baru, lihat [Membuat dokumen komponen YAMAL](#).

Buat AWSTOE komponen dengan Image Builder dengan AWS CLI

Di bagian ini, Anda akan belajar cara mengatur dan menggunakan perintah Image Builder di AWS CLI untuk membuat komponen AWSTOE aplikasi, sebagai berikut.

- Unggah dokumen komponen YAMB Anda ke bucket S3 yang dapat Anda referensikan dari baris perintah.
- Buat komponen AWSTOE aplikasi dengan create-component perintah.
- Daftar versi komponen dengan list-components perintah dan filter nama untuk melihat versi apa yang sudah ada. Anda dapat menggunakan output untuk menentukan versi berikutnya untuk pembaruan.

Untuk membuat komponen AWSTOE aplikasi dari dokumen YAMM masukan, ikuti langkah-langkah yang sesuai dengan platform sistem operasi gambar Anda.

Linux

Simpan dokumen komponen aplikasi Anda di Amazon S3

Anda dapat menggunakan bucket S3 sebagai repositori untuk dokumen sumber komponen AWSTOE aplikasi Anda. Untuk menyimpan dokumen komponen Anda, ikuti langkah-langkah berikut:

- Unggah dokumen ke Amazon S3

Jika dokumen Anda lebih kecil dari 64 KB, Anda dapat melewati langkah ini. Dokumen yang berukuran 64 KB atau lebih besar harus disimpan di Amazon S3.

```
aws s3 cp update-linux-os.yaml s3://my-s3-bucket/my-path/update-linux-os.yaml
```

Buat komponen dari dokumen YAMB

Untuk merampingkan create-component perintah yang Anda gunakan AWS CLI, buat file JSON yang berisi semua parameter komponen yang ingin Anda lewatkan ke perintah. Sertakan lokasi *update-linux-os.yaml* dokumen yang Anda buat di langkah sebelumnya. Pasangan uri kunci-nilai berisi referensi file.

Note

Konvensi penamaan untuk nilai data dalam file JSON mengikuti pola yang ditentukan untuk parameter permintaan tindakan Image Builder API. Untuk meninjau parameter permintaan perintah API, lihat [CreateComponent](#) perintah di Referensi API EC2 Image Builder.

Untuk memberikan nilai data sebagai parameter baris perintah, lihat nama parameter yang ditentukan dalam Referensi AWS CLI Perintah.

1. Buat file JSON input CLI

Gunakan alat pengeditan file untuk membuat file bernama *create-update-linux-os-component.json*. Sertakan konten berikut:

```
{  
  "name": "update-linux-os",
```

```
"semanticVersion": "1.1.2",
"description": "An example component that updates the Linux operating system",
"changeDescription": "Initial version.",
"platform": "Linux",
"uri": "s3://my-s3-bucket/my-path/update-linux-os.yaml",
"kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/98765432-
b123-456b-7f89-0123456f789c",
"tags": {
  "MyTagKey-purpose": "security-updates"
}
}
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (`\`) sedangkan Linux menggunakan garis miring (`/`) untuk merujuk ke jalur direktori.

2. Buat komponen

Gunakan perintah berikut untuk membuat komponen, merujuk nama file untuk file JSON yang Anda buat pada langkah sebelumnya:

```
aws imagebuilder create-component --cli-input-json file://create-update-linux-
os-component.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (`\`) sedangkan Linux menggunakan garis miring (`/`) untuk merujuk ke jalur direktori.

Windows

Simpan dokumen komponen aplikasi Anda di Amazon S3

Anda dapat menggunakan bucket S3 sebagai repositori untuk dokumen sumber komponen AWSTOE aplikasi Anda. Untuk menyimpan dokumen komponen Anda, ikuti langkah-langkah berikut:

- Unggah dokumen ke Amazon S3

Jika dokumen Anda lebih kecil dari 64 KB, Anda dapat melewati langkah ini. Dokumen yang berukuran 64 KB atau lebih besar harus disimpan di Amazon S3.

```
aws s3 cp update-windows-os.yaml s3://my-s3-bucket/my-path/update-windows-os.yaml
```

Buat komponen dari dokumen YAMB

Untuk merampingkan create-component perintah yang Anda gunakan AWS CLI, buat file JSON yang berisi semua parameter komponen yang ingin Anda lewatkan ke perintah. Sertakan lokasi *update-windows-os.yaml* dokumen yang Anda buat di langkah sebelumnya. Pasangan uri kunci-nilai berisi referensi file.

Note

Konvensi penamaan untuk nilai data dalam file JSON mengikuti pola yang ditentukan untuk parameter permintaan tindakan Image Builder API. Untuk meninjau parameter permintaan perintah API, lihat [CreateComponent](#) perintah di Referensi API EC2 Image Builder.

Untuk memberikan nilai data sebagai parameter baris perintah, lihat nama parameter yang ditentukan dalam Referensi AWS CLI Perintah..

1. Buat file JSON input CLI

Gunakan alat pengeditan file untuk membuat file bernama *create-update-windows-os-component.json*. Sertakan konten berikut:

```
{
```

```
"name": "update-windows-os",
"semanticVersion": "1.1.2",
"description": "An example component that updates the Windows operating
system.",
"changeDescription": "Initial version.",
"platform": "Windows",
"uri": "s3://my-s3-bucket/my-path/update-windows-os.yaml",
"kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/98765432-
b123-456b-7f89-0123456f789c",
"tags": {
  "MyTagKey-purpose": "security-updates"
}
}
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (`\`) sedangkan Linux menggunakan garis miring (`/`) untuk merujuk ke jalur direktori.

2. Buat komponen

Gunakan perintah berikut untuk membuat komponen, merujuk nama file untuk file JSON yang Anda buat pada langkah sebelumnya:

```
aws imagebuilder create-component --cli-input-json file://create-update-windows-
os-component.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (`\`) sedangkan Linux menggunakan garis miring (`/`) untuk merujuk ke jalur direktori.

AWSTOEpembuatan versi komponen untuk pembaruan () AWS CLI

AWSTOEnama dan versi komponen disematkan di Amazon Resource Name (ARN) komponen, setelah awalan komponen. Setiap versi baru komponen memiliki ARN uniknya sendiri. Langkah-langkah untuk membuat versi baru persis sama dengan langkah-langkah untuk membuat komponen baru, selama versi semantik unik untuk nama komponen itu. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat [Versi semantik](#)

Untuk memastikan bahwa Anda menetapkan versi logis berikutnya, pertama-tama dapatkan daftar versi yang ada untuk komponen yang ingin Anda ubah. Gunakan `list-components` perintah dengan AWS CLI, dan filter pada nama.

Dalam contoh ini, Anda memfilter nama komponen yang Anda buat di contoh Linux sebelumnya. Untuk membuat daftar komponen yang Anda buat, gunakan nilai `name` parameter dari file JSON yang Anda gunakan dalam `create-component` perintah.

```
aws imagebuilder list-components --filters name="name",values="update-linux-os"
{
  "requestId": "123a4567-b890-123c-45d6-ef789ab0cd1e",
  "componentVersionList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:1234560087789012:component/update-
linux-os/1.0.0",
      "name": "update-linux-os",
      "version": "1.0.0",
      "platform": "Linux",
      "type": "BUILD",
      "owner": "123456789012",
      "dateCreated": "2020-09-24T16:58:24.444Z"
    },
    {
      "arn": "arn:aws:imagebuilder:us-west-2:1234560087789012:component/update-
linux-os/1.0.1",
      "name": "update-linux-os",
      "version": "1.0.1",
      "platform": "Linux",
      "type": "BUILD",
      "owner": "123456789012",
      "dateCreated": "2021-07-10T03:38:46.091Z"
    }
  ]
}
```

Berdasarkan hasil Anda, Anda dapat menentukan versi berikutnya.

Impor komponen (AWS CLI)

Untuk beberapa skenario, mungkin lebih mudah untuk memulai dengan skrip yang sudah ada sebelumnya. Untuk skenario ini, Anda dapat menggunakan contoh berikut.

Contoh ini mengasumsikan bahwa Anda memiliki file bernama *import-component.json* (seperti yang ditunjukkan). Perhatikan bahwa file tersebut secara langsung mereferensikan PowerShell skrip AdminConfig.ps1 yang disebut yang sudah diunggah. *my-s3-bucket* Saat SHELL ini, didukung untuk komponenformat.

```
{
  "name": "MyImportedComponent",
  "semanticVersion": "1.0.0",
  "description": "An example of how to import a component",
  "changeDescription": "First commit message.",
  "format": "SHELL",
  "platform": "Windows",
  "type": "BUILD",
  "uri": "s3://my-s3-bucket/AdminConfig.ps1",
  "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/60763706-
b131-418b-8f85-3420912f020c"
}
```

Untuk mengimpor komponen, jalankan perintah berikut.

```
aws imagebuilder import-component --cli-input-json file://import-component.json
```

Pembersihan sumber daya

Untuk menghindari biaya yang tidak terduga, pastikan untuk membersihkan sumber daya dan saluran pipa yang Anda buat dari contoh dalam panduan ini. Untuk informasi selengkapnya tentang menghapus sumber daya di Image Builder, lihat [Hapus sumber EC2 Image Builder](#).

Kelola resep

Resep EC2 Image Builder mendefinisikan gambar dasar yang akan digunakan sebagai titik awal Anda untuk membuat gambar baru, bersama dengan kumpulan komponen yang Anda tambahkan untuk menyesuaikan gambar Anda dan memverifikasi bahwa semuanya berfungsi seperti yang

diharapkan. Image Builder menyediakan pilihan versi otomatis untuk setiap komponen. Jumlah komponen yang dapat Anda terapkan pada resep dibatasi hingga 20 komponen secara keseluruhan. Ini termasuk komponen build dan test.

Setelah Anda membuat resep, Anda tidak dapat memodifikasi atau menggantinya. Untuk memperbarui komponen setelah Anda membuat resep, Anda harus membuat resep atau versi resep baru. Anda selalu dapat menerapkan tag ke resep yang ada. Untuk informasi selengkapnya tentang menandai sumber daya Anda menggunakan perintah Image Builder diAWS CLI, lihat [Memberi tanda pada sumber daya](#) bagian panduan ini.

Tip

Anda dapat menggunakan komponen terkelola Amazon dalam resep Anda, atau Anda dapat mengembangkan komponen kustom Anda sendiri dengan aplikasi AWS Task Orchestrator and Executor (AWSTOE). Untuk memulai, lihat [Memulai dengan AWSTOE](#).

Bagian ini mencakup cara membuat daftar, melihat, dan membuat resep.

Daftar Isi

- [Daftar dan lihat detail resep gambar](#)
- [Daftar dan lihat detail resep wadah](#)
- [Buat versi baru dari resep gambar](#)
- [Buat versi baru resep wadah](#)
- [Pembersihan sumber daya](#)

Daftar dan lihat detail resep gambar

Bagian ini menjelaskan berbagai cara agar Anda dapat menemukan informasi dan melihat detail untuk resep gambar EC2 Image Builder Anda.

Detail resep gambar

- [Daftar resep gambar \(konsol\)](#)
- [Daftar resep gambar \(AWS CLI\)](#)
- [Lihat detail resep gambar \(konsol\)](#)
- [Dapatkan detail resep gambar \(AWS CLI\)](#)

- [Dapatkan detail kebijakan resep gambar \(AWS CLI\)](#)

Daftar resep gambar (konsol)

Untuk melihat daftar resep gambar yang dibuat di bawah akun Anda di konsol Image Builder, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Resep gambar dari panel navigasi. Ini menunjukkan daftar resep gambar yang dibuat di bawah akun Anda.
3. Untuk melihat detail atau membuat versi resep baru, pilih tautan Nama resep. Ini membuka tampilan detail untuk resep.

Note

Anda juga dapat memilih kotak centang di sebelah nama Resep, lalu pilih Lihat detail.

Daftar resep gambar (AWS CLI)

Contoh berikut menunjukkan bagaimana untuk daftar semua resep gambar Anda, menggunakan AWS CLI.

```
aws imagebuilder list-image-recipes
```

Lihat detail resep gambar (konsol)

Untuk melihat detail resep gambar tertentu menggunakan konsol Image Builder, pilih resep gambar yang akan ditinjau, menggunakan langkah-langkah yang dijelaskan di [Daftar resep gambar \(konsol\)](#).

Pada halaman detail resep, Anda dapat:

- Hapus resepnya. Untuk informasi selengkapnya tentang menghapus sumber daya di Image Builder, lihat [Hapus sumber EC2 Image Builder](#).
- Buat versi baru.
- Buat pipa dari resep. Setelah memilih Buat pipeline dari resep ini, Anda akan dibawa ke wizard pipeline. Untuk informasi selengkapnya tentang membuat pipeline Image Builder menggunakan panduan pipeline, lihat [Buat pipeline gambar menggunakan wizard konsol EC2 Image Builder](#)

Note

Saat Anda membuat pipeline dari resep yang ada, opsi untuk membuat resep baru tidak tersedia.

Dapatkan detail resep gambar (AWS CLI)

Contoh berikut menunjukkan cara menggunakan perintah imagebuilder CLI untuk mendapatkan detail resep gambar dengan menentukan Nama Sumber Daya Amazon (ARN).

```
aws imagebuilder get-image-recipe --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-recipe/2020.12.03
```

Dapatkan detail kebijakan resep gambar (AWS CLI)

Contoh berikut menunjukkan cara menggunakan perintah imagebuilder CLI untuk mendapatkan detail kebijakan resep gambar dengan menentukan ARN-nya.

```
aws imagebuilder get-image-recipe-policy --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-recipe/2020.12.03
```

Daftar dan lihat detail resep wadah

Bagian ini menjelaskan cara Anda dapat menemukan informasi dan melihat detail untuk resep wadah EC2 Image Builder Anda.

Detail resep wadah

- [Daftar resep wadah di konsol](#)
- [Daftar resep wadah dengan AWS CLI](#)
- [Lihat detail resep wadah di konsol](#)
- [Dapatkan detail resep wadah dengan AWS CLI](#)
- [Dapatkan detail kebijakan resep kontainer dengan AWS CLI](#)

Daftar resep wadah di konsol

Untuk melihat daftar resep kontainer yang telah dibuat di bawah akun Anda di konsol Image Builder, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih resep kontainer dari panel navigasi. Ini menunjukkan daftar resep wadah yang dibuat di bawah akun Anda.
3. Untuk melihat detail atau membuat versi resep baru, pilih tautan Nama resep. Ini membuka tampilan detail untuk resep.

Note

Anda juga dapat memilih kotak centang di sebelah nama Resep, lalu pilih Lihat detail.

Daftar resep wadah dengan AWS CLI

Contoh berikut menunjukkan cara membuat daftar semua resep wadah Anda, menggunakan AWS CLI.

```
aws imagebuilder list-container-recipes
```

Lihat detail resep wadah di konsol

Untuk melihat detail resep kontainer tertentu dengan konsol Image Builder, pilih resep penampung yang akan ditinjau, dan gunakan langkah-langkah yang dijelaskan di dalamnya [Daftar resep wadah di konsol](#).

Pada halaman detail resep, Anda dapat melakukan hal berikut:

- Hapus resepnya. Untuk informasi selengkapnya tentang cara menghapus sumber daya di Image Builder, lihat [Hapus sumber EC2 Image Builder](#).
- Buat versi baru.
- Buat pipa dari resep. Setelah Anda memilih Buat pipeline dari resep ini, Anda akan dibawa ke wizard pipeline. Untuk informasi selengkapnya tentang cara membuat pipeline Image Builder menggunakan panduan pipeline, lihat [Buat pipeline gambar menggunakan wizard konsol EC2 Image Builder](#)

Note

Saat Anda membuat pipeline dari resep yang ada, opsi untuk membuat resep baru tidak tersedia.

Dapatkan detail resep wadah dengan AWS CLI

Contoh berikut menunjukkan cara menggunakan perintah imagebuilder CLI untuk mendapatkan rincian resep wadah dengan menentukan ARN nya.

```
aws imagebuilder get-container-recipe --container-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:container-recipe/my-example-recipe/2020.12.03
```

Dapatkan detail kebijakan resep kontainer dengan AWS CLI

Contoh berikut menunjukkan cara menggunakan perintah imagebuilder CLI untuk mendapatkan rincian kebijakan resep kontainer dengan menentukan ARN nya.

```
aws imagebuilder get-container-recipe-policy --container-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:container-recipe/my-example-recipe/2020.12.03
```

Buat versi baru dari resep gambar

Bagian ini menjelaskan cara membuat versi baru dari resep gambar.

Daftar Isi

- [Buat versi resep gambar baru \(konsol\)](#)
- [Buat resep gambar dengan AWS CLI](#)
- [Impor VM sebagai gambar dasar Anda di konsol](#)

Buat versi resep gambar baru (konsol)

Saat Anda membuat versi resep baru, itu hampir sama dengan membuat resep baru. Perbedaannya adalah bahwa detail tertentu dipilih sebelumnya agar sesuai dengan resep dasar, dalam banyak

kasus. Daftar berikut menjelaskan perbedaan antara membuat resep baru dan membuat versi baru dari resep yang sudah ada.


Detail resep dasar dalam versi baru

- Nama - Tidak dapat diedit.
- Versi - Diperlukan. Detail dasar ini tidak diisi sebelumnya dengan versi saat ini atau jenis urutan apa pun. Masukkan nomor versi yang ingin Anda buat dalam format <major>. <minor>. <patch>. Jika versi sudah ada, Anda mengalami kesalahan.
- Opsi Pilih gambar - Dipilih sebelumnya, tetapi Anda dapat mengeditnya. Jika Anda mengubah pilihan Anda untuk sumber gambar dasar Anda, Anda mungkin kehilangan detail lain yang bergantung pada opsi asli yang Anda pilih.

Untuk melihat detail yang terkait dengan pemilihan gambar dasar Anda, pilih tab yang cocok dengan pilihan Anda.

Managed image

- Sistem Operasi Gambar (OS) - Tidak dapat diedit.
- Nama gambar — Dipilih sebelumnya, berdasarkan kombinasi pilihan gambar dasar yang Anda buat untuk resep yang ada. Namun, jika Anda mengubah opsi Pilih gambar, Anda kehilangan nama Gambar yang dipilih sebelumnya.
- Opsi versi otomatis — Tidak cocok dengan resep dasar Anda. Opsi gambar ini default ke opsi Gunakan versi OS yang dipilih.

 Important

Jika Anda menggunakan versi semantik untuk memulai build pipeline, pastikan Anda mengubah nilai ini menjadi Gunakan versi OS terbaru yang tersedia. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. [Versi semantik](#)

AWS Marketplace image

- Langganan — Tab ini harus terbuka, dan gambar berlangganan dari AWS Marketplace harus dipilih sebelumnya agar sesuai dengan resep dasar Anda. Jika Anda mengubah gambar yang digunakan resep Anda sebagai gambar dasarnya, Anda mungkin kehilangan detail lain yang bergantung pada gambar asli yang Anda pilih.

Untuk informasi selengkapnya tentang AWS Marketplace produk, lihat [Membeli produk](#) di Panduan AWS Marketplace Pembeli.

Custom AMI

- ID AMI - Diperlukan. Namun, pengaturan ini tidak diisi sebelumnya dengan entri asli Anda. Anda harus memasukkan ID AMI untuk gambar dasar Anda.
- Konfigurasi instans - Pengaturan sudah dipilih sebelumnya, tetapi Anda dapat mengeditnya.
- Agen Systems Manager - Anda dapat memilih atau mengosongkan kotak centang ini untuk mengontrol pemasangan agen Systems Manager pada gambar baru. Kotak centang dihapus secara default untuk menyertakan agen Systems Manager dalam gambar baru Anda. Untuk menghapus agen Systems Manager dari gambar akhir, pilih kotak centang agar agen tidak disertakan dalam AMI Anda.
- Data pengguna — Anda dapat menggunakan area ini untuk menyediakan perintah, atau skrip perintah untuk dijalankan, saat meluncurkan instance build. Namun, nilai ini menggantikan perintah apa pun yang mungkin ditambahkan Image Builder untuk memastikan bahwa Systems Manager diinstal. Perintah ini termasuk skrip pembersihan yang biasanya dijalankan Image Builder untuk gambar Linux sebelum membuat gambar baru.

Note

- Jika Anda memasukkan data pengguna, pastikan bahwa agen Systems Manager sudah diinstal sebelumnya pada image dasar Anda, atau bahwa Anda menyertakan instalasi dalam data pengguna Anda.
- Untuk gambar Linux, pastikan bahwa langkah-langkah pembersihan dijalankan dengan menyertakan perintah untuk membuat file kosong bernama `perform_cleanup` dalam skrip data pengguna Anda. Image Builder mendeteksi file ini, dan menjalankan skrip pembersihan sebelum membuat gambar baru. Untuk informasi selengkapnya dan contoh skrip, lihat [Praktik terbaik keamanan untuk EC2 Image Builder](#).

- Direktori kerja - Dipilih sebelumnya, tetapi Anda dapat mengeditnya.
- Komponen — Komponen yang sudah disertakan dalam resep ditampilkan di bagian Komponen yang dipilih di akhir setiap daftar komponen (build dan test). Anda dapat menghapus atau menyusun ulang komponen yang dipilih sesuai dengan kebutuhan Anda.

Komponen pengerasan CIS tidak mengikuti aturan pemesanan komponen standar dalam resep Image Builder. Komponen pengerasan CIS selalu berjalan terakhir untuk memastikan bahwa tes benchmark berjalan terhadap gambar keluaran Anda.

Note

Daftar komponen build dan test menampilkan komponen yang tersedia berdasarkan tipe pemilik komponen. Untuk menambah atau memperbarui komponen resep Anda, pilih jenis pemilik untuk komponen yang Anda cari. Misalnya, jika Anda ingin menambahkan komponen yang terkait dengan gambar dasar tempat Anda berlangganan AWS Marketplace, pilih `Third party managed` dari daftar jenis pemilik, di samping bilah pencarian.

Anda dapat mengonfigurasi pengaturan berikut untuk komponen yang Anda pilih:

- Opsi pembuatan versi - Dipilih sebelumnya, tetapi Anda dapat mengubahnya. Kami menyarankan Anda memilih opsi `Gunakan versi komponen terbaru yang tersedia` untuk memastikan bahwa build gambar Anda selalu mengambil versi terbaru dari komponen tersebut. Jika Anda perlu menggunakan versi komponen tertentu dalam resep Anda, Anda dapat memilih `Tentukan versi komponen`, dan masukkan versi dalam kotak `Versi komponen yang muncul`.
- Parameter input - Menampilkan parameter input yang diterima komponen. Nilai sudah diisi sebelumnya dengan nilai dari versi resep sebelumnya. Jika Anda menggunakan komponen ini untuk pertama kalinya dalam resep ini, dan nilai default ditentukan untuk komponen, nilai default muncul di kotak `Nilai` dengan teks berwarna abu-abu. Jika tidak ada nilai lain yang dimasukkan, `AWSTOE` gunakan nilai default.

Important

Parameter komponen adalah nilai teks biasa, dan masuk AWS CloudTrail. Kami menyarankan Anda menggunakan AWS Secrets Manager atau AWS Systems Manager Parameter Store untuk menyimpan rahasia Anda. Untuk informasi selengkapnya tentang Secrets Manager, lihat [Apa itu Secrets Manager?](#) dalam AWS Secrets Manager User Guide. Untuk informasi selengkapnya tentang AWS Systems Manager Parameter Store, lihat [AWS Systems Manager Parameter Store](#) di Panduan AWS Systems Manager Pengguna.

Untuk memperluas pengaturan untuk opsi Pembuatan versi atau Parameter input, Anda dapat memilih panah di sebelah nama pengaturan. Untuk memperluas semua pengaturan untuk semua komponen yang dipilih, Anda dapat mengaktifkan tombol Expand all off and on.

- Penyimpanan (volume) — sudah diisi sebelumnya. Volume root Nama perangkat, Snapshot, dan pilihan IOPS, tidak dapat diedit. Namun, Anda dapat mengubah semua pengaturan yang tersisa, seperti Ukuran. Anda juga dapat menambahkan volume baru, dan mengenkripsi volume baru atau yang sudah ada.

Untuk mengenkripsi volume gambar yang dibuat Image Builder di bawah akun Anda di Wilayah sumber (tempat build berjalan), Anda harus menggunakan enkripsi volume penyimpanan dalam resep gambar. Enkripsi yang berjalan selama fase distribusi build hanya untuk gambar yang didistribusikan ke akun atau Wilayah lain.

Untuk membuat versi resep gambar baru:

1. Di bagian atas halaman detail resep, pilih Buat versi baru. Ini membawa Anda ke halaman resep Buat gambar.
2. Untuk membuat versi baru, buat perubahan, lalu pilih Buat resep gambar.

Untuk informasi selengkapnya tentang cara membuat resep gambar saat Anda membuat pipeline gambar, lihat [Langkah 2: Pilih resep](#) di bagian Memulai panduan ini.

Buat resep gambar dengan AWS CLI

Untuk membuat resep gambar dengan `create-image-recipe` perintah Image Builder di AWS CLI, ikuti langkah-langkah berikut:

Prasyarat

Sebelum Anda menjalankan perintah Image Builder di bagian ini untuk membuat resep gambar dari AWS CLI, Anda harus membuat komponen yang digunakan resep. Contoh resep gambar pada langkah berikut mengacu pada komponen contoh yang dibuat di [Buat komponen dengan AWS CLI](#) bagian panduan ini.

Setelah Anda membuat komponen Anda, atau jika Anda menggunakan komponen yang ada, perhatikan ARN yang ingin Anda sertakan dalam resep.

1. Buat file JSON input CLI

Anda dapat memberikan semua input untuk `create-image-recipe` perintah dengan parameter perintah inline. Namun, perintah yang dihasilkan bisa sangat panjang. Untuk merampingkan perintah, Anda dapat menyediakan file JSON yang berisi semua pengaturan resep.

Note

Konvensi penamaan untuk nilai data dalam file JSON mengikuti pola yang ditentukan untuk parameter permintaan tindakan Image Builder API. Untuk meninjau parameter permintaan perintah API, lihat [CreateImageRecipe](#) perintah di Referensi API EC2 Image Builder.

Untuk memberikan nilai data sebagai parameter baris perintah, lihat nama parameter yang ditentukan dalam Referensi AWS CLI Perintah.

Berikut adalah ringkasan parameter yang ditentukan oleh contoh-contoh ini:

- `nama` (string, wajib) — Nama resep gambar.
- `deskripsi` (string) — Deskripsi resep gambar.
- `ParentImage` (string, required) - Gambar yang digunakan resep gambar sebagai dasar untuk gambar kustom Anda. Nilainya bisa berupa gambar dasar ARN atau ID AMI.

Note

Contoh Linux menggunakan Image Builder AMI, dan contoh Windows menggunakan ARN.

- `SemanticVersion <major>`(string, required) — Versi semantik dari resep gambar, dinyatakan dalam format berikut, dengan nilai numerik di setiap posisi untuk menunjukkan versi tertentu: `<minor>`. `<patch>`. Misalnya nilai mungkin `1.0.0`. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat [Versi semantik](#)
- `komponen` (array, diperlukan) - Berisi array `ComponentConfiguration` objek. Setidaknya satu komponen build harus ditentukan:

Note

Image Builder menginstal komponen dalam urutan yang Anda tentukan dalam resep. Namun, komponen pengerasan CIS selalu berjalan terakhir untuk memastikan bahwa tes benchmark berjalan terhadap gambar keluaran Anda.

- `ComponentARN` (string, diperlukan) — Komponen ARN.

Tip

Untuk menggunakan salah satu contoh untuk membuat resep gambar Anda sendiri, Anda harus mengganti contoh ARN dengan ARN untuk komponen yang Anda gunakan untuk resep Anda.

- `parameter` (array objek) — Berisi array `ComponentParameter` objek.

Important

Parameter komponen adalah nilai teks biasa, dan masuk AWS CloudTrail. Kami menyarankan Anda menggunakan AWS Secrets Manager atau AWS Systems Manager Parameter Store untuk menyimpan rahasia Anda. Untuk informasi selengkapnya tentang Secrets Manager, lihat [Apa itu Secrets Manager?](#) dalam AWS Secrets Manager User Guide. Untuk informasi selengkapnya tentang AWS Systems Manager Parameter Store, lihat [AWS Systems Manager Parameter Store](#) di Panduan AWS Systems Manager Pengguna.

- `name` (string, required) — Nama parameter komponen yang akan ditetapkan.
- `value` (array string, required) — Berisi array string untuk mengatur nilai untuk parameter komponen bernama. Jika ada nilai default yang ditentukan untuk komponen, dan tidak ada nilai lain yang disediakan, AWSTOE gunakan nilai default.
- `additionalInstanceConfiguration`(object) — Tentukan setelan tambahan dan luncurkan skrip untuk instance build Anda.
- `systemsManagerAgent`(object) — Berisi pengaturan untuk agen Systems Manager pada instance build Anda.

- `uninstallAfterBuild(Boolean)` — Mengontrol apakah agen Systems Manager dihapus dari image build akhir Anda sebelum membuat AMI baru. Jika opsi ini diatur ke `true`, maka agen dihapus dari gambar akhir. Jika opsi diatur ke `false`, maka agen dibiarkan sehingga disertakan dalam AMI baru. Nilai default-nya adalah `false`.

Note

Jika `uninstallAfterBuild` atribut tidak disertakan dalam file JSON, dan kondisi berikut benar, maka Image Builder menghapus agen Systems Manager dari gambar akhir sehingga tidak tersedia di AMI:

- `userDataOverrideKosong` atau telah dihilangkan dari file JSON.
- Image Builder secara otomatis menginstal agen Systems Manager pada instance build untuk sistem operasi yang tidak memiliki agen pra-instal pada image dasar.

- `userDataOverride(string)` — Berikan perintah atau skrip perintah untuk dijalankan saat Anda meluncurkan instance build.

Note

Data pengguna selalu dikodekan dengan basis 64. Misalnya, perintah berikut dikodekan sebagai:

IyEvYm1uL2Jhc2gKbWtkaXIgLXAgL3Zhci9iYi8KdG91Y2ggL3Zhcg==

```
#!/bin/bash
mkdir -p /var/bb/
touch /var
```

Contoh Linux menggunakan nilai yang dikodekan ini.

Linux

Gambar dasar (`parentImageproperty`) dalam contoh berikut adalah AMI. Saat Anda menggunakan AMI, Anda harus memiliki akses ke AMI, dan AMI harus berada di Wilayah sumber (Wilayah yang sama tempat Image Builder menjalankan perintah). Simpan file sebagai `create-image-recipe.json`, dan gunakan dalam `create-image-recipe` perintah.


```
{
  "name": "BB Ubuntu Image recipe",
  "description": "Hello World image recipe for Linux.",
  "parentImage": "ami-0a01b234c5de6fab",
  "semanticVersion": "1.0.0",
  "components": [
    {
      "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/bb$"
    }
  ],
  "additionalInstanceConfiguration": {
    "systemsManagerAgent": {
      "uninstallAfterBuild": true
    },
    "userDataOverride": "IyEvYmluL2Jhc2gKbWtkaXIgLXAgL3Zhci9iYi8KdG91Y2ggL3Zhcg=="
  }
}
```

Windows

Contoh berikut mengacu pada versi terbaru dari gambar Windows Server 2016 English Full Base. ARN dalam contoh ini mereferensikan gambar terbaru di SKU berdasarkan filter versi semantik yang telah Anda tentukan: `arn:aws:imagebuilder:us-west-2:aws:image/windows-server-2016-english-full-base-x86/x.x.x`

```
{
  "name": "MyBasicRecipe",
  "description": "This example image recipe creates a Windows 2016 image.",
  "parentImage": "arn:aws:imagebuilder:us-west-2:aws:image/windows-server-2016-english-full-base-x86/x.x.x",
  "semanticVersion": "1.0.0",
  "components": [
    {
      "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-component/2019.12.02/1"
    },
    {
      "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/my-imported-component/1.0.0/1"
    }
  ]
}
```

```
}
```

Note

Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. [Versi semantik](#)

2. Buat resepnya

Gunakan perintah berikut untuk membuat resep. Berikan nama file JSON yang Anda buat pada langkah sebelumnya dalam `--cli-input-json` parameter:

```
aws imagebuilder create-image-recipe --cli-input-json file://create-image-recipe.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (`\`) sedangkan Linux menggunakan garis miring (`/`) untuk merujuk ke jalur direktori.

Impor VM sebagai gambar dasar Anda di konsol

Pada bagian ini, kami fokus pada cara mengimpor mesin virtual (VM) sebagai gambar dasar untuk resep gambar Anda. Kami tidak membahas langkah-langkah lain yang terlibat dengan membuat resep atau versi resep di sini. Untuk langkah-langkah tambahan untuk membuat resep gambar baru dengan wizard pembuatan pipeline di konsol Image Builder, lihat [Buat pipeline gambar \(AMI\)](#). Untuk langkah-langkah tambahan untuk membuat resep gambar atau versi resep baru, lihat [Buat versi baru dari resep gambar](#).

Untuk mengimpor VM sebagai gambar dasar untuk resep gambar Anda di konsol Image Builder, ikuti langkah-langkah berikut, bersama dengan langkah-langkah lain yang diperlukan, untuk membuat versi resep atau resep Anda.

1. Di bagian Pilih gambar untuk gambar dasar, pilih opsi Impor gambar dasar.

2. Pilih Image Operating System (OS) dan versi OS seperti biasanya.

Konfigurasi impor VM

Saat Anda mengekspor VM Anda dari lingkungan virtualisasi, proses itu membuat satu set file wadah disk atau lebih yang bertindak sebagai snapshot dari lingkungan, pengaturan, dan data VM Anda. Anda dapat menggunakan file-file ini untuk mengimpor VM Anda sebagai gambar dasar untuk resep gambar Anda. Untuk informasi selengkapnya tentang mengimpor VM di Image Builder, lihat [Impor dan ekspor gambar VM](#)

Untuk menentukan lokasi sumber impor Anda, ikuti langkah-langkah berikut:

Impor sumber

Tentukan sumber untuk wadah disk gambar VM pertama atau snapshot yang akan diimpor di bagian Disk container 1.

1. Sumber — Ini bisa berupa bucket S3 atau snapshot EBS.
2. Pilih lokasi disk S3 — Masukkan lokasi di Amazon S3 tempat gambar disk Anda disimpan. Untuk menelusuri lokasi, pilih Browse S3.
3. Untuk menambahkan wadah disk, pilih Tambahkan wadah disk.

IAM role

Untuk mengaitkan peran IAM dengan konfigurasi impor VM Anda, pilih peran dari daftar tarik-turun peran IAM, atau pilih Buat peran baru untuk membuat peran baru. Jika Anda membuat peran baru, halaman konsol Peran IAM akan terbuka di tab terpisah.

Pengaturan lanjutan - opsional

Pengaturan berikut bersifat opsional. Dengan pengaturan ini, Anda dapat mengonfigurasi enkripsi, lisensi, tag, dan lainnya untuk gambar dasar yang dibuat impor.

Umum

1. Tentukan Nama unik untuk gambar dasar. Jika Anda tidak memasukkan nilai, gambar dasar mewarisi nama resep.
2. Tentukan Versi untuk gambar dasar. Gunakan format berikut: `<major>.<minor>.<patch>`. Jika Anda tidak memasukkan nilai, gambar dasar mewarisi versi resep.

3. Anda juga dapat memasukkan Deskripsi untuk gambar dasar.

Arsitektur gambar dasar

Untuk menentukan arsitektur sumber impor VM Anda, pilih nilai dari daftar Arsitektur.

Enkripsi

Jika gambar disk VM Anda dienkripsi, Anda harus memberikan kunci untuk digunakan untuk proses impor. Untuk menentukan impor, pilih nilai dari daftar Enkripsi (kunci KMS). AWS KMS key Daftar ini berisi kunci KMS yang dapat diakses akun Anda di Wilayah saat ini.

Manajemen lisensi

Saat Anda mengimpor VM, proses impor secara otomatis mendeteksi OS VM dan menerapkan lisensi yang sesuai ke gambar dasar. Bergantung pada platform OS Anda, jenis lisensinya adalah sebagai berikut:

- Termasuk lisensi - AWS Lisensi yang sesuai untuk platform Anda diterapkan pada gambar dasar Anda.
- Bawa lisensi Anda sendiri (BYOL) - Mempertahankan lisensi dari VM Anda, jika berlaku.

Untuk melampirkan konfigurasi lisensi yang dibuat AWS License Manager dengan gambar dasar Anda, pilih dari daftar nama konfigurasi Lisensi. Untuk informasi selengkapnya tentang License Manager, lihat [Bekerja dengan AWS License Manager](#)

Note

- Konfigurasi lisensi berisi aturan lisensi berdasarkan ketentuan perjanjian perusahaan Anda.
- Linux hanya mendukung lisensi BYOL.

Tag (gambar dasar)

Tag menggunakan pasangan nilai kunci untuk menetapkan teks yang dapat dicari ke sumber daya Image Builder Anda. Untuk menentukan tag untuk gambar dasar yang diimpor, masukkan pasangan kunci-nilai dengan kotak Kunci dan Nilai.

Untuk menambahkan tag, pilih Tambahkan tag. Untuk menghapus sebuah tag, pilih Hapus tag.

Buat versi baru resep wadah

Bagian ini menunjukkan cara membuat versi baru dari resep wadah.

Daftar Isi

- [Buat versi resep wadah baru dengan konsol](#)
- [Buat resep wadah dengan AWS CLI](#)

Buat versi resep wadah baru dengan konsol

Membuat versi baru dari resep wadah hampir sama dengan membuat resep baru. Perbedaannya adalah bahwa detail tertentu dipilih sebelumnya agar sesuai dengan resep dasar, dalam banyak kasus. Daftar berikut menjelaskan perbedaan antara membuat resep baru dan membuat versi baru dari resep yang sudah ada.

Detail resep

- Nama - tidak dapat diedit.
- Versi - Diperlukan. Detail ini tidak diisi sebelumnya dengan versi saat ini atau jenis urutan apa pun. Masukkan nomor versi yang ingin Anda buat dalam format `major.minor.patch`. Jika versi sudah ada, Anda mengalami kesalahan.

Gambar dasar

- Pilih opsi gambar - Dipilih sebelumnya, tetapi dapat diedit. Jika Anda mengubah pilihan Anda untuk sumber gambar dasar Anda, Anda mungkin kehilangan detail lain yang bergantung pada opsi asli yang Anda pilih.

Untuk melihat detail yang terkait dengan pemilihan gambar dasar Anda, pilih tab yang cocok dengan pilihan Anda.

Managed images

- Sistem Operasi Gambar (OS) - Tidak dapat diedit.
- Nama gambar — Dipilih sebelumnya, berdasarkan kombinasi pilihan gambar dasar yang Anda buat untuk resep yang ada. Namun, jika Anda mengubah opsi Pilih gambar, Anda kehilangan nama Gambar yang telah dipilih sebelumnya.
- Opsi versi otomatis — Tidak cocok dengan resep dasar Anda. Opsi versi otomatis default ke opsi Gunakan versi OS yang dipilih.

⚠ Important

Jika Anda menggunakan versi semantik untuk memulai build pipeline, pastikan Anda mengubah nilai ini menjadi Gunakan versi OS terbaru yang tersedia. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. [Versi semantik](#)

ECR image

- Sistem Operasi Gambar (OS) - Dipilih sebelumnya, tetapi dapat diedit.
- Versi OS - Dipilih sebelumnya, tetapi dapat diedit.
- ID gambar ECR - Diisi sebelumnya, tetapi dapat diedit.

Docker Hub image

- Sistem Operasi Gambar (OS) - Tidak dapat diedit.
- Versi OS - Dipilih sebelumnya, tetapi dapat diedit.
- ID gambar Docker - Diisi sebelumnya, tetapi dapat diedit.

Konfigurasi instans

- ID AMI - Diisi sebelumnya, tetapi dapat diedit.
- Penyimpanan (volume)

EBS volume 1 (root AMI) — Diisi sebelumnya. Anda tidak dapat mengedit volume root Nama perangkat, Snapshot, atau pilihan IOPS. Namun, Anda dapat mengubah semua pengaturan yang tersisa, seperti Ukuran. Anda juga dapat menambahkan volume baru.

ℹ Note

Jika Anda menetapkan AMI dasar yang dibagikan dengan Anda dari akun lain, snapshot untuk volume sekunder apa pun yang ditentukan juga harus dibagikan dengan akun Anda.

Direktori kerja

- Jalur direktori kerja - Diisi sebelumnya, tetapi dapat diedit.

Komponen-komponen

- **Komponen** — Komponen yang sudah disertakan dalam resep ditampilkan di bagian Komponen yang dipilih di akhir setiap daftar komponen (build dan test). Anda dapat menghapus atau menyusun ulang komponen yang dipilih sesuai dengan kebutuhan Anda.

Komponen pengerasan CIS tidak mengikuti aturan pemesanan komponen standar dalam resep Image Builder. Komponen pengerasan CIS selalu berjalan terakhir untuk memastikan bahwa tes benchmark berjalan terhadap gambar keluaran Anda.

Note

Daftar komponen build dan test menampilkan komponen yang tersedia berdasarkan tipe pemilik komponen. Untuk menambah atau memperbarui komponen resep Anda, pilih jenis pemilik untuk komponen yang Anda cari. Misalnya, jika Anda ingin menambahkan komponen yang terkait dengan gambar dasar tempat Anda berlangganan AWS Marketplace, pilih `Third party managed` dari daftar jenis pemilik, di samping bilah pencarian.

Anda dapat mengonfigurasi pengaturan berikut untuk komponen yang Anda pilih:

- **Opsi pembuatan versi** - Dipilih sebelumnya, tetapi Anda dapat mengubahnya. Kami menyarankan Anda memilih opsi `Gunakan versi komponen terbaru` yang tersedia untuk memastikan bahwa build gambar Anda selalu mengambil versi terbaru dari komponen tersebut. Jika Anda perlu menggunakan versi komponen tertentu dalam resep Anda, Anda dapat memilih `Tentukan versi komponen`, dan masukkan versi dalam kotak `Versi komponen` yang muncul.
- **Parameter input** - Menampilkan parameter input yang diterima komponen. Nilai sudah diisi sebelumnya dengan nilai dari versi resep sebelumnya. Jika Anda menggunakan komponen ini untuk pertama kalinya dalam resep ini, dan nilai default ditentukan untuk komponen, nilai default muncul di kotak `Nilai` dengan teks berwarna abu-abu. Jika tidak ada nilai lain yang dimasukkan, `AWSTOE` gunakan nilai default.

Important

Parameter komponen adalah nilai teks biasa, dan masuk AWS CloudTrail. Kami menyarankan Anda menggunakan AWS Secrets Manager atau AWS Systems Manager Parameter Store untuk menyimpan rahasia Anda. Untuk informasi selengkapnya tentang

Secrets Manager, lihat [Apa itu Secrets Manager?](#) dalam AWS Secrets Manager User Guide. Untuk informasi selengkapnya tentang AWS Systems Manager Parameter Store, lihat [AWS Systems Manager Parameter Menyimpan](#) di Panduan AWS Systems Manager Pengguna.

Untuk memperluas pengaturan untuk opsi Pembuatan versi atau Parameter input, Anda dapat memilih panah di sebelah nama pengaturan. Untuk memperluas semua pengaturan untuk semua komponen yang dipilih, Anda dapat mengaktifkan tombol Expand all off and on.

Repositori target

- Nama repositori target - Repositori Amazon ECR tempat image keluaran Anda disimpan jika tidak ada repositori lain yang ditentukan dalam konfigurasi distribusi pipeline untuk Wilayah tempat pipeline berjalan (Wilayah 1).

Untuk membuat versi resep wadah baru:

1. Di bagian atas halaman detail resep kontainer, pilih Buat versi baru. Anda dibawa ke halaman Buat resep untuk resep wadah.
2. Untuk membuat versi baru, buat perubahan, lalu pilih Buat resep.

Untuk informasi selengkapnya tentang cara membuat resep kontainer saat Anda membuat pipeline gambar, lihat [Langkah 2: Pilih resep](#) di bagian Memulai panduan ini.

Buat resep wadah dengan AWS CLI

Untuk membuat resep kontainer Image Builder dengan `imagebuilder create-container-recipe` perintah di AWS CLI, ikuti langkah-langkah berikut:

Prasyarat

Sebelum Anda menjalankan perintah Image Builder di bagian ini untuk membuat resep kontainer dengan AWS CLI, Anda harus membuat komponen yang akan digunakan resep. Contoh resep wadah pada langkah berikut mengacu pada contoh komponen yang dibuat di [Buat komponen dengan AWS CLI](#) bagian panduan ini.

Setelah Anda membuat komponen Anda, atau jika Anda menggunakan komponen yang ada, perhatikan ARN yang ingin Anda sertakan dalam resep.

1. Buat file JSON input CLI

Anda dapat memberikan semua input untuk `create-container-recipe` perintah dengan parameter perintah inline. Namun, perintah yang dihasilkan bisa sangat panjang. Untuk merampingkan perintah, Anda dapat menyediakan file JSON yang berisi semua pengaturan resep wadah

Note

Konvensi penamaan untuk nilai data dalam file JSON mengikuti pola yang ditentukan untuk parameter permintaan tindakan Image Builder API. Untuk meninjau parameter permintaan perintah API, lihat [CreateContainerRecipe](#) perintah di Referensi API EC2 Image Builder.

Untuk memberikan nilai data sebagai parameter baris perintah, lihat nama parameter yang ditentukan dalam Referensi AWS CLI Perintah.

Berikut adalah ringkasan parameter dalam contoh ini:

- `komponen` (array objek, diperlukan) - Berisi array `ComponentConfiguration` objek. Setidaknya satu komponen build harus ditentukan:

Note

Image Builder menginstal komponen dalam urutan yang Anda tentukan dalam resep. Namun, komponen pengerasan CIS selalu berjalan terakhir untuk memastikan bahwa tes benchmark berjalan terhadap gambar keluaran Anda.

- `ComponentARN` (string, diperlukan) — Komponen ARN.

Tip

Untuk menggunakan contoh untuk membuat resep wadah Anda sendiri, ganti contoh ARN dengan ARN untuk komponen yang Anda gunakan untuk resep Anda. Ini termasuk Wilayah AWS, nama, dan nomor versi untuk masing-masing.

- `parameter` (array objek) — Berisi array `ComponentParameter` objek.

⚠ Important

Parameter komponen adalah nilai teks biasa, dan masuk AWS CloudTrail. Kami menyarankan Anda menggunakan AWS Secrets Manager atau AWS Systems Manager Parameter Store untuk menyimpan rahasia Anda. Untuk informasi selengkapnya tentang Secrets Manager, lihat [Apa itu Secrets Manager?](#) dalam AWS Secrets Manager User Guide. Untuk informasi selengkapnya tentang AWS Systems Manager Parameter Store, lihat [AWS Systems Manager Parameter Menyimpan](#) di Panduan AWS Systems Manager Pengguna.

- `name` (string, required) — Nama parameter komponen yang akan ditetapkan.
- `value` (array string, required) — Berisi array string untuk mengatur nilai untuk parameter komponen bernama. Jika ada nilai default yang ditentukan untuk komponen, dan tidak ada nilai lain yang disediakan, AWSTOE gunakan nilai default.
- `ContainerType` (string, required) - Jenis wadah untuk membuat. Nilai yang valid termasuk DOCKER.
- `dockerfileTemplateData`(string) - Template Dockerfile yang digunakan untuk membangun gambar Anda, dinyatakan sebagai gumpalan data sebaris.
- `nama` (string, wajib) — Nama resep wadah.
- `deskripsi` (string) — Deskripsi resep wadah.
- `ParentImage` (string, required) - Gambar yang digunakan resep kontainer sebagai dasar untuk gambar khusus Anda. Nilainya bisa berupa gambar dasar ARN atau ID AMI.
- `PlatformOverride` (string) - Menentukan platform sistem operasi ketika Anda menggunakan gambar dasar kustom.
- `SemanticVersion` <major>(string, required) - Versi semantik dari resep kontainer yang ditentukan dalam format berikut, dengan nilai numerik di setiap posisi untuk menunjukkan versi tertentu: <minor>. <patch>. Sebuah contoh akan 1.0.0. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. [Versi semantik](#)
- `tag` (peta string) — Tag yang dilampirkan pada resep wadah.
- `InstanceConfiguration` (object) - Sekelompok opsi yang dapat digunakan untuk mengonfigurasi instance untuk membangun dan menguji gambar kontainer.

- `image` (string) — ID AMI untuk digunakan sebagai image dasar untuk container build dan test instance. Jika Anda tidak menentukan nilai ini, Image Builder menggunakan AMI yang dioptimalkan Amazon ECS yang sesuai sebagai gambar dasar.
- `blockDeviceMappings`(array objek) — Mendefinisikan perangkat blok yang akan dilampirkan untuk membangun instance dari Image Builder AMI yang ditentukan dalam image parameter.
 - `DeviceName` (string) - Perangkat yang diterapkan pemetaan ini.
 - `ebs` (object) - Digunakan untuk mengelola konfigurasi khusus Amazon EBS untuk pemetaan ini.
 - `deleteOnTermination`(Boolean) — Digunakan untuk mengkonfigurasi penghapusan pada penghentian perangkat terkait.
 - `dienkripsi` (Boolean) - Digunakan untuk mengkonfigurasi enkripsi perangkat.
 - `VolumeSize` (integer) - Digunakan untuk mengganti ukuran volume perangkat.
 - `VolumeType` (string) - Digunakan untuk mengganti jenis volume perangkat.
- `TargetRepository` (object, required) - Repositori tujuan untuk image container jika tidak ada repositori lain yang ditentukan dalam konfigurasi distribusi pipeline Anda untuk Wilayah tempat pipeline berjalan (Wilayah 1).
 - `RepositoryName` (string, required) - Nama repositori kontainer tempat gambar kontainer output disimpan. Nama ini diawali dengan lokasi repositori.
 - `service` (string, required) - Menentukan layanan di mana gambar ini terdaftar.
- `WorkingDirectory` (string) - Direktori kerja untuk digunakan selama alur kerja build dan pengujian.

```
{
  "components": [
    {
      "componentArn": "arn:aws:imagebuilder:us-east-1:123456789012:component/helloworldal2/x.x.x"
    }
  ],
  "containerType": "DOCKER",
  "description": "My Linux Docker container image",
  "dockerfileTemplateData": "FROM
  {{{ imagebuilder:parentImage }}}\n{{{ imagebuilder:environments }}}\n{{{ imagebuilder:comp
  "name": "amazonlinux-container-recipe",
```

```
"parentImage": "amazonlinux:latest",
"platformOverride": "Linux",
"semanticVersion": "1.0.2",
"tags": {
  "sometag" : "Tag detail"
},
"instanceConfiguration": {
  "image": "ami-1234567890",
  "blockDeviceMappings": [
    {
      "deviceName": "/dev/xvda",
      "ebs": {
        "deleteOnTermination": true,
        "encrypted": false,
        "volumeSize": 8,
        "volumeType": "gp2"
      }
    }
  ]
},
"targetRepository": {
  "repositoryName": "myrepo",
  "service": "ECR"
},
"workingDirectory": "/tmp"
}
```

2. Buat resepnya

Gunakan perintah berikut untuk membuat resep. Berikan nama file JSON yang Anda buat pada langkah sebelumnya dalam `--cli-input-json` parameter:

```
aws imagebuilder create-container-recipe --cli-input-json file://create-container-recipe.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan

garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

Pembersihan sumber daya

Untuk menghindari biaya yang tidak terduga, pastikan untuk membersihkan sumber daya dan saluran pipa yang Anda buat dari contoh dalam panduan ini. Untuk informasi selengkapnya tentang menghapus sumber daya di Image Builder, lihat [Hapus sumber EC2 Image Builder](#).

Mengelola gambar EC2 Image Builder

Setelah Anda membuat sumber daya gambar untuk AMI atau gambar kontainer dengan Image Builder, Anda dapat mengelolanya menggunakan konsol Image Builder, melalui Image Builder API, atau dengan imagebuilder perintah diAWS CLI.

Tip

Bila Anda memiliki beberapa sumber daya dari jenis yang sama, penandaan membantu Anda mengidentifikasi sumber daya tertentu berdasarkan tag yang telah Anda tetapkan padanya. Untuk informasi selengkapnya tentang menandai sumber daya Anda menggunakan perintah Image Builder diAWS CLI, lihat [Memberi tanda pada sumber daya](#) bagian panduan ini.

Bagian ini mencakup cara membuat daftar, melihat, dan membuat gambar. Untuk informasi tentang alur kerja gambar dan cara mengelolanya, lihat [Mengelola alur kerja build dan test untuk image EC2 Image Builder](#).

Daftar Isi

- [Buat daftar gambar dan versi build](#)
- [Lihat detail gambar](#)
- [Buat gambar](#)
- [Impor gambar VM](#)
- [Mengelola temuan keamanan untuk image Image Builder](#)
- [Pembersihan sumber daya](#)

Buat daftar gambar dan versi build

Pada halaman Gambar di konsol Image Builder, Anda dapat melihat daftar semua sumber daya gambar Image Builder yang Anda miliki, yang dibagikan dengan Anda, dan yang dapat Anda akses. Hasil daftar mencakup beberapa detail kunci tentang sumber daya tersebut.

Anda juga dapat melihat semua gambar di akun Anda yang memiliki tindakan alur kerja yang tertunda.

Daftar Isi

- [Daftar gambar](#)
- [Daftar gambar menunggu tindakan](#)
- [Daftar versi pembuatan gambar](#)

Daftar gambar

Bagian ini menjelaskan berbagai cara yang dapat Anda daftar informasi tentang gambar Anda.

Anda dapat menggunakan salah satu metode berikut untuk mencantumkan sumber daya gambar Image Builder yang dapat Anda akses. Untuk tindakan API, lihat [ListImages](#) di Referensi API EC2 Image Builder. Untuk permintaan SDK terkait, lihat tautan [Lihat Juga](#) di halaman yang sama.

Daftar Isi

- [Daftar gambar di konsol](#)
- [Daftar gambar dengan AWS CLI perintah](#)

Daftar gambar di konsol

Untuk membuka halaman daftar Gambar di konsol, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Gambar dari panel navigasi.

Halaman Gambar di konsol dibagi menjadi tab, berdasarkan kepemilikan gambar atau tindakan alur kerja yang tertunda. Bagian ini mencakup tiga tab pertama yang menampilkan gambar yang Anda miliki atau dapat diakses.

Tab konsol: Dimiliki oleh saya

Di tab Dimiliki oleh saya, Anda dapat menggunakan filter berikut untuk merampingkan hasil daftar gambar.

- Anda dapat mencari semua atau sebagian nama di bilah pencarian.
- Anda dapat memfilter gambar berdasarkan platform sistem operasi mereka (Windows atau Linux).
- Anda dapat memfilter gambar berdasarkan jenis output yang mereka hasilkan (AMI atau gambar kontainer).
- Anda dapat menggunakan sumber Filter untuk menemukan gambar yang diimpor dari mesin virtual dengan VMIE.

Mengikuti kontrol filter, tab Dimiliki oleh saya menampilkan daftar gambar Image Builder yang Anda buat, dengan rincian berikut untuk sumber daya yang terdaftar:

Nama/Versi

Nama sumber daya gambar Image Builder dimulai dengan nama resep dan versi dari mana mereka dibangun. Pilih tautan untuk melihat semua versi pembuatan gambar terkait.

Tipe

Jenis gambar keluaran yang dibuat Image Builder untuk sumber daya gambar ini (AMI atau gambar kontainer).

Platform

Platform sistem operasi dari versi sumber daya gambar, misalnya, "Windows" atau "Linux".

Sumber gambar

Asal gambar dasar yang digunakan Image Builder untuk membangun sumber daya gambar ini. Ini terutama digunakan untuk memfilter hasil gambar yang diimpor dari mesin virtual (VMIE).

Waktu pembuatan

Tanggal dan waktu ketika Image Builder membuat versi sumber daya gambar saat ini.

ARN

Nama Sumber Daya Amazon (ARN) dari versi sumber daya gambar saat ini.

Tab konsol: Berbagi dengan saya

Di tab Dibagikan dengan saya, Anda dapat menggunakan filter berikut untuk merampingkan hasil daftar gambar.

- Anda dapat mencari semua atau sebagian nama di bilah pencarian.
- Anda dapat memfilter gambar berdasarkan platform sistem operasi mereka (Windows atau Linux).
- Anda dapat memfilter gambar berdasarkan jenis output yang mereka hasilkan (AMI atau gambar kontainer).
- Anda dapat menggunakan sumber Filter untuk menemukan gambar yang diimpor dari mesin virtual dengan VMIE.

Mengikuti kontrol filter, tab Dibagikan dengan saya menampilkan daftar gambar Image Builder yang dibagikan dengan Anda, dengan detail berikut untuk sumber daya yang terdaftar:

Nama gambar

Nama sumber daya gambar yang dibagikan dengan Anda. Untuk menggunakan gambar bersama dalam resep, Anda memilih opsi Pilih gambar terkelola, dan ubah asal Gambar menjadi Gambar yang dibagikan dengan saya.

Tipe

Jenis gambar keluaran yang dibuat Image Builder untuk sumber daya gambar ini (AMI atau gambar kontainer).

Versi

Platform sistem operasi dari versi sumber daya gambar, misalnya, "Windows" atau "Linux".

Sumber gambar

Asal gambar dasar yang digunakan Image Builder untuk membangun sumber daya gambar ini, jika berlaku. Ini terutama digunakan untuk memfilter hasil gambar yang diimpor dari mesin virtual (VMIE).

Platform

Platform sistem operasi dari versi sumber daya gambar, misalnya, "Windows" atau "Linux".

Waktu pembuatan

Tanggal dan waktu ketika Image Builder membuat versi sumber daya gambar yang dibagikan dengan Anda.

Pemilik

Pemilik sumber daya gambar bersama.

ARN

Nama Sumber Daya Amazon (ARN) dari versi sumber daya gambar yang dibagikan dengan Anda.

Tab konsol: Dikelola oleh Amazon

Di tab Dikelola oleh Amazon, Anda dapat menggunakan filter berikut untuk merampingkan hasil daftar gambar.

- Anda dapat mencari semua atau sebagian nama di bilah pencarian.
- Anda dapat memfilter gambar berdasarkan platform sistem operasi mereka (Windows atau Linux).
- Anda dapat memfilter gambar berdasarkan jenis output yang mereka hasilkan (AMI atau gambar kontainer).
- Anda dapat menggunakan sumber Filter untuk menemukan gambar yang diimpor dari mesin virtual dengan VMIE.

Mengikuti kontrol filter, tab Dikelola oleh Amazon menampilkan daftar gambar Image Builder terkelola Amazon yang dapat Anda gunakan sebagai gambar dasar untuk resep Anda. Image Builder menampilkan rincian berikut untuk sumber daya yang terdaftar:

Nama gambar

Nama gambar yang dikelola. Saat Anda membuat resep, default untuk gambar dasar Anda adalah Mulai cepat (dikelola Amazon). Gambar yang tercantum di tab ini mengisi daftar nama Gambar yang terkait dengan platform sistem operasi yang Anda pilih untuk gambar dasar saat Anda membuat resep.

Tipe

Jenis gambar keluaran yang dibuat Image Builder untuk sumber daya gambar ini (AMI atau gambar kontainer).

Versi

Platform sistem operasi dari versi sumber daya gambar, misalnya, "Windows" atau "Linux".

Platform

Platform sistem operasi dari versi sumber daya gambar, misalnya, "Windows" atau "Linux".

Waktu pembuatan

Tanggal dan waktu ketika Image Builder membuat versi sumber daya gambar yang dibagikan dengan Anda.

Pemilik

Amazon memiliki gambar yang dikelola.

ARN

Nama Sumber Daya Amazon (ARN) dari versi sumber daya gambar yang dibagikan dengan Anda.

Daftar gambar dengan AWS CLI perintah

Ketika Anda menjalankan [list-images](#) perintah diAWS CLI, Anda bisa mendapatkan daftar gambar yang Anda miliki atau memiliki akses ke.

Contoh perintah berikut menunjukkan cara menggunakan list-images perintah tanpa filter untuk mencantumkan semua sumber daya gambar Image Builder yang Anda miliki.

Contoh: daftar semua gambar

```
aws imagebuilder list-images
```

Output:

```
{
  "requestId": "1abcd234-e567-8fa9-0123-4567b890cd12",
  "imageVersionList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-name/1.0.0",
      "name": "image-recipe-name",
```

```
"type": "AMI",
"version": "1.0.0",
"platform": "Linux",
"owner": "123456789012",
"dateCreated": "2022-04-28T01:38:23.286Z"
},
{
  "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-win/1.0.1",
  "name": "image-recipe-win",
  "type": "AMI",
  "version": "1.0.1",
  "platform": "Windows",
  "owner": "123456789012",
  "dateCreated": "2022-04-28T01:38:23.286Z"
}
]
}
```

Saat Anda menjalankan `list-images` perintah, Anda dapat menerapkan filter untuk merampingkan hasil, seperti yang ditunjukkan contoh berikut. Untuk informasi selengkapnya tentang cara memfilter hasil, lihat perintah [list-images](#) di Command Reference AWS CLI.

Contoh: filter untuk gambar Linux

```
aws imagebuilder list-images --filters name="platform",values="Linux"
```

Output:

```
{
  "requestId": "1abcd234-e567-8fa9-0123-4567b890cd12",
  "imageVersionList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-name/1.0.0",
      "name": "image-recipe-name",
      "type": "AMI",
      "version": "1.0.0",
      "platform": "Linux",
      "owner": "123456789012",
      "dateCreated": "2022-04-28T01:38:23.286Z"
    }
  ]
}
```

Daftar gambar menunggu tindakan

Saat Anda menggunakan tindakan `WaitForAction` langkah dalam alur kerja gambar Anda, itu menjeda alur kerja hingga Anda mengirimkannya sinyal untuk melanjutkan pemrosesan atau gagal alur kerja. Anda dapat menggunakan tindakan langkah ini jika Anda memiliki proses eksternal yang perlu dijalankan sebelum melanjutkan. Anda kemudian dapat menggunakan `SendWorkflowStepAction` untuk mengirim sinyal ke langkah yang dijeda ke `RESUME` atau `STOP`. Anda juga dapat menghentikan atau melanjutkan alur kerja Anda dari konsol.

Tab berikut menunjukkan cara mendapatkan daftar semua sumber daya gambar di akun Anda dengan langkah-langkah alur kerja yang saat ini dijeda untuk menunggu sinyal dilanjutkan atau dihentikan. Tab mencakup langkah-langkah konsol dan AWS CLI perintah.

Anda juga dapat menggunakan API atau SDK untuk mendapatkan daftar langkah alur kerja yang menunggu tindakan. Untuk tindakan API, lihat [ListWaitingWorkflowSteps](#) di Referensi API EC2 Image Builder. Untuk permintaan SDK terkait, lihat tautan [Lihat Juga](#) di halaman yang sama.

Console

Untuk masuk ke tab Menunggu tindakan di konsol, ikuti langkah-langkah ini:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Gambar dari panel navigasi. Ini membuka halaman daftar Gambar.
3. Pilih tab Menunggu tindakan dari halaman daftar.
4. (opsional) Untuk menghentikan atau melanjutkan langkah, pilih kotak centang di sebelah nama, lalu pilih Berhenti langkah atau Lanjutkan langkah. Anda dapat memilih lebih dari satu kotak centang untuk melakukan tindakan yang sama untuk semua langkah yang dipilih.

Rincian langkah alur kerja yang tertunda

Rincian alur kerja untuk langkah yang tertunda meliputi:

- Nama gambar — Nama sumber daya gambar yang memiliki langkah tertunda. Anda dapat memilih link nama untuk menampilkan halaman detail untuk gambar itu.
- Nama langkah tertunda - Nama langkah alur kerja yang menunggu tindakan.
- Id eksekusi langkah - Secara unik mengidentifikasi instance runtime dari langkah alur kerja. Anda dapat memilih ID tertaut untuk menampilkan detail runtime untuk langkah tersebut.

- Langkah mulai - Timestamp saat instance runtime dari langkah alur kerja dimulai.
- Alur Kerja ARN - Nama Sumber Daya Amazon (ARN) dari alur kerja dengan langkah tertunda.
- Tindakan — Tindakan langkah yang dalam keadaan menunggu.

AWS CLI

Ketika Anda menjalankan [list-waiting-workflow-steps](#) perintah diAWS CLI, Anda akan mendapatkan daftar semua gambar di akun Anda yang memiliki langkah-langkah alur kerja yang menunggu tindakan sebelum menyelesaikan proses pembuatan gambar.

Contoh perintah berikut menunjukkan cara menggunakan `list-waiting-workflow-steps` perintah untuk mencantumkan semua gambar di akun Anda dengan langkah-langkah alur kerja yang menunggu tindakan.

Contoh: daftar gambar di akun Anda dengan langkah-langkah alur kerja menunggu

```
aws imagebuilder list-waiting-workflow-steps
```

Output:

Output untuk contoh ini menunjukkan satu gambar di akun dengan langkah yang menunggu tindakan.

```
{
  "steps": [
    {
      "imageBuildVersionArn": "arn:aws:imagebuilder:us-
west-2:111122223333:image/example-image/1.0.0/8",
      "name": "WaitForAction",
      "workflowExecutionId": "wf-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
      "stepExecutionId": "step-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "workflowBuildVersionArn": "arn:aws:imagebuilder:us-
west-2:111122223333:workflow/test/wait-for-action/1.0.0/1",
      "startTime": "2023-11-21T23:21:23.609Z",
      "action": "WaitForAction"
    }
  ]
}
```

Daftar versi pembuatan gambar

Pada halaman Image build version konsol Image Builder, Anda dapat melihat daftar versi build dan detail tambahan untuk sumber daya gambar yang Anda miliki. Anda juga dapat menggunakan perintah atau tindakan dengan Image Builder API, SDK, atau AWS CLI untuk membuat daftar versi pembuatan gambar.

Anda dapat menggunakan salah satu metode berikut untuk membuat daftar versi pembuatan gambar untuk sumber daya gambar yang Anda miliki. Untuk tindakan API, lihat [ListImageBuildVersions](#) di Referensi API EC2 Image Builder. Untuk permintaan SDK terkait, lihat tautan [Lihat Juga](#) di halaman yang sama.

Console

Detail versi

Detail pada halaman Image build version di konsol Image Builder mencakup hal-hal berikut:

- **Versi** - Versi pembuatan sumber daya gambar. Di konsol Image Builder, versi menautkan ke halaman detail gambar.
- **Jenis** - Jenis output yang didistribusikan Image Builder saat membuat sumber daya gambar ini (AMI atau gambar kontainer).
- **Tanggal dibuat** - Tanggal dan waktu ketika Image Builder membuat versi build gambar.
- **Status gambar** - Status saat ini dari versi pembuatan gambar. Status dapat berhubungan dengan build atau disposisi gambar. Misalnya, selama proses pembuatan, Anda mungkin melihat status `Building` atau `Distributing`. Untuk disposisi gambar, Anda mungkin melihat status `Deprecated` atau `Deleted`.
- **Alasan kegagalan** — Alasan status gambar. Konsol Image Builder hanya menampilkan alasan ketika build gagal (Status gambar sama dengan `Failed`).
- **Temuan keamanan** — Temuan pemindaian gambar gabungan untuk versi pembuatan gambar yang direferensikan.
- **ARN** — Nama Sumber Daya Amazon (ARN) untuk versi referensi sumber daya gambar.
- **Aliran log** — Tautan ke detail aliran log untuk versi pembuatan gambar yang direferensikan.

Daftar versi

Untuk membuat daftar versi build gambar di konsol Image Builder, lakukan langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Gambar dari panel navigasi. Secara default, daftar gambar menunjukkan versi saat ini dari setiap gambar yang Anda miliki.
3. Untuk melihat daftar semua versi untuk gambar, pilih tautan versi saat ini. Tautan membuka halaman Image build version yang mencantumkan semua versi build untuk gambar tertentu.

AWS CLI

Saat menjalankan [list-image-build-versions](#) perintah diAWS CLI, Anda akan mendapatkan daftar lengkap versi build untuk sumber daya gambar yang ditentukan. Anda harus memiliki gambar untuk menjalankan perintah ini.

Contoh perintah berikut menunjukkan cara menggunakan list-image-build-versions perintah untuk mencantumkan semua versi build untuk gambar yang ditentukan.

Contoh: daftar versi build untuk gambar tertentu

```
aws imagebuilder list-image-build-versions --image-version-arn
arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-name/1.0.0
```

Output:

Output untuk contoh ini mencakup dua versi build untuk resep gambar yang ditentukan.

```
{
  "requestId": "12f3e45d-67cb-8901-af23-45ed678c9b01",
  "imageSummaryList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-
name/1.0.0/2",
      "name": "image-recipe-name",
      "type": "AMI",
      "version": "1.0.0/2",
      "platform": "Linux",
      "osVersion": "Amazon Linux 2",
      "state": {
        "status": "AVAILABLE"
      },
      "owner": "123456789012",
      "dateCreated": "2023-03-10T01:04:40.609Z",
      "outputResources": {
```

```

    "amis": [
      {
        "region": "us-west-2",
        "image": "ami-012b3456789012c3d",
        "name": "image-recipe-name 2023-03-10T01-05-12.541Z",
        "description": "First verison of image-recipe-name",
        "accountId": "123456789012"
      }
    ]
  },
  "tags": {}
},
{
  "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/recipe-
name/1.0.0/1",
  "name": "image-recipe-name",
  "type": "AMI",
  "version": "1.0.0/1",
  "platform": "Linux",
  "osVersion": "Amazon Linux 2",
  "state": {
    "status": "AVAILABLE"
  },
  "owner": "123456789012",
  "dateCreated": "2023-03-10T00:07:16.384Z",
  "outputResources": {
    "amis": [
      {
        "region": "us-west-2",
        "image": "ami-0d1e23456789f0a12",
        "name": "image-recipe-name 2023-03-10T00-07-18.146132Z",
        "description": "First verison of image-recipe-name",
        "accountId": "123456789012"
      }
    ]
  },
  "tags": {}
}
]
}

```


Note

Output dari `list-image-build-versions` perintah tidak menyertakan temuan keamanan atau aliran log saat ini.

Lihat detail gambar

Pada halaman detail gambar di konsol Image Builder, Anda dapat melihat detail untuk sumber daya gambar tertentu yang Anda miliki. Anda juga dapat menggunakan perintah atau tindakan dengan Image Builder API, SDK, atau AWS CLI untuk mendapatkan detail gambar.

Untuk informasi selengkapnya tentang sumber daya yang Akun AWS dibagikan orang lain dengan Anda melalui sumber daya AWS Resource Access Manager (AWS RAM), lihat [AWSSumber daya akses yang dibagikan dengan Anda](#) di Panduan AWS RAM Pengguna.

Daftar Isi

- [Melihat detail gambar di konsol Image Builder](#)
- [Dapatkan detail kebijakan gambar \(AWS CLI\)](#)

Melihat detail gambar di konsol Image Builder

Halaman detail gambar di konsol Image Builder menyertakan bagian ringkasan, dengan informasi tambahan dikelompokkan ke dalam tab. Judul halaman adalah nama dan versi build dari resep yang membuat gambar.

Bagian dan tab detail konsol

- [Bagian ringkasan](#)
- [Tab sumber daya keluaran](#)
- [Tab konfigurasi infrastruktur](#)
- [Tab pengaturan distribusi](#)
- [Tab alur kerja](#)
- [Tab temuan keamanan](#)
- [Tab tag](#)

Bagian ringkasan

Bagian ringkasan mencakup lebar halaman dan mencakup detail berikut. Detail ini selalu ditampilkan.

Resep

Nama resep dan versi yang tidak menyertakan versi build. Misalnya, jika versi build adalah `sample-linux-recipe | 1.0.1/2`, resepnya adalah `sample-linux-recipe | 1.0.1`, dan versi build-nya `2`.

Tanggal dibuat

Tanggal dan waktu ketika Image Builder membuat versi build gambar.

Status gambar

Status saat ini dari versi build gambar. Status dapat berhubungan dengan build atau disposisi gambar. Misalnya, selama proses pembuatan, Anda mungkin melihat status `Building` atau `Distributing`. Untuk disposisi gambar, Anda mungkin melihat status `Deprecated` atau `Deleted`.

Alasan kegagalan

Alasan status gambar. Konsol Image Builder hanya menampilkan alasan ketika build gagal (Status gambar sama dengan `Failed`).

Tab sumber daya keluaran

Tab Sumber daya keluaran mencantumkan detail keluaran dan distribusi untuk sumber daya gambar yang saat ini ditampilkan. Informasi yang ditampilkan Image Builder bergantung pada jenis resep yang digunakan pipeline untuk membuat gambar, sebagai berikut.

Resep gambar

- Wilayah - Wilayah distribusi untuk output Amazon Machine Image (AMI) yang ditentukan dalam kolom Gambar.
- Image - ID AMI yang didistribusikan Image Builder ke tujuan. ID ini ditautkan ke halaman Amazon Machine Images (AMI) di konsol Amazon EC2.

Note

Image Builder membuat AMI setelah membuat sumber daya gambar keluaran, dan sebelum mendistribusikan AMI ke tujuan.

- Nama — Nama AMI yang didistribusikan Image Builder ke tujuan.
- Deskripsi — Deskripsi opsional dari resep gambar yang digunakan pipeline untuk membuat sumber daya gambar keluaran.
- Akun — Akun AWS Yang memiliki sumber daya gambar Image Builder yang saat ini ditampilkan.

Resep kontainer

Image Builder menampilkan rincian berikut untuk output yang dibuat dari resep kontainer.

- Wilayah - Wilayah distribusi untuk gambar kontainer yang ditentukan dalam kolom URI Gambar.
- URI Gambar - URI dari image kontainer keluaran yang didistribusikan Image Builder ke repositori ECR di Wilayah tujuan.

Note

Image Builder menampilkan satu baris per tujuan. Gambar output selalu memiliki setidaknya satu entri untuk distribusi ke akun yang membuat gambar. Tujuan tambahan dapat mencakup distribusi di seluruh Wilayah, Akun AWS, atau AWS Organizations. Untuk informasi selengkapnya, lihat [Mengelola pengaturan distribusi EC2 Image Builder](#).

Tab konfigurasi infrastruktur

Tab konfigurasi Infrastruktur menampilkan setelan infrastruktur Amazon EC2 yang digunakan Image Builder untuk membuat dan menguji gambar yang saat ini ditampilkan. Image Builder selalu menampilkan nama sumber daya konfigurasi infrastruktur (nama Konfigurasi) dan Nama Sumber Daya Amazon (ARN). Jika konfigurasi infrastruktur Anda menetapkan nilai, detail infrastruktur tambahan dapat mencakup yang berikut

- Tipe instans
- Sebuah contoh profil

- Infrastruktur jaringan
- Pengaturan grup keamanan
- Lokasi Amazon S3 tempat Image Builder menyimpan log aplikasi
- Sebuah key pair Amazon EC2 untuk pemecahan masalah
- Topik Amazon SNS untuk pemberitahuan acara

Untuk informasi selengkapnya, lihat [Kelola konfigurasi infrastruktur EC2 Image Builder](#).

Tab pengaturan distribusi

Tab Pengaturan distribusi menampilkan pengaturan yang digunakan Image Builder untuk mendistribusikan gambar keluaran Anda. Image Builder selalu menampilkan nama sumber daya konfigurasi distribusi (Nama konfigurasi) dan Nama Sumber Daya Amazon (ARN). Detail distribusi tambahan bergantung pada jenis resep yang digunakan pipeline Image Builder untuk membuat gambar, sebagai berikut:

Resep gambar

Jika sumber daya konfigurasi distribusi Anda menetapkan nilai, detail distribusi tambahan dapat mencakup hal berikut:

- Wilayah — Wilayah distribusi untuk output Amazon Machine Image (AMI).
- Nama AMI Keluaran — Nama AMI yang didistribusikan Image Builder ke tujuan.
- Enkripsi (kunci KMS) - Jika dikonfigurasi, Image Builder AWS KMS key yang digunakan untuk mengenkripsi gambar untuk didistribusikan ke Wilayah target.
- Akun target untuk distribusi - Jika Anda mengonfigurasi distribusi lintas akun, kolom ini menampilkan daftar terpisah koma Akun AWS untuk berbagi gambar keluaran di Wilayah target.
- Prinsipal dengan izin bersama — Daftar kepala AWS sekolah yang dipisahkan koma yang memiliki izin untuk meluncurkan gambar Anda, misalnya, Akun AWS atau grup, atau unit organisasi (OU).
AWS Organizations

Note

Ketika Anda memberikan izin kepada prinsipal lain untuk meluncurkan gambar Anda, Anda masih memiliki gambar tersebut. AWS menagih akun Anda untuk semua instance yang diluncurkan Amazon EC2 dari gambar Anda.

- Akun target untuk konfigurasi peluncuran yang lebih cepat -
- Konfigurasi lisensi terkait — ARN konfigurasi lisensi License Manager untuk dikaitkan dengan AMI di Wilayah tertentu.
- Luncurkan konfigurasi template -
- Mengatur versi default template peluncuran -

Resep kontainer

Distribusi kontainer selalu mencakup rincian berikut:

- Wilayah - Wilayah distribusi untuk gambar kontainer yang ditentukan dalam kolom URI Gambar.
- URI Gambar - URI image kontainer keluaran yang didistribusikan Image Builder ke repositori Amazon ECR di Wilayah tujuan.

Note

Image Builder menampilkan satu baris per tujuan. Gambar output selalu memiliki setidaknya satu entri untuk distribusi ke akun yang membuat gambar. Tujuan tambahan dapat mencakup distribusi di seluruh Wilayah, Akun AWS, atau AWS Organizations. Untuk informasi selengkapnya, lihat [Mengelola pengaturan distribusi EC2 Image Builder](#).

Tab alur kerja

Alur kerja menentukan urutan langkah yang dilakukan Image Builder saat membuat gambar baru. Semua gambar memiliki alur kerja build dan test. Kontainer memiliki alur kerja tambahan untuk distribusi. Tab Workflow menampilkan alur kerja yang berlaku yang dijalankan Image Builder untuk gambar Anda.

Filter jenis alur kerja

Image Builder awalnya menampilkan ringkasan alur kerja build dan langkah alur kerja secara default. Namun, filter Alur Kerja menunjukkan semua alur kerja yang sedang berlangsung atau selesai untuk gambar Anda. Untuk melihat alur kerja yang berbeda, pilih dari daftar, sebagai berikut:

Alur kerja gambar (keluaran AMI)

- `build-image`

- `test-image`

Alur kerja kontainer (keluaran kontainer)

- `build-container`
- `test-container`
- `distribute-container`

Note

Jika alur kerja belum dimulai, alur kerja tidak muncul dalam daftar. Misalnya, jika pembuatan gambar Anda baru saja dimulai, `build-image` adalah satu-satunya jenis alur kerja yang muncul dalam daftar. Ketika alur kerja berikutnya dimulai, `test-image` dalam hal ini, Image Builder menambahkannya ke daftar.

Mengikuti filter Alur Kerja, alur kerja yang dipilih menampilkan ringkasan runtime yang menyertakan detail berikut untuk setiap jenis alur kerja:

Status alur kerja

Status runtime saat ini untuk alur kerja ini. Nilai dapat mencakup yang berikut:

- Tertunda
- Dilewati
- Berjalan
- Selesai
- Failed (Gagal)
- Rollback-in-progress
- Rollback-selesai

ID Eksekusi

Pengidentifikasi unik yang ditetapkan Image Builder untuk melacak sumber daya runtime setiap kali menjalankan alur kerja.

Mulai

Stempel waktu saat instance runtime dari alur kerja ini dimulai.

Akhiri

Stempel waktu saat instance runtime dari alur kerja ini selesai.

Total langkah

Jumlah total langkah dalam alur kerja. Ini harus sama dengan jumlah langkah untuk langkah-langkah yang berhasil, dilewati, dan gagal.

Langkah-langkah berhasil

Hitungan runtime untuk jumlah langkah dalam alur kerja yang berhasil dijalankan.

Langkah-langkahnya gagal

Hitungan runtime untuk jumlah langkah dalam alur kerja yang gagal.

Langkah-langkah dilewati

Hitungan runtime untuk jumlah langkah dalam alur kerja yang dilewati.

Detail dalam daftar berikut melaporkan status saat ini untuk semua langkah dalam instance runtime alur kerja ini. Image Builder menampilkan detail yang sama untuk semua jenis gambar.

Langkah

Angka yang mewakili urutan Image Builder menjalankan langkah-langkah alur kerja.

ID Langkah

Pengenal unik untuk langkah alur kerja, ditetapkan saat runtime.

Status langkah

Status runtime saat ini dari langkah alur kerja yang ditentukan.

Status rollback

Status rollback saat ini jika instance runtime dari alur kerja ini gagal.

Nama langkah

Nama langkah alur kerja yang ditentukan.

Mulai

Stempel waktu saat langkah yang ditentukan untuk instance runtime alur kerja ini dimulai.

Akhiri

Stempel waktu saat langkah yang ditentukan untuk instance runtime alur kerja ini selesai.

Tab temuan keamanan

Jika Anda telah mengaktifkan pemindaian, tab Temuan keamanan akan menampilkan temuan Common Vulnerabilities and Exposures (CVE). Amazon Inspector mengidentifikasi temuan ini pada instance pengujian yang diluncurkan Image Builder untuk membuat gambar baru Anda. Untuk memastikan bahwa Image Builder menangkap temuan untuk gambar Anda, Anda harus mengonfigurasi pemindaian sebagai berikut:

1. Aktifkan pemindaian Amazon Inspector untuk akun Anda. Untuk informasi selengkapnya, lihat [Memulai Amazon Inspector di Panduan Pengguna Amazon Inspector](#).
2. Aktifkan temuan keamanan untuk pipeline yang membuat gambar ini. Saat mengaktifkan temuan keamanan untuk pipeline, Image Builder menyimpan snapshot temuan sebelum menghentikan instance pengujian. Untuk informasi selengkapnya, silakan lihat [Konfigurasi pemindaian keamanan untuk gambar Image Builder di AWS Management Console](#)

Tab Temuan Keamanan mencakup detail berikut untuk setiap kerentanan yang diidentifikasi oleh Amazon Inspector untuk gambar Anda.

Keparahan

Tingkat keparahan temuan CVE. Nilai adalah sebagai berikut:

- Tidak dikerjakan
- Informasi
- Rendah
- Sedang
- Tinggi
- Kritis

Menemukan ID

Pengidentifikasi unik untuk temuan CVE yang dideteksi Amazon Inspector untuk gambar Anda saat memindai instance pengujian. ID ditautkan ke halaman Temuan Keamanan > Berdasarkan kerentanan. Untuk informasi selengkapnya, lihat [Mengelola temuan keamanan untuk image Image Builder di AWS Management Console](#).

Sumber

Sumber informasi kerentanan untuk temuan CVE.

Umur

Jumlah hari sejak penemuan pertama kali diamati untuk gambar Anda.

Skor Inspector

Skor yang diberikan Amazon Inspector untuk temuan CVE.

Tab tag

Tab Tag menampilkan tag apa pun yang telah Anda tetapkan untuk gambar Anda.

Dapatkan detail kebijakan gambar (AWS CLI)

Contoh berikut menunjukkan cara mendapatkan detail kebijakan gambar dengan Nama Sumber Daya Amazon (ARN).

```
aws imagebuilder get-image-policy --image-arn arn:aws:imagebuilder:us-  
west-2:123456789012:image/example-image/2019.12.02
```

Buat gambar

Bagian ini menunjukkan cara membuat image Image Builder dan membatalkan build yang sedang berlangsung.

Daftar Isi

- [Membuat citra](#)
- [Batalkan pembuatan gambar \(AWS CLI\)](#)

Membuat citra

Ada beberapa cara berbeda untuk membuat image Image Builder baru. Misalnya, Anda dapat menggunakan salah satu metode berikut untuk membuat gambar dengan AWS Management Console atau AWS CLI. Anda juga dapat menggunakan tindakan [CreateImageAPI](#). Untuk permintaan SDK terkait, Anda dapat merujuk ke tautan [Lihat Juga](#) untuk perintah tersebut di Referensi API EC2 Image Builder.

AWS Management Console

Untuk membuat gambar baru dari pipeline yang ada, Anda dapat menjalankan pipeline secara manual, sebagai berikut. Anda juga dapat menggunakan wizard pipeline untuk membuat gambar baru dari awal. Lihat [Buat pipeline gambar \(AMI\)](#) atau [Buat pipeline gambar \(Docker\)](#), tergantung pada jenis gambar yang ingin Anda buat.

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Pipeline gambar dari panel navigasi.
3. Pilih kotak centang di sebelah nama Pipeline yang ingin Anda jalankan.
4. Untuk membuat gambar, pilih Run pipeline dari menu Actions. Ini memulai pipa.

Anda juga dapat menentukan jadwal untuk menjalankan pipeline, atau menggunakan Amazon EventBridge untuk menjalankan pipeline berdasarkan aturan yang Anda konfigurasi.

AWS CLI

Sebelum Anda menjalankan [create-image](#) perintah diAWS CLI, Anda harus membuat sumber daya berikut jika belum ada:

Sumber daya yang dibutuhkan

- Resep — Anda harus menentukan persis satu resep untuk gambar Anda, sebagai berikut:

Resep gambar

Tentukan Nama Sumber Daya Amazon (ARN) untuk sumber resep gambar Anda dengan parameter. `--image-recipe-arn`

Resep kontainer

Tentukan ARN untuk sumber daya resep kontainer Anda dengan parameter. `--container-recipe-arn`

- Konfigurasi infrastruktur — Tentukan ARN untuk sumber daya konfigurasi infrastruktur Anda dengan parameter. `--infrastructure-configuration-arn`

Anda juga dapat menentukan salah satu sumber daya berikut yang dibutuhkan gambar Anda:

Sumber daya dan konfigurasi opsional

- Konfigurasi distribusi - Secara default, Image Builder mendistribusikan sumber daya gambar keluaran ke akun Anda di Wilayah tempat Anda menjalankan `create-image` perintah. Untuk menyediakan tujuan atau konfigurasi tambahan untuk distribusi Anda, tentukan ARN untuk sumber daya konfigurasi distribusi Anda dengan parameter. `--distribution-configuration-arn`
- Pemindaian gambar - Untuk mengonfigurasi snapshot untuk temuan Amazon Inspector pada instance pengujian gambar atau kontainer, gunakan `--image-scanning-configuration` parameter. Untuk gambar kontainer, Anda juga menentukan repositori ECR yang digunakan Amazon Inspector untuk pemindaian.
- Tes gambar - Untuk menekan tahap pengujian Image Builder, gunakan `--image-tests-configuration` parameter. Atau, Anda dapat mengatur batas waktu untuk berapa lama waktu dapat berjalan.
- Tag gambar - Gunakan `--tags` parameter untuk menambahkan tag ke gambar output Anda.
- Alur kerja gambar - Jika Anda tidak menentukan alur kerja build atau pengujian apa pun, Image Builder akan membuat gambar Anda dengan alur kerja gambar defaultnya. Untuk menentukan alur kerja yang telah Anda buat, gunakan `--workflows` parameter.

Note

Jika Anda menentukan alur kerja gambar, Anda juga harus memberikan nama atau ARN peran IAM yang digunakan Image Builder untuk menjalankan tindakan alur kerja Anda dalam parameter. `--execution-role`

Contoh berikut menunjukkan cara membuat gambar dengan perintah [AWS CLI `create-image`](#). Untuk informasi selengkapnya, lihat Referensi Perintah AWS CLI.

Contoh: Buat gambar dasar dengan distribusi default

```
aws imagebuilder create-image --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/simple-recipe-linux/1.0.0 --infrastructure-configuration-arn arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/simple-infra-config-linux
```

Output:

```
{
  "requestId": "1abcd234-e567-8fa9-0123-4567b890cd12",
  "imageVersionList": [
    {
      "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/simple-recipe-  
linux/1.0.0",
      "name": "simple-recipe-linux",
      ...
    }
  ]
}
```

Batalkan pembuatan gambar (AWS CLI)

Untuk membatalkan build gambar yang sedang berlangsung, gunakan `cancel-image-creation` perintah, sebagai berikut:

```
aws imagebuilder cancel-image-creation --image-build-version-arn  
arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-recipe/2019.12.03/1
```

Impor gambar VM

Image Builder terintegrasi dengan Amazon EC2 VM Import/Export API untuk memungkinkan proses impor berjalan secara asinkron di latar belakang. Image Builder mereferensikan ID tugas dari impor VM untuk melacak kemajuannya, dan membuat sumber daya gambar Image Builder sebagai output. Ini memungkinkan Anda untuk mereferensikan sumber daya gambar Image Builder dalam resep Anda sebelum impor VM selesai.

Impor VM (konsol)

Untuk mengimpor VM dengan konsol Image Builder, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di](https://console.aws.amazon.com/imagebuilder/) <https://console.aws.amazon.com/imagebuilder/>.
2. Pilih Gambar dari panel navigasi.
3. Pilih Impor gambar.
4. Berikan detail untuk bagian berikut di halaman Impor gambar. Kemudian pilih Impor gambar setelah selesai.

Umum

1. Tentukan Nama unik untuk gambar dasar.
2. Tentukan Versi untuk gambar dasar. Gunakan format berikut: *major.minor.patch*.
3. Anda juga dapat memasukkan Deskripsi opsional untuk gambar dasar.

Sistem operasi gambar dasar

1. Pilih opsi Sistem Operasi Gambar (OS) yang cocok dengan platform OS VM Anda.
2. Pilih versi OS yang cocok dengan versi untuk VM Anda dari daftar.

Konfigurasi impor VM

Saat Anda mengekspor VM Anda dari lingkungan virtualisasinya, proses itu membuat satu set atau lebih file wadah disk. Ini bertindak sebagai snapshot dari lingkungan, pengaturan, dan data VM Anda. Anda dapat menggunakan file-file ini untuk mengimpor VM Anda sebagai gambar dasar untuk resep gambar Anda. Untuk informasi selengkapnya tentang mengimpor VM di Image Builder, lihat. [Impor dan ekspor gambar VM](#)

Untuk menentukan lokasi sumber impor Anda, ikuti langkah-langkah berikut:

Impor sumber

Tentukan sumber untuk wadah disk gambar VM pertama atau snapshot yang akan diimpor di bagian Disk container 1.

1. Sumber — Ini bisa berupa bucket S3, atau snapshot EBS.
2. Pilih lokasi disk S3 — Masukkan lokasi di Amazon S3 tempat gambar disk Anda disimpan. Untuk menelusuri lokasi, pilih Browse S3.
3. Untuk menambahkan wadah disk, pilih Tambahkan wadah disk.

IAM role

Untuk mengaitkan peran IAM dengan konfigurasi impor VM Anda, pilih peran dari daftar tarik-turun peran IAM, atau pilih Buat peran baru untuk membuat peran baru. Jika Anda membuat peran baru, halaman konsol Peran IAM akan terbuka di tab terpisah.

Pengaturan lanjutan - opsional

Pengaturan berikut bersifat opsional. Dengan pengaturan ini, Anda dapat mengonfigurasi enkripsi, lisensi, tag, dan lainnya untuk gambar dasar yang dibuat impor.

Arsitektur gambar dasar

Untuk menentukan arsitektur sumber impor VM Anda, pilih nilai dari daftar Arsitektur.

Enkripsi

Jika gambar disk VM Anda dienkripsi, Anda harus memberikan kunci untuk digunakan untuk proses impor. Untuk menentukan kunci KMS untuk impor, pilih nilai dari daftar Enkripsi (kunci KMS). Daftar ini berisi kunci KMS yang dapat diakses akun Anda di Wilayah saat ini.

Manajemen lisensi

Saat Anda mengimpor VM, proses impor secara otomatis mendeteksi OS VM dan menerapkan lisensi yang sesuai ke gambar dasar. Bergantung pada platform OS Anda, jenis lisensinya adalah sebagai berikut:

- Termasuk lisensi - AWS Lisensi yang sesuai untuk platform Anda diterapkan pada gambar dasar Anda.
- Bawa lisensi Anda sendiri (BYOL) - Mempertahankan lisensi dari VM Anda, jika berlaku.

Untuk melampirkan konfigurasi lisensi yang dibuat AWS License Manager dengan gambar dasar Anda, pilih dari daftar nama konfigurasi Lisensi. Untuk informasi selengkapnya tentang License Manager, lihat [Bekerja dengan AWS License Manager](#)

Note

- Konfigurasi lisensi berisi aturan lisensi berdasarkan ketentuan perjanjian perusahaan Anda.
- Linux hanya mendukung lisensi BYOL.

Tag (gambar dasar)

Tag menggunakan pasangan nilai kunci untuk menetapkan teks yang dapat dicari ke sumber daya Image Builder Anda. Untuk menentukan tag untuk gambar dasar yang diimpor, masukkan pasangan kunci-nilai menggunakan kotak Kunci dan Nilai.

Untuk menambahkan tag, pilih Tambahkan tag. Untuk menghapus sebuah tag, pilih Hapus tag.

Impor VM () AWS CLI

Untuk mengimpor VM dari disk ke AMI dan membuat sumber daya gambar Image Builder yang dapat Anda referensikan segera, ikuti langkah-langkah berikut dari: AWS CLI

1. Memulai impor VM, dengan perintah Impor/Ekspor VM import-image Amazon EC2 di file. AWS CLI Catat ID tugas yang dikembalikan dalam respons perintah. Anda akan membutuhkannya untuk langkah selanjutnya. Untuk informasi selengkapnya, lihat [Mengimpor VM sebagai gambar menggunakan Impor/Ekspor VM di Panduan Pengguna Impor/Ekspor VM](#).
2. Buat file JSON input CLI

Untuk merampingkan import-vm-image perintah Image Builder yang digunakan dalam AWS CLI, kita membuat file JSON yang berisi semua konfigurasi impor yang ingin kita masukkan ke dalam perintah.

Note

Konvensi penamaan untuk nilai data dalam file JSON mengikuti pola yang ditentukan untuk parameter permintaan tindakan Image Builder API. Untuk meninjau parameter permintaan perintah API, lihat [ImportVmImage](#) perintah di Referensi API EC2 Image Builder.

Untuk memberikan nilai data sebagai parameter baris perintah, lihat nama parameter yang ditentukan dalam Referensi AWS CLI Perintah. ke import-vm-image perintah Image Builder sebagai opsi.

Berikut adalah ringkasan parameter yang kami tentukan dalam contoh ini:

- name (string, required) - Nama untuk sumber daya gambar Image Builder untuk dibuat sebagai output dari impor.
- SemanticVersion <major>(string, required) - Versi semantik untuk gambar keluaran yang menentukan versi dalam format berikut, dengan nilai numerik di setiap posisi untuk menunjukkan versi tertentu: <minor>. <patch>. Sebagai contoh, 1.0.0. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. [Versi semantik](#)
- deskripsi (string) — Deskripsi resep gambar.

- `platform` (string, diperlukan) — Platform sistem operasi untuk VM yang diimpor.
- `vmImportTaskId` (string, required) - The `ImportTaskId` (AWS CLI) dari proses impor Amazon EC2 VM. Image Builder memantau proses impor untuk menarik AMI yang dibuatnya dan membangun sumber daya gambar Image Builder yang dapat langsung digunakan dalam resep.
- `ClientToken` (string, required) - Pengidentifikasi unik dan peka huruf besar/kecil yang Anda berikan untuk memastikan idempotensi permintaan. Untuk informasi selengkapnya, lihat [Memastikan idempotensi di Referensi](#) API Amazon EC2.
- `tag` (peta string) — Tag adalah pasangan nilai kunci yang dilampirkan ke sumber daya impor. Hingga 50 pasangan nilai kunci diperbolehkan.

Simpan file sebagai `import-vm-image.json`, untuk digunakan dalam `import-vm-image` perintah Image Builder.

```
{
  "name": "example-request",
  "semanticVersion": "1.0.0",
  "description": "vm-import-test",
  "platform": "Linux",
  "vmImportTaskId": "import-ami-01ab234567890cd1e",
  "clientToken": "asz1231231234cs3z",
  "tags": {
    "Usage": "VMIE"
  }
}
```

3. Impor gambar

Jalankan [import-vm-image](#) perintah, dengan file yang Anda buat sebagai input:

```
aws imagebuilder import-vm-image --cli-input-json file://import-vm-image.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan

garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

Mengelola temuan keamanan untuk image Image Builder

Saat Anda mengaktifkan pemindaian keamanan dengan Amazon Inspector, ini terus memindai gambar mesin dan menjalankan instance di akun Anda untuk mencari kerentanan sistem operasi dan bahasa pemrograman. Jika diaktifkan, pemindaian keamanan otomatis, dan Image Builder dapat menyimpan snapshot temuan dari instance pengujian saat Anda membuat gambar baru. Amazon Inspector adalah layanan berbayar.

Saat Amazon Inspector menemukan kerentanan di perangkat lunak atau pengaturan jaringan Anda, diperlukan tindakan berikut:

- Memberitahu Anda bahwa ada temuan.
- Nilai tingkat keparahan temuan. Peringkat keparahan mengkategorikan kerentanan untuk membantu Anda memprioritaskan temuan Anda, dan mencakup nilai-nilai berikut:
 - Tidak digarisbawahi
 - Informasi
 - Rendah
 - Sedang
 - Tinggi
 - Kritis
- Memberikan informasi tentang temuan, dan tautan ke sumber daya tambahan untuk lebih detail.
- Menawarkan panduan remediasi untuk membantu Anda menyelesaikan masalah yang menghasilkan temuan.

Konfigurasi pemindaian keamanan untuk gambar Image Builder di AWS Management Console

Jika Anda telah mengaktifkan Amazon Inspector untuk akun Anda, Amazon Inspector secara otomatis memindai instans EC2 yang diluncurkan Image Builder untuk membuat dan menguji gambar baru. Contoh tersebut memiliki umur yang pendek selama proses pembuatan dan pengujian, dan temuan mereka biasanya akan kedaluwarsa segera setelah instance tersebut ditutup. Untuk

membantu Anda menyelidiki dan memulihkan temuan untuk image baru Anda, Image Builder secara opsional dapat menyimpan temuan apa pun yang diidentifikasi oleh Amazon Inspector pada instance pengujian Anda selama proses pembuatan sebagai snapshot.


Langkah 1: Aktifkan pemindaian keamanan Amazon Inspector untuk akun Anda

Untuk mengaktifkan pemindaian keamanan Amazon Inspector untuk akun Anda dari konsol Image Builder, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Pengaturan pemindaian keamanan dari panel navigasi. Ini membuka kotak dialog Pemindaian keamanan.

Kotak dialog menampilkan status pemindaian untuk akun Anda. Jika Amazon Inspector sudah diaktifkan untuk akun Anda, status akan ditampilkan Diaktifkan.

3. Ikuti langkah 1 dan 2 petunjuk untuk mengaktifkan pemindaian Amazon Inspector.

 Note

Amazon Inspector dikenakan biaya. Untuk informasi selengkapnya, lihat [harga Amazon Inspector](#).

Jika Anda telah mengaktifkan pemindaian untuk pipeline, Image Builder akan mengambil snapshot temuan untuk instance build saat membuat image baru. Dengan cara ini, Anda dapat mengakses temuan setelah Image Builder menghentikan instance build.

Langkah 2: Konfigurasi pipeline Anda untuk menyimpan snapshot untuk temuan kerentanan

Untuk mengonfigurasi snapshot pencarian kerentanan untuk pipeline Anda, lakukan langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Pipeline gambar dari panel navigasi.
3. Pilih salah satu metode berikut untuk menentukan detail pipeline:

Buat pipeline baru

1. Dari halaman Pipelines gambar, pilih Buat pipeline gambar. Ini membuka halaman Tentukan rincian pipeline di wizard pipeline.

Perbarui pipeline yang ada

1. Dari halaman Pipelines gambar, pilih tautan nama Pipeline untuk pipeline yang ingin Anda perbarui. Ini membuka halaman detail pipeline.

Note

Atau, Anda dapat memilih kotak centang di samping nama pipeline yang ingin Anda perbarui, lalu pilih Lihat detail.

2. Dari halaman rincian pipeline, pilih Edit pipeline dari menu Action. Ini akan membawa Anda ke halaman Edit pipeline.
4. Di bagian Umum dari panduan pipeline atau halaman Edit pipeline, pilih kotak centang Aktifkan pemindaian keamanan.

Note

Jika Anda ingin mematikan snapshot nanti, Anda dapat mengedit pipeline untuk menghapus kotak centang. Ini tidak menonaktifkan pemindaian Amazon Inspector untuk akun Anda. Untuk menonaktifkan pemindaian Amazon Inspector, [lihat Menonaktifkan Amazon Inspector](#) di Panduan Pengguna Amazon Inspector.

Mengelola temuan keamanan untuk image Image Builder di AWS Management Console

Halaman daftar temuan Keamanan menampilkan informasi tingkat tinggi tentang temuan untuk sumber daya Anda, dengan tampilan berdasarkan beberapa filter berbeda yang dapat Anda terapkan. Setiap tampilan menyertakan opsi berikut di bagian atas untuk mengubah tampilan Anda:

- Semua temuan keamanan — Ini adalah tampilan default jika Anda memilih halaman Temuan keamanan dari panel navigasi di konsol Image Builder.
- Berdasarkan kerentanan — Tampilan ini menampilkan daftar tingkat tinggi dari semua sumber daya gambar di akun Anda yang memiliki temuan. Finding ID ditautkan ke informasi yang lebih rinci tentang temuan tersebut. Informasi ini muncul di panel yang terbuka di sisi kanan halaman. Panel mencakup informasi berikut:

- Penjelasan rinci tentang temuan ini.
- Tab Menemukan detail. Tab ini mencakup ikhtisar temuan, paket yang terpengaruh, saran perbaikan ringkasan, detail kerentanan, dan kerentanan terkait. ID Kerentanan menautkan ke informasi kerentanan terperinci dalam Database Kerentanan Nasional.
- Tab rincian Skor. Tab ini mencakup side-by-side perbandingan skor CVSS dan Amazon Inspector sehingga Anda dapat melihat di mana Amazon Inspector telah memodifikasi skor, jika berlaku.
- Berdasarkan pipeline gambar - Tampilan ini menunjukkan jumlah temuan untuk setiap pipeline gambar di akun Anda. Image Builder menampilkan jumlah untuk tingkat keparahan sedang dan temuan yang lebih tinggi, ditambah total untuk semua temuan. Semua data dalam daftar ditautkan, sebagai berikut:
 - Kolom nama pipeline Image menautkan ke halaman detail untuk pipeline gambar yang ditentukan.
 - Tautan kolom tingkat keparahan membuka tampilan Semua temuan keamanan, difilter berdasarkan nama pipeline gambar terkait dan tingkat keparahan.

Anda juga dapat menggunakan kriteria pencarian untuk menyempurnakan hasil Anda.

- Berdasarkan gambar - Tampilan ini menunjukkan jumlah temuan untuk setiap pembuatan gambar di akun Anda. Image Builder menampilkan jumlah untuk tingkat keparahan sedang dan temuan yang lebih tinggi, ditambah total untuk semua temuan. Semua data dalam daftar ditautkan, sebagai berikut:
 - Kolom nama gambar menautkan ke halaman detail gambar untuk pembuatan gambar yang ditentukan. Untuk informasi selengkapnya, lihat [Lihat detail gambar](#).
 - Tautan kolom tingkat keparahan membuka tampilan Semua temuan keamanan, difilter berdasarkan nama build gambar terkait dan tingkat keparahan.

Anda juga dapat menggunakan kriteria pencarian untuk menyempurnakan hasil Anda.

Image Builder menampilkan detail berikut di bagian Daftar temuan dari tampilan default Semua temuan keamanan.

Keparahan

Tingkat keparahan temuan CVE. Nilai adalah sebagai berikut:

- Tidak digarisbawahi

- Informasi
- Rendah
- Sedang
- Tinggi
- Kritis

Menemukan ID

Pengidentifikasi unik untuk temuan CVE yang dideteksi Amazon Inspector untuk gambar Anda saat memindai instance build. ID ditautkan ke halaman Temuan Keamanan> Berdasarkan kerentanan.

Gambar ARN

Nama Sumber Daya Amazon (ARN) untuk gambar dengan temuan yang ditentukan dalam kolom Finding ID.

Pipa

Pipeline yang membangun gambar yang ditentukan dalam kolom Gambar ARN.

Deskripsi

Deskripsi singkat tentang temuan tersebut.

Skor Inspector

Skor yang diberikan Amazon Inspector untuk temuan CVE.

Remediasi

Tautan ke detail tentang tindakan yang disarankan untuk memulihkan temuan.

Tanggal dipublikasikan

Tanggal dan waktu ketika kerentanan ini pertama kali ditambahkan ke database vendor.

Pembersihan sumber daya

Untuk menghindari biaya yang tidak terduga, pastikan untuk membersihkan sumber daya dan saluran pipa yang Anda buat dari contoh dalam panduan ini. Untuk informasi selengkapnya tentang menghapus sumber daya di Image Builder, lihat [Hapus sumber EC2 Image Builder](#).

Kelola konfigurasi infrastruktur EC2 Image Builder

Anda dapat menggunakan konfigurasi infrastruktur untuk menentukan infrastruktur Amazon EC2 yang digunakan Image Builder untuk membangun dan menguji image EC2 Image Builder Anda. Pengaturan infrastruktur meliputi:

- Jenis instans untuk membangun dan menguji infrastruktur Anda. Sebaiknya tentukan lebih dari satu jenis instans karena ini memungkinkan Image Builder meluncurkan instance dari pool dengan kapasitas yang memadai. Ini dapat mengurangi kegagalan build sementara Anda.
- Profil instans yang menyediakan izin yang diperlukan untuk melakukan aktivitas penyesuaian pada instance build dan pengujian Anda. Misalnya, jika Anda memiliki komponen yang mengambil sumber daya dari Amazon S3, profil instans memerlukan izin untuk mengakses file tersebut. Profil instans juga memerlukan seperangkat izin minimal agar EC2 Image Builder berhasil berkomunikasi dengan instans. Untuk informasi selengkapnya, lihat [Prasyarat](#).
- Grup VPC, subnet, dan keamanan untuk instance pembuatan dan pengujian pipeline Anda.
- Lokasi Amazon S3 tempat Image Builder menyimpan log aplikasi dari build dan pengujian Anda. Jika Anda mengonfigurasi logging, profil instance yang ditentukan dalam konfigurasi infrastruktur Anda harus memiliki `s3:PutObject` izin untuk bucket target (`arn:aws:s3:::BucketName/*`).
- Key pair Amazon EC2 yang memungkinkan Anda masuk ke instans untuk memecahkan masalah jika build gagal dan Anda menyetelnya. `terminateInstanceOnFailure false`
- Topik SNS tempat Image Builder mengirimkan pemberitahuan acara. Untuk informasi selengkapnya tentang cara Image Builder terintegrasi dengan Amazon SNS, lihat. [Integrasi Amazon SNS di Image Builder](#)

Note

Jika topik SNS Anda dienkripsi, kunci yang mengenkripsi topik ini harus berada di akun tempat layanan Image Builder berjalan. Image Builder tidak dapat mengirim notifikasi ke topik SNS yang dienkripsi dengan kunci dari akun lain.

Anda dapat membuat dan mengelola konfigurasi infrastruktur menggunakan konsol Image Builder, melalui Image Builder API, atau dengan `imagebuilder` perintah diAWS CLI.

Daftar Isi

- [Daftar dan lihat detail konfigurasi infrastruktur](#)

- [Membuat konfigurasi infrastruktur](#)
- [Memperbarui konfigurasi infrastruktur](#)
- [EC2 Image Builder dan antarmuka titik akhir VPC \(\) AWS PrivateLink](#)

Tip

Bila Anda memiliki beberapa sumber daya dari jenis yang sama, penandaan membantu Anda mengidentifikasi sumber daya tertentu berdasarkan tag yang telah Anda tetapkan padanya. Untuk informasi selengkapnya tentang menandai sumber daya Anda menggunakan perintah Image Builder diAWS CLI, lihat [Memberi tanda pada sumber daya](#) bagian panduan ini.

Daftar dan lihat detail konfigurasi infrastruktur

Bagian ini menjelaskan berbagai cara agar Anda dapat menemukan informasi dan melihat detail untuk konfigurasi infrastruktur EC2 Image Builder Anda.

Detail konfigurasi infrastruktur

- [Daftar konfigurasi infrastruktur \(\) AWS CLI](#)
- [Dapatkan detail konfigurasi infrastruktur \(AWS CLI\)](#)

Daftar konfigurasi infrastruktur () AWS CLI

Contoh berikut menunjukkan cara membuat daftar semua konfigurasi infrastruktur Anda, menggunakan [list-infrastructure-configurations](#) perintah diAWS CLI.

```
aws imagebuilder list-infrastructure-configurations
```

Dapatkan detail konfigurasi infrastruktur (AWS CLI)

Contoh berikut menunjukkan bagaimana menggunakan [get-infrastructure-configuration](#) perintah dalam AWS CLI untuk mendapatkan rincian konfigurasi infrastruktur dengan menentukan Amazon Resource Name (ARN).

```
aws imagebuilder get-infrastructure-configuration --infrastructure-configuration-arn
arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/my-example-
infrastructure-configuration
```

Membuat konfigurasi infrastruktur

Bagian ini menjelaskan bagaimana Anda dapat menggunakan konsol Image Builder atau imagebuilder perintah AWS CLI untuk membuat konfigurasi infrastruktur,

Console


Untuk membuat sumber daya konfigurasi infrastruktur dari konsol Image Builder, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Dari panel navigasi, pilih Konfigurasi infrastruktur.
3. Pilih Buat konfigurasi infrastruktur.
4. Di bagian Umum, masukkan informasi yang diperlukan berikut:
 - Masukkan Nama sumber daya konfigurasi infrastruktur Anda.
 - Pilih peran IAM yang ingin Anda kaitkan dengan profil instance untuk izin komponen pada instance build dan pengujian Anda. Image Builder menggunakan izin ini untuk mengunduh dan menjalankan komponen, mengunggah log CloudWatch, dan melakukan tindakan tambahan apa pun yang ditentukan oleh komponen dalam resep Anda.
5. Di panel AWSInfrastruktur, Anda dapat mengonfigurasi semua pengaturan infrastruktur yang tersisa yang tersedia. Masukkan informasi yang diperlukan berikut:
 - Jenis instans - Anda dapat menentukan satu atau beberapa jenis instance yang akan digunakan untuk build ini. Layanan akan memilih salah satu jenis instance ini berdasarkan ketersediaan.

Jika Anda tidak memberikan nilai untuk pengaturan berikut, mereka menggunakan default khusus layanan, jika berlaku.

- VPC, subnet, dan grup keamanan - Image Builder menggunakan VPC dan subnet default Anda. Untuk informasi selengkapnya tentang mengonfigurasi titik akhir antarmuka VPC, lihat [EC2 Image Builder dan antarmuka titik akhir VPC \(\) AWS PrivateLink](#)
- Di bagian Pengaturan Pemecahan Masalah, Anda dapat mengonfigurasi nilai berikut:
 - Secara default, kotak centang Terminate instance on failure dipilih. Namun, jika build gagal, Anda dapat masuk ke instans EC2 untuk memecahkan masalah. Jika Anda ingin instance Anda terus berjalan setelah kegagalan build, kosongkan kotak centang.

- Pasangan kunci — Jika instans EC2 Anda terus berjalan setelah kegagalan build, Anda dapat membuat key pair atau menggunakan key pair yang ada untuk masuk ke instance dan memecahkan masalah.
- Log — Anda dapat menentukan bucket S3 tempat Image Builder dapat menulis log aplikasi untuk membantu memecahkan masalah build dan pengujian Anda. Jika Anda tidak menentukan bucket S3, Image Builder akan menulis log aplikasi ke instance.
- Di bagian Pengaturan metadata Instans, Anda dapat mengonfigurasi nilai berikut untuk diterapkan ke instans EC2 yang digunakan Image Builder untuk membangun dan menguji gambar Anda:
 - Pilih versi Metadata untuk menentukan apakah EC2 memerlukan header token yang ditandatangani misalnya permintaan pengambilan metadata.
 - V1 dan V2 (token opsional) — Nilai default jika Anda tidak memilih apa pun.
 - V2 (token diperlukan)

 Note

Sebaiknya Anda mengonfigurasi semua instans EC2 yang diluncurkan Image Builder dari build pipeline untuk menggunakan IMDSv2 sehingga permintaan pengambilan metadata instance memerlukan header token yang ditandatangani.

- Batas hop respons token metadata — Jumlah hop jaringan yang dapat ditempuh oleh token metadata. Hop minimum: 1, hop maksimum: 64, dengan default satu hop.

AWS CLI

Contoh berikut menunjukkan cara mengkonfigurasi infrastruktur untuk gambar Anda dengan [create-infrastructure-configuration](#) perintah Image Builder diAWS CLI.

1. Buat file JSON input CLI

Contoh konfigurasi infrastruktur ini menentukan dua jenis instance, `m5.large` dan `m5.xlarge`. Sebaiknya tentukan lebih dari satu jenis instans karena ini memungkinkan Image Builder meluncurkan instance dari pool dengan kapasitas yang memadai. Ini dapat mengurangi kegagalan build sementara Anda.

`instanceProfileName` Menentukan profil instance yang menyediakan instance dengan izin yang diperlukan profil untuk melakukan aktivitas penyesuaian. Misalnya, jika Anda memiliki

komponen yang mengambil sumber daya dari Amazon S3, profil instans memerlukan izin untuk mengakses file tersebut. Profil instans juga memerlukan seperangkat izin minimal agar EC2 Image Builder berhasil berkomunikasi dengan instans. Untuk informasi selengkapnya, lihat [Prasyarat](#).

Gunakan alat pengeditan file untuk membuat file JSON dengan kunci yang ditunjukkan dalam contoh berikut, ditambah nilai yang valid untuk lingkungan Anda. Contoh ini menggunakan file bernama `create-infrastructure-configuration.json`:

```
{
  "name": "MyExampleInfrastructure",
  "description": "An example that will retain instances of failed builds",
  "instanceTypes": [
    "m5.large", "m5.xlarge"
  ],
  "instanceProfileName": "myIAMInstanceProfileName",
  "securityGroupIds": [
    "sg-12345678"
  ],
  "subnetId": "sub-12345678",
  "logging": {
    "s3Logs": {
      "s3BucketName": "my-logging-bucket",
      "s3KeyPrefix": "my-path"
    }
  },
  "keyPair": "myKeyPairName",
  "terminateInstanceOnFailure": false,
  "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:MyTopic"
}
```

- Gunakan file yang Anda buat sebagai input saat Anda menjalankan perintah berikut.

```
aws imagebuilder create-infrastructure-configuration --cli-input-json
file://create-infrastructure-configuration.json
```

Memperbarui konfigurasi infrastruktur

Bagian ini mencakup bagaimana Anda dapat menggunakan konsol atau imagebuilder perintah Image Builder AWS CLI untuk memperbarui sumber daya konfigurasi infrastruktur.

Console

Anda dapat mengedit detail konfigurasi infrastruktur berikut dari konsol Image Builder:

- Deskripsi untuk konfigurasi infrastruktur Anda.
- Peran IAM untuk dikaitkan dengan profil instance.
- AWS infrastruktur, termasuk jenis Instance dan topik SNS untuk notifikasi.
- VPC, subnet, dan grup keamanan.
- Pengaturan pemecahan masalah, termasuk Instance Terminate saat gagal, pasangan Kunci untuk menghubungkan, dan lokasi bucket S3 opsional untuk log instance.

Untuk memperbarui sumber daya konfigurasi infrastruktur dari konsol Image Builder, ikuti langkah-langkah berikut:

Pilih konfigurasi infrastruktur Image Builder yang ada

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Untuk melihat daftar sumber daya konfigurasi infrastruktur di bawah akun Anda, pilih Konfigurasi infrastruktur dari panel navigasi.
3. Untuk melihat detail atau mengedit konfigurasi infrastruktur, pilih tautan Nama konfigurasi. Ini membuka tampilan detail untuk konfigurasi infrastruktur.

Note

Anda juga dapat memilih kotak centang di sebelah nama Konfigurasi, lalu pilih Lihat detail.

4. Dari sudut kanan atas panel Detail infrastruktur, pilih Edit.
5. Saat Anda siap untuk menyimpan pembaruan yang telah Anda buat pada konfigurasi infrastruktur, pilih Simpan perubahan.

AWS CLI

Contoh berikut menunjukkan cara memperbarui konfigurasi infrastruktur untuk gambar Anda dengan [update-infrastructure-configuration](#) perintah Image Builder di AWS CLI.

1. Buat file JSON input CLI

Contoh konfigurasi infrastruktur ini menggunakan pengaturan yang sama dengan contoh buat, kecuali bahwa kami telah memperbarui `terminateInstanceOnFailure` pengaturan ke `false`. Setelah kita menjalankan `update-infrastructure-configuration` perintah, pipeline yang menggunakan konfigurasi infrastruktur ini menghentikan instance build dan test saat build gagal.

Gunakan alat pengeditan file untuk membuat file JSON dengan kunci yang ditunjukkan dalam contoh berikut, ditambah nilai yang valid untuk lingkungan Anda. Contoh ini menggunakan file bernama `update-infrastructure-configuration.json`:

```
{
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/my-example-infrastructure-
configuration",
  "description": "An example that will terminate instances of failed builds",
  "instanceTypes": [
    "m5.large", "m5.2xlarge"
  ],
  "instanceProfileName": "myIAMInstanceProfileName",
  "securityGroupIds": [
    "sg-12345678"
  ],
  "subnetId": "sub-12345678",
  "logging": {
    "s3Logs": {
      "s3BucketName": "my-logging-bucket",
      "s3KeyPrefix": "my-path"
    }
  },
  "terminateInstanceOnFailure": true,
  "snsTopicArn": "arn:aws:sns:us-west-2:123456789012:MyTopic"
}
```

2. Gunakan file yang Anda buat sebagai input saat Anda menjalankan perintah berikut.

```
aws imagebuilder update-infrastructure-configuration --cli-input-json
file://update-infrastructure-configuration.json
```

EC2 Image Builder dan antarmuka titik akhir VPC () AWS PrivateLink

Anda dapat membuat koneksi pribadi antara VPC dan EC2 Image Builder dengan membuat antarmuka VPC endpoint. Endpoint antarmuka didukung oleh [AWS PrivateLink](#), teknologi yang memungkinkan Anda mengakses Image Builder API secara pribadi tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi dengan Image Builder API. Lalu lintas antara VPC dan Image Builder Anda tidak meninggalkan jaringan Amazon.

Setiap titik akhir antarmuka diwakili oleh satu atau beberapa [Antarmuka Jaringan Elastis](#) di subnet Anda. Saat Anda membuat gambar baru, Anda dapat menentukan subnet-id VPC dalam konfigurasi infrastruktur Anda.

Note

Setiap layanan yang Anda akses dari dalam VPC memiliki endpoint antarmuka sendiri, dengan kebijakan endpoint sendiri. Image Builder mengunduh aplikasi pengelola AWSTOE komponen dan mengakses sumber daya terkelola dari bucket S3 untuk membuat gambar khusus. Untuk memberikan akses ke bucket tersebut, Anda harus memperbarui kebijakan endpoint S3 untuk mengizinkannya. Untuk informasi selengkapnya, lihat [Kebijakan khusus untuk akses bucket S3](#).

Untuk informasi selengkapnya tentang titik akhir VPC, lihat Titik akhir [VPC Antarmuka \(\) di AWS PrivateLink Panduan Pengguna Amazon VPC](#).

Pertimbangan untuk titik akhir VPC Image Builder

Sebelum menyiapkan titik akhir VPC antarmuka untuk Image Builder, pastikan Anda meninjau [properti dan batasan titik akhir Antarmuka di](#) Panduan Pengguna Amazon VPC.

Image Builder mendukung panggilan ke semua tindakan API-nya dari VPC Anda.

Buat antarmuka VPC endpoint untuk Image Builder

Untuk membuat titik akhir VPC untuk layanan Image Builder, Anda dapat menggunakan konsol Amazon VPC atau (). AWS Command Line Interface AWS CLI Untuk informasi lebih lanjut, lihat [Membuat titik akhir antarmuka](#) di Panduan Pengguna Amazon VPC.

Buat endpoint VPC untuk Image Builder menggunakan nama layanan berikut:

- `com.amazonaws.wilayah.imagebuilder`

Jika Anda mengaktifkan DNS pribadi untuk titik akhir, Anda dapat membuat permintaan API ke Image Builder menggunakan nama DNS default untuk Wilayah, misalnya: `imagebuilder.us-east-1.amazonaws.com` Untuk mencari titik akhir yang berlaku untuk Wilayah target Anda, lihat [titik akhir dan kuota EC2 Image Builder](#) di Referensi Umum Amazon Web

Untuk informasi lebih lanjut, lihat [Mengakses layanan melalui titik akhir antarmuka](#) di Panduan Pengguna Amazon VPC.

Membuat kebijakan titik akhir VPC untuk Image Builder

Anda dapat melampirkan kebijakan titik akhir ke titik akhir VPC yang mengontrol akses ke Image Builder. Kebijakan menentukan informasi berikut ini:

- Prinsip-prinsip yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.
- Sumber daya yang dapat digunakan untuk mengambil tindakan.

Jika Anda menggunakan komponen yang dikelola Amazon dalam resep Anda, titik akhir VPC untuk Image Builder harus mengizinkan akses ke pustaka komponen milik layanan berikut:

```
arn:aws:imagebuilder:region:aws:component/*
```

Important

Jika kebijakan non-default diterapkan ke titik akhir VPC antarmuka untuk EC2 Image Builder, permintaan API tertentu yang gagal, seperti yang `RequestLimitExceeded` gagal, mungkin tidak dicatat atau Amazon. AWS CloudTrail CloudWatch

Untuk informasi lebih lanjut, lihat [Mengendalikan akses ke layanan dengan VPC endpoint](#) di Panduan Pengguna Amazon VPC.

Kebijakan khusus untuk akses bucket S3

Image Builder menggunakan bucket S3 yang tersedia untuk umum untuk menyimpan dan mengakses sumber daya terkelola, seperti komponen. Ini juga mengunduh aplikasi manajemen

AWSTOE komponen dari bucket S3 terpisah. Jika Anda menggunakan titik akhir VPC untuk Amazon S3 di lingkungan Anda, Anda harus memastikan bahwa kebijakan titik akhir VPC S3 memungkinkan Image Builder mengakses bucket S3 berikut. Nama bucket unik per AWS Wilayah (*wilayah*) dan lingkungan aplikasi (*lingkungan*). Image Builder dan AWSTOE mendukung lingkungan aplikasi berikut: prod, preprod, dan beta.

- Bucket manajer AWSTOE komponen:

```
s3://ec2imagebuilder-toe-region-environment
```

Contoh: s3://ec2 imagebuilder-toe-us-west -2-prod/*

- Bucket sumber daya terkelola Image Builder:

```
s3://ec2imagebuilder-managed-resources-region-environment/components
```

Contoh: s3://ec2 imagebuilder-managed-resources-us -west-2-prod/components/*

Contoh kebijakan titik akhir VPC

Bagian ini mencakup contoh kebijakan titik akhir VPC kustom.

Kebijakan titik akhir VPC umum untuk tindakan Image Builder

Contoh kebijakan endpoint berikut untuk Image Builder menolak izin untuk menghapus image dan komponen Image Builder. Kebijakan contoh juga memberikan izin untuk melakukan semua tindakan EC2 Image Builder lainnya.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "imagebuilder:*",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "imagebuilder: DeleteImage"
      ],
      "Effect": "Deny",
```

```

    "Resource": "*",
  },
  {
    "Action": [
      "imagebuilder: DeleteComponent"
    ],
    "Effect": "Deny",
    "Resource": "*",
  }
]
}

```

Batasi akses menurut organisasi, izinkan akses komponen terkelola

Contoh kebijakan endpoint berikut menunjukkan cara membatasi akses ke identitas dan sumber daya milik organisasi Anda dan menyediakan akses ke komponen yang dikelola Amazon. AWSTOE Ganti *wilayah principal-org-id*, dan *resource-org-id* dengan nilai-nilai organisasi Anda.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRequestsByOrgsIdentitiesToOrgsResources",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "principal-org-id",
          "aws:ResourceOrgID": "resource-org-id"
        }
      }
    },
    {
      "Sid": "AllowAccessToEC2ImageBuilderComponents",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "imagebuilder:GetComponent"
      ]
    }
  ]
}

```



```

    ],
    "Resource": [
      "arn:aws:imagebuilder:region:aws:component/*"
    ]
  }
]
}

```

Kebijakan titik akhir VPC untuk akses bucket Amazon S3

Contoh kebijakan titik akhir S3 berikut menunjukkan cara menyediakan akses ke bucket S3 yang digunakan Image Builder untuk membuat gambar kustom. Ganti *wilayah* dan *lingkungan* dengan nilai-nilai organisasi Anda. Tambahkan izin lain yang diperlukan ke kebijakan berdasarkan persyaratan aplikasi Anda.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowImageBuilderAccessToAppAndComponentBuckets",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::ec2imagebuilder-toe-region-environment/*",
        "arn:aws:s3:::ec2imagebuilder-managed-resources-region-environment/components/*"
      ]
    }
  ]
}

```

Mengelola pengaturan distribusi EC2 Image Builder

Setelah membuat setelan distribusi dengan Image Builder, Anda dapat mengelolanya menggunakan konsol Image Builder, Image Builder API, atau imagebuilder perintah diAWS CLI. Dengan pengaturan distribusi, Anda dapat melakukan tindakan berikut:

Distribusi AMI

- Tentukan nama dan deskripsi AMI keluaran Anda.
- Otorisasi pihak lain Akun AWS, organisasi, dan OU untuk meluncurkan AMI dari akun pemilik. Akun pemilik ditagih untuk biaya yang terkait dengan AMI.

Note

Untuk membuat publik AMI, atur akun terotorisasi izin peluncuran ke `all`. Lihat contoh untuk membuat AMI publik di EC2 [ModifyImageAttribute](#).

- Buat salinan AMI keluaran untuk setiap akun target, organisasi, dan OU yang ditentukan di Wilayah tujuan. Akun target, organisasi, dan OU memiliki salinan AMI mereka, dan ditagih untuk biaya terkait. Untuk informasi selengkapnya tentang mendistribusikan AMI ke AWS Organizations dan OU Anda, lihat [Berbagi AMI dengan organisasi atau OU](#).
- Salin AMI ke akun pemilik di akun lain Wilayah AWS.
- Ekspor disk gambar VM ke Amazon Simple Storage Service (Amazon S3). Untuk informasi selengkapnya, lihat [Buat pengaturan distribusi untuk output VM disk \(\) AWS CLI](#).

Distribusi gambar kontainer

- Tentukan repositori ECR tempat Image Builder menyimpan gambar keluaran di Region distribusi.

Anda dapat menggunakan setelan distribusi dengan cara berikut untuk mengirimkan gambar ke Wilayah, akun, AWS Organizations dan unit organisasi (OU) target satu kali, atau dengan setiap build pipeline:

- Untuk secara otomatis mengirimkan gambar yang diperbarui ke Wilayah, akun, Organizations, dan OU tertentu, gunakan setelan distribusi dengan pipeline Image Builder yang berjalan sesuai jadwal.
- Untuk membuat gambar baru dan mengirimkannya ke Wilayah, akun, Organizations, dan OU yang ditentukan, gunakan pengaturan distribusi dengan pipeline Image Builder yang Anda jalankan satu kali dari konsol Image Builder, menggunakan Run pipeline dari menu Actions.
- Untuk membuat image baru dan mengirimkannya ke Regions, account, Organizations, dan OU yang ditentukan, gunakan pengaturan distribusi dengan tindakan API berikut atau perintah Image Builder di AWS CLI:
 - [CreateImage](#) Tindakan di Image Builder API.

- [create-image](#) Perintah diAWS CLI.
- Untuk mengekspor disk gambar mesin virtual (VM) ke bucket S3 di Wilayah target sebagai bagian dari proses pembuatan gambar biasa Anda.

Tip

Bila Anda memiliki beberapa sumber daya dari jenis yang sama, penandaan membantu Anda mengidentifikasi sumber daya tertentu berdasarkan tag yang telah Anda tetapkan padanya. Untuk informasi selengkapnya tentang menandai sumber daya Anda menggunakan perintah Image Builder diAWS CLI, lihat [Memberi tanda pada sumber daya](#) bagian panduan ini.

Topik ini mencakup cara membuat daftar, melihat, dan membuat setelan distribusi.

Daftar Isi

- [Daftar dan lihat detail pengaturan distribusi](#)
- [Membuat dan memperbarui konfigurasi distribusi AMI](#)
- [Membuat dan memperbarui pengaturan distribusi untuk gambar kontainer](#)
- [Siapkan distribusi AMI lintas akun dengan Image Builder](#)
- [Konfigurasi setelan distribusi AMI untuk menggunakan templat peluncuran Amazon EC2](#)

Daftar dan lihat detail pengaturan distribusi

Bagian ini menjelaskan berbagai cara agar Anda dapat menemukan informasi dan melihat detail untuk pengaturan distribusi Image Builder EC2 Anda.

Detail pengaturan distribusi

- [Konfigurasi distribusi daftar \(konsol\)](#)
- [Lihat detail konfigurasi distribusi \(konsol\)](#)
- [Daftar distribusi \(\) AWS CLI](#)
- [Dapatkan detail konfigurasi distribusi \(AWS CLI\)](#)

Konfigurasi distribusi daftar (konsol)

Untuk melihat daftar konfigurasi distribusi yang dibuat di bawah akun Anda di konsol Image Builder, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Pengaturan distribusi dari panel navigasi. Ini menunjukkan daftar konfigurasi distribusi yang dibuat di bawah akun Anda.
3. Untuk melihat detail atau membuat konfigurasi distribusi baru, pilih tautan Nama konfigurasi. Ini membuka tampilan detail untuk pengaturan distribusi.

Note

Anda juga dapat memilih kotak centang di sebelah nama Konfigurasi, lalu pilih Lihat detail.

Lihat detail konfigurasi distribusi (konsol)

Untuk melihat detail konfigurasi distribusi tertentu menggunakan konsol Image Builder, pilih konfigurasi yang akan ditinjau, menggunakan langkah-langkah yang dijelaskan dalam [Konfigurasi distribusi daftar \(konsol\)](#).

Pada halaman detail distribusi, Anda dapat:

- Hapus konfigurasi distribusi. Untuk informasi selengkapnya tentang menghapus sumber daya di Image Builder, lihat [Hapus sumber EC2 Image Builder](#).
- Edit detail distribusi.

Daftar distribusi () AWS CLI

Contoh berikut menunjukkan cara menggunakan [list-distribution-configurations](#) perintah dalam AWS CLI daftar semua distribusi Anda.

```
aws imagebuilder list-distribution-configurations
```

Dapatkan detail konfigurasi distribusi (AWS CLI)

Contoh berikut menunjukkan cara menggunakan [get-distribution-configuration](#) perintah dalam AWS CLI untuk mendapatkan rincian konfigurasi distribusi dengan menentukan Amazon Resource Name (ARN).

```
aws imagebuilder get-distribution-configuration --distribution-configuration-arn
arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-example-
distribution-configuration
```

Membuat dan memperbarui konfigurasi distribusi AMI

Bagian ini mencakup pembuatan dan pembaruan konfigurasi distribusi untuk Image Builder AMI.

Daftar Isi

- [Buat konfigurasi distribusi AMI \(konsol\)](#)
- [Buat pengaturan distribusi untuk AMI keluaran \(AWS CLI\)](#)
- [Perbarui pengaturan distribusi AMI \(konsol\)](#)
- [Buat pengaturan distribusi untuk Windows AMI dengan EC2 Fast Launch diaktifkan \(\) AWS CLI](#)
- [Buat pengaturan distribusi untuk output VM disk \(\) AWS CLI](#)
- [Perbarui pengaturan distribusi AMI \(AWS CLI\)](#)

Buat konfigurasi distribusi AMI (konsol)

Konfigurasi distribusi mencakup nama AMI keluaran, pengaturan Wilayah khusus untuk enkripsi, izin peluncuran, dan, organisasi Akun AWS, dan unit organisasi (OU) yang dapat meluncurkan AMI keluaran, dan konfigurasi lisensi.

Untuk membuat konfigurasi distribusi AMI baru:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Pengaturan distribusi dari panel navigasi. Ini menunjukkan daftar konfigurasi distribusi yang dibuat di bawah akun Anda.
3. Pilih Buat pengaturan distribusi di dekat bagian atas panel Pengaturan distribusi.
4. Di bagian Jenis gambar, pilih jenis keluaran Amazon Machine Image (AMI).
5. Di bagian Umum, masukkan Nama untuk konfigurasi distribusi Anda, dan deskripsi opsional.

6. Di bagian Pengaturan Wilayah, masukkan detail berikut untuk setiap Wilayah tempat Anda mendistribusikan AMI Anda:
 - a. AMI didistribusikan ke Wilayah saat ini (Wilayah 1), secara default. Wilayah 1 adalah sumber distribusi. Beberapa pengaturan untuk Wilayah 1 tidak terbuka untuk diedit. Untuk setiap Wilayah yang Anda tambahkan, Anda dapat memilih Region dari daftar dropdown Region.

Kunci Kms mengidentifikasi AWS KMS key yang digunakan untuk mengenkripsi volume EBS untuk gambar Anda di Wilayah target. Penting untuk dicatat bahwa ini tidak berlaku untuk AMI asli yang dibuat build di bawah akun Anda di Wilayah sumber (Wilayah 1). Enkripsi yang berjalan selama fase distribusi build hanya untuk gambar yang didistribusikan ke akun atau Wilayah lain.

Untuk mengenkripsi volume EBS untuk AMI yang dibuat di Wilayah sumber untuk akun Anda, Anda harus menyetel kunci KMS di pemetaan perangkat blok resep gambar (Penyimpanan (volume) di konsol).

Image Builder menyalin AMI ke akun Target yang Anda tentukan untuk Wilayah.

Prasyarat

Untuk menyalin gambar di seluruh akun, Anda harus membuat `EC2ImageBuilderDistributionCrossAccountRole` peran di semua akun target di Wilayah target, dan melampirkan kebijakan [Ec2ImageBuilderCrossAccountDistributionAccesskebijakan](#) terkelola ke peran tersebut.

Nama Output AMI adalah opsional. Jika Anda memberikan nama, nama AMI keluaran akhir menyertakan stempel waktu yang ditambahkan saat AMI dibuat. Jika Anda tidak menentukan nama, Image Builder akan menambahkan stempel waktu build ke nama resep. Ini memastikan nama AMI unik untuk setiap build.

- i. Dengan berbagi AMI, Anda dapat memberikan akses untuk AWS Prinsipal tertentu untuk meluncurkan instans dari AMI Anda. Jika Anda memperluas bagian berbagi AMI, Anda dapat memasukkan detail berikut:
 - Buka izin — Pilih Pribadi jika Anda ingin menjaga AMI tetap pribadi, dan izinkan akses untuk AWS Prinsipal tertentu untuk meluncurkan instance dari AMI pribadi

Anda. Pilih Publik jika Anda ingin membuat AMI Anda publik. Setiap AWS Principal dapat meluncurkan instance dari AMI publik Anda.

- Prinsipal — Anda dapat memberikan akses untuk jenis AWS Prinsipal berikut untuk meluncurkan instance:
 - AWSakun — Berikan akses ke AWS akun tertentu
 - Unit Organisasi (OU) — Memberikan akses ke OU, dan semua entitas anaknya. Entitas anak termasuk OU dan AWS akun.
 - Organisasi — Berikan akses ke AndaAWS Organizations, dan semua entitas anaknya. Entitas anak termasuk OU dan AWS akun.

Pertama, pilih tipe Principal. Kemudian masukkan ID untuk AWS Principal yang ingin Anda berikan akses di kotak di sebelah kanan daftar drop-down. Anda dapat memasukkan beberapa ID dari berbagai jenis.

- ii. Anda dapat memperluas bagian konfigurasi Lisensi untuk melampirkan konfigurasi lisensi yang dibuat dengan AWS License Manager gambar Image Builder Anda. Konfigurasi lisensi berisi aturan lisensi berdasarkan ketentuan perjanjian perusahaan Anda. Image Builder secara otomatis menyertakan konfigurasi lisensi yang dikaitkan dengan AMI dasar Anda.
- iii. Anda dapat memperluas bagian konfigurasi Template Luncurkan untuk menentukan templat peluncuran EC2 yang akan digunakan untuk meluncurkan instance dari AMI yang Anda buat.

Jika Anda menggunakan template peluncuran EC2, Anda dapat menginstruksikan Image Builder untuk membuat versi baru template peluncuran yang menyertakan ID AMI terbaru setelah build selesai. Untuk memperbarui template peluncuran, konfigurasi pengaturan sebagai berikut:

- Luncurkan nama template - Pilih nama template peluncuran yang ingin Anda perbarui Image Builder.
- Setel versi default - Pilih kotak centang ini untuk memperbarui versi default template peluncuran ke versi baru.

Untuk menambahkan konfigurasi template peluncuran lainnya, pilih Tambahkan konfigurasi templat peluncuran. Anda dapat memiliki hingga lima konfigurasi template peluncuran per Wilayah.

b. Untuk menambahkan setelan distribusi untuk Wilayah lain, pilih Tambah Wilayah.

7. Pilih Buat pengaturan setelah Anda selesai.

Buat pengaturan distribusi untuk AMI keluaran (AWS CLI)

Konfigurasi distribusi memungkinkan Anda menentukan nama dan deskripsi AMI keluaran Anda, mengotorisasi pihak lain Akun AWS untuk meluncurkan AMI, menyalin AMI ke akun lain, dan mereplikasi AMI ke Wilayah lain AWS. Ini juga memungkinkan Anda untuk mengekspor AMI ke Amazon Simple Storage Service (Amazon S3), atau mengkonfigurasi EC2 Fast Launch untuk output AMI Windows. Untuk membuat publik AMI, atur akun terotorisasi izin peluncuran ke `all`. Lihat contoh untuk membuat AMI publik di EC2 [ModifyImageAttribute](#).

Contoh berikut menunjukkan cara menggunakan `create-distribution-configuration` perintah untuk membuat konfigurasi distribusi baru untuk AMI Anda, menggunakan AWS CLI.

1. Buat file JSON input CLI

Gunakan alat pengeditan file untuk membuat file JSON dengan kunci yang ditampilkan di salah satu contoh berikut, dan nilai yang valid untuk lingkungan Anda. Contoh-contoh ini menentukan unit organisasi (OU) mana Akun AWS yang memiliki izin untuk meluncurkan AMI yang Anda distribusikan ke Wilayah tertentu. AWS Organizations Beri nama file `create-ami-distribution-configuration.json`, untuk digunakan pada langkah berikutnya:

Accounts

Contoh ini mendistribusikan AMI ke dua Wilayah, dan menetapkan Akun AWS yang memiliki izin peluncuran di setiap Wilayah.

```
{
  "name": "MyExampleAccountDistribution",
  "description": "Copies AMI to eu-west-1, and specifies accounts that can
launch instances in each Region.",
  "distributions": [
    {
      "region": "us-west-2",
      "amiDistributionConfiguration": {
        "name": "Name {{imagebuilder:buildDate}}",
        "description": "An example image name with parameter
references",
        "amiTags": {
          "KeyName": "Some Value"
        }
      }
    }
  ]
}
```



```

        "launchPermission": {
            "userIds": [
                "987654321012"
            ]
        }
    },
    {
        "region": "eu-west-1",
        "amiDistributionConfiguration": {
            "name": "My {{imagebuilder:buildVersion}} image
{{imagebuilder:buildDate}}",
            "amiTags": {
                "KeyName": "Some value"
            },
            "launchPermission": {
                "userIds": [
                    "100000000001"
                ]
            }
        }
    }
]
}

```

Organizations and OUs

Contoh ini mendistribusikan AMI ke Wilayah sumber, dan menentukan izin peluncuran organisasi dan OU.

```

{
  "name": "MyExampleAWSOrganizationDistribution",
  "description": "Shares AMI with the Organization and OU",
  "distributions": [
    {
      "region": "us-west-2",
      "amiDistributionConfiguration": {
        "name": "Name {{ imagebuilder:buildDate }}",
        "launchPermission": {
          "organizationArns": [
            "arn:aws:organizations::123456789012:organization/o-
myorganization123"
          ]
        }
      }
    }
  ]
}

```

```
        "organizationalUnitArns": [  
            "arn:aws:organizations::123456789012:ou/o-123example/ou-1234-  
myorganizationalunit"  
        ]  
    }  
}  
]  
}
```

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://create-  
ami-distribution-configuration.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

Untuk informasi lebih rinci, lihat [create-distribution-configuration](#) di AWS CLI Command Reference.

Perbarui pengaturan distribusi AMI (konsol)

Anda dapat mengubah setelan distribusi AMI menggunakan konsol Image Builder. Pengaturan distribusi yang diperbarui digunakan untuk semua penerapan pipa otomatis dan manual ke depan. Namun, perubahan yang Anda buat tidak berlaku untuk sumber daya apa pun yang telah didistribusikan Image Builder. Misalnya, jika Anda telah mendistribusikan AMI ke Wilayah yang kemudian Anda hapus dari distribusi, AMI yang sudah didistribusikan tetap berada di Wilayah tersebut hingga Anda menghapusnya secara manual.

Perbarui konfigurasi distribusi AMI

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Pengaturan distribusi dari panel navigasi. Ini menunjukkan daftar konfigurasi distribusi yang dibuat di bawah akun Anda.
3. Untuk melihat detail atau memperbarui konfigurasi distribusi, pilih tautan Nama konfigurasi. Ini membuka tampilan detail untuk pengaturan distribusi.

Note

Anda juga dapat memilih kotak centang di sebelah nama Konfigurasi, lalu pilih Lihat detail.

4. Untuk mengedit konfigurasi distribusi, pilih Edit dari sudut kanan atas bagian Detail distribusi. Beberapa bidang dikunci, seperti Nama konfigurasi distribusi, dan Wilayah default yang ditampilkan sebagai Wilayah 1. Untuk informasi selengkapnya tentang pengaturan konfigurasi distribusi, lihat [Buat konfigurasi distribusi AMI \(konsol\)](#).
5. Setelah selesai, pilih Simpan perubahan.

Buat pengaturan distribusi untuk Windows AMI dengan EC2 Fast Launch diaktifkan () AWS CLI

Contoh berikut menunjukkan cara menggunakan [create-distribution-configuration](#) perintah untuk membuat pengaturan distribusi yang memiliki EC2 Fast Launch dikonfigurasi untuk AMI Anda, menggunakan AWS CLI

1. Buat file JSON input CLI

Gunakan alat pengeditan file untuk membuat file JSON dengan kunci seperti yang ditunjukkan pada contoh berikut, ditambah nilai yang valid untuk lingkungan Anda.

Contoh ini meluncurkan instance untuk semua sumber daya targetnya secara bersamaan, karena jumlah maksimum peluncuran paralel lebih besar daripada jumlah sumber daya target. File ini dinamai `ami-dist-config-win-fast-launch.json` dalam contoh perintah yang ditunjukkan pada langkah berikutnya.

```
{  
  "name": "WinFastLaunchDistribution",
```

```
"description": "An example of Windows AMI EC2 Fast Launch settings in the
distribution configuration.",
"distributions": [
  {
    "region": "us-west-2",
    "amiDistributionConfiguration": {
      "name": "Name {{imagebuilder:buildDate}}",
      "description": "Includes Windows AMI EC2 Fast Launch settings with
cross-account distribution.",
      "amiTags": {
        "KeyName": "Some Value"
      }
    },
    "fastLaunchConfigurations": [{
      "enabled": true,
      "snapshotConfiguration": {
        "targetResourceCount": 5
      },
      "maxParallelLaunches": 6,
      "launchTemplate": {
        "launchTemplateId": "lt-0ab1234c56d789012",
        "launchTemplateVersion": "1"
      },
      "accountId": "123456789012"
    }],
    "launchTemplateConfigurations": [{
      "launchTemplateId": "lt-0ab1234c56d789012",
      "setDefaultVersion": true
    }
  ]
}]
}
```

Note

Anda dapat menentukan `launchTemplateName` alih-alih `launchTemplateId` di `launchTemplate` bagian, tetapi Anda tidak dapat menentukan nama dan Id.

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://ami-  
dist-config-win-fast-launch.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

Untuk informasi lebih rinci, lihat [create-distribution-configuration](#) di AWS CLI Command Reference.

Buat pengaturan distribusi untuk output VM disk () AWS CLI

Contoh berikut menunjukkan cara menggunakan `create-distribution-configuration` perintah untuk membuat pengaturan distribusi yang akan mengekspor disk gambar VM ke Amazon S3 dengan setiap build gambar.

1. Buat file JSON input CLI

Anda dapat merampingkan `create-distribution-configuration` perintah yang Anda gunakan di AWS CLI Untuk melakukan ini, buat file JSON yang berisi semua konfigurasi ekspor yang ingin Anda berikan ke perintah.

Note

Konvensi penamaan untuk nilai data dalam file JSON mengikuti pola yang ditentukan untuk parameter permintaan tindakan Image Builder API. Untuk meninjau parameter permintaan perintah API, lihat [CreateDistributionConfiguration](#) perintah di Referensi API EC2 Image Builder.

Untuk memberikan nilai data sebagai parameter baris perintah, lihat nama parameter yang ditentukan dalam Referensi AWS CLI Perintah. ke `create-distribution-configuration` perintah sebagai opsi.

Berikut adalah ringkasan parameter yang kami tentukan dalam objek `s3ExportConfiguration` JSON untuk contoh ini:

- `RoleName` (string, required) - Nama peran yang memberikan izin Impor/Ekspor VM untuk mengekspor gambar ke bucket S3 Anda.
- `diskImageFormat`(string, wajib) - Ekspor gambar disk yang diperbarui ke salah satu format yang didukung berikut:
 - Virtual Hard Disk (VHD) — Kompatibel dengan produk virtualisasi Citrix Xen dan Microsoft Hyper-V.
 - Disk Mesin Virtual ESX yang dioptimalkan Stream (VMDK) - Kompatibel dengan VMware ESX dan VMware vSphere versi 4, 5, dan 6.
 - Mentah - Format mentah.
- `S3bucket` (string, required) - Bucket S3 untuk menyimpan gambar disk output untuk VM Anda.

Simpan file sebagai `export-vm-disks.json`. Gunakan nama file dalam `create-distribution-configuration` perintah.

```
{
  "name": "example-distribution-configuration-with-vm-export",
  "description": "example",
  "distributions": [
    {
      "region": "us-west-2",
      "amiDistributionConfiguration": {
        "description": "example-with-vm-export"
      },
      "s3ExportConfiguration": {
        "roleName": "vmimport",
        "diskImageFormat": "RAW",
        "s3Bucket": "vm-bucket-export"
      }
    },
  ],
  "clientToken": "abc123def4567ab"
}
```

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://export-vm-disks.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

Untuk informasi lebih rinci, lihat [create-distribution-configuration](#) di AWS CLI Command Reference.

Perbarui pengaturan distribusi AMI (AWS CLI)

Contoh berikut menunjukkan cara menggunakan [update-distribution-configuration](#) perintah untuk memperbarui pengaturan distribusi untuk AMI Anda, menggunakan AWS CLI.

1. Buat file JSON input CLI

Gunakan alat pengeditan file favorit Anda untuk membuat file JSON dengan kunci yang ditunjukkan pada contoh berikut, ditambah nilai yang valid untuk lingkungan Anda. Contoh ini menggunakan file bernama `update-ami-distribution-configuration.json`.

```
{
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/update-ami-distribution-configuration.json",
  "description": "Copies AMI to eu-west-2, and specifies accounts that can launch instances in each Region.",
  "distributions": [
    {
      "region": "us-west-2",
      "amiDistributionConfiguration": {
```

```

        "name": "Name {{imagebuilder:buildDate}}",
        "description": "An example image name with parameter references",
        "launchPermissions": {
            "userIds": [
                "987654321012"
            ]
        }
    },
    {
        "region": "eu-west-2",
        "amiDistributionConfiguration": {
            "name": "My {{imagebuilder:buildVersion}} image
{{imagebuilder:buildDate}}",
            "tags": {
                "KeyName": "Some value"
            },
            "launchPermissions": {
                "userIds": [
                    "100000000001"
                ]
            }
        }
    }
]
}

```

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input.

```
aws imagebuilder update-distribution-configuration --cli-input-json file://update-ami-distribution-configuration.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

Untuk informasi lebih rinci, lihat [update-distribution-configuration](#) di AWS CLI Command Reference. Untuk memperbarui tag untuk sumber daya konfigurasi distribusi Anda, lihat [Memberi tanda pada sumber daya](#) bagian.

Membuat dan memperbarui pengaturan distribusi untuk gambar kontainer

Bagian ini mencakup pembuatan dan pembaruan pengaturan distribusi untuk image kontainer Image Builder.

Daftar Isi

- [Buat pengaturan distribusi untuk gambar kontainer Image Builder \(AWS CLI\)](#)
- [Perbarui pengaturan distribusi untuk gambar kontainer Anda \(AWS CLI\)](#)

Buat pengaturan distribusi untuk gambar kontainer Image Builder (AWS CLI)

Konfigurasi distribusi memungkinkan Anda menentukan nama dan deskripsi gambar kontainer keluaran Anda dan mereplikasi gambar kontainer ke AWS Wilayah lain. Anda juga dapat menerapkan tag terpisah ke sumber daya konfigurasi distribusi dan ke gambar kontainer di setiap Wilayah.

1. Buat file JSON input CLI

Gunakan alat pengeditan file favorit Anda untuk membuat file JSON dengan kunci yang ditunjukkan pada contoh berikut, ditambah nilai yang valid untuk lingkungan Anda. Contoh ini menggunakan file bernama `create-container-distribution-configuration.json`:

```
{
  "name": "distribution-configuration-name",
  "description": "Distributes container image to Amazon ECR repository in two
regions.",
  "distributions": [
    {
      "region": "us-west-2",
      "containerDistributionConfiguration": {
        "description": "My test image.",
        "targetRepository": {
          "service": "ECR",
          "repositoryName": "testrepo"
        }
      }
    }
  ]
}
```

```
        "containerTags": ["west2", "image1"]
    }
},
{
    "region": "us-east-1",
    "containerDistributionConfiguration": {
        "description": "My test image.",
        "targetRepository": {
            "service": "ECR",
            "repositoryName": "testrepo"
        },
        "containerTags": ["east1", "imagedist"]
    }
}
],
"tags": {
    "DistributionConfigurationTestTagKey1":
    "DistributionConfigurationTestTagValue1",
    "DistributionConfigurationTestTagKey2":
    "DistributionConfigurationTestTagValue2"
}
}
```

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input.

```
aws imagebuilder create-distribution-configuration --cli-input-json file://create-container-distribution-configuration.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

Untuk informasi lebih rinci, lihat [create-distribution-configuration](#) di AWS CLI Command Reference.

Perbarui pengaturan distribusi untuk gambar kontainer Anda (AWS CLI)

Contoh berikut menunjukkan cara menggunakan [update-distribution-configuration](#) perintah untuk memperbarui pengaturan distribusi untuk gambar kontainer Anda, menggunakan AWS CLI. Anda juga dapat memperbarui tag untuk gambar kontainer di setiap Wilayah.

1. Buat file JSON input CLI

Gunakan alat pengeditan file favorit Anda untuk membuat file JSON dengan kunci yang ditunjukkan pada contoh berikut, ditambah nilai yang valid untuk lingkungan Anda. Contoh ini menggunakan file bernama `update-container-distribution-configuration.json`:

```
{
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/update-container-distribution-
configuration.json",
  "description": "Distributes container image to Amazon ECR repository in two
regions.",
  "distributions": [
    {
      "region": "us-west-2",
      "containerDistributionConfiguration": {
        "description": "My test image.",
        "targetRepository": {
          "service": "ECR",
          "repositoryName": "testrepo"
        },
        "containerTags": ["west2", "image1"]
      }
    },
    {
      "region": "us-east-2",
      "containerDistributionConfiguration": {
        "description": "My test image.",
        "targetRepository": {
          "service": "ECR",
          "repositoryName": "testrepo"
        },
        "containerTags": ["east2", "imagedist"]
      }
    }
  ]
}
```

```
}
```

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input:

```
aws imagebuilder update-distribution-configuration --cli-input-json file://update-container-distribution-configuration.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

Untuk informasi lebih rinci, lihat [update-distribution-configuration](#) di AWS CLI Command Reference. Untuk memperbarui tag untuk sumber daya konfigurasi distribusi Anda, lihat [Memberi tanda pada sumber daya](#) bagian.

Siapkan distribusi AMI lintas akun dengan Image Builder

Bagian ini menjelaskan bagaimana Anda dapat mengonfigurasi setelan distribusi untuk mengirimkan Image Builder AMI ke akun lain yang Anda tentukan.

Akun tujuan kemudian dapat meluncurkan atau memodifikasi AMI, sesuai kebutuhan.

Note

AWS CLI contoh perintah di bagian ini mengasumsikan bahwa Anda sebelumnya telah membuat resep gambar dan konfigurasi infrastruktur file JSON. Untuk membuat file JSON untuk resep gambar, lihat [Buat resep gambar dengan AWS CLI](#). Untuk membuat file JSON untuk konfigurasi infrastruktur, lihat [Membuat konfigurasi infrastruktur](#).

Prasyarat

Untuk memastikan bahwa akun target berhasil meluncurkan instance dari image Image Builder, Anda harus mengonfigurasi izin yang sesuai untuk semua akun tujuan di semua Wilayah.

Jika Anda mengenkripsi AMI menggunakan AWS Key Management Service (AWS KMS), Anda harus mengonfigurasi AWS KMS key untuk akun Anda yang digunakan untuk mengenkripsi gambar baru.

Saat Image Builder melakukan distribusi lintas akun untuk AMI terenkripsi, gambar di akun sumber didekripsi dan didorong ke Wilayah target, di mana gambar tersebut dienkripsi ulang menggunakan kunci yang ditentukan untuk Wilayah tersebut. Karena Image Builder bertindak atas nama akun target, dan menggunakan peran IAM yang Anda buat di Wilayah tujuan, akun tersebut harus memiliki akses ke kunci di Wilayah sumber dan tujuan.

Kunci enkripsi

Prasyarat berikut diperlukan jika gambar Anda dienkripsi menggunakan AWS KMS Prasyarat IAM tercakup dalam bagian selanjutnya.

Persyaratan akun sumber

- Buat kunci KMS di akun Anda di semua Wilayah tempat Anda membangun dan mendistribusikan AMI Anda. Anda juga dapat menggunakan kunci yang ada.
- Perbarui kebijakan kunci untuk semua kunci tersebut agar akun tujuan dapat menggunakan kunci Anda.

Persyaratan akun tujuan

- Tambahkan kebijakan inline `EC2ImageBuilderDistributionCrossAccountRole` yang memungkinkan peran melakukan tindakan yang diperlukan untuk mendistribusikan AMI terenkripsi. Untuk langkah-langkah konfigurasi IAM, lihat bagian [Kebijakan IAM](#) prasyarat.

Untuk informasi selengkapnya tentang penggunaan akses lintas akun AWS KMS, lihat [Mengizinkan pengguna di akun lain menggunakan kunci KMS di Panduan AWS Key Management Service](#) Pengembang.

Tentukan kunci enkripsi Anda dalam resep gambar, sebagai berikut:

- Jika Anda menggunakan konsol Image Builder, pilih kunci enkripsi Anda dari daftar dropdown Encryption (alias KMS) di bagian Storage (volume) resep Anda.

- Jika Anda menggunakan tindakan `CreateImageRecipe` API, atau `create-image-recipe` perintah di AWS CLI, konfigurasi kunci Anda di `ebs` bagian `blockDeviceMappings` di bawah input JSON Anda.

Cuplikan JSON berikut menunjukkan pengaturan enkripsi untuk resep gambar. Selain menyediakan kunci enkripsi Anda, Anda juga harus menyetel `encrypted` flag ke `true`.

```
{
  ...
  "blockDeviceMappings": [
    {
      "deviceName": "Example root volume",
      "ebs": {
        "deleteOnTermination": true,
        "encrypted": true,
        "iops": 100,
        "kmsKeyId": "image-owner-key-id",
        ...
      },
      ...
    }],
    ...
  }
}
```

Kebijakan IAM

Untuk mengonfigurasi izin distribusi lintas akun di AWS Identity and Access Management (IAM), ikuti langkah-langkah berikut:

1. Untuk menggunakan AMI Image Builder yang didistribusikan di seluruh akun, pemilik akun tujuan harus membuat peran IAM baru di akun mereka yang dipanggil `EC2ImageBuilderDistributionCrossAccountRole`.
2. Mereka harus melampirkan [Ec2ImageBuilderCrossAccountDistributionAccesskebijakan](#) peran untuk memungkinkan distribusi lintas akun. Untuk informasi selengkapnya tentang kebijakan terkelola, lihat [Kebijakan Terkelola dan Kebijakan Inline](#) dalam Panduan Pengguna AWS Identity and Access Management.
3. Verifikasi bahwa ID akun sumber ditambahkan ke kebijakan kepercayaan yang dilampirkan pada peran IAM dari akun tujuan. Untuk informasi selengkapnya tentang kebijakan kepercayaan, lihat [Kebijakan Berbasis Sumber Daya](#) di Panduan Pengguna. AWS Identity and Access Management

4. Jika AMI yang Anda distribusikan dienkripsi, pemilik akun tujuan harus menambahkan kebijakan inline berikut ke akun mereka sehingga mereka dapat menggunakan kunci KMS Anda. `EC2ImageBuilderDistributionCrossAccountRole` `PrincipalBagian` ini berisi nomor akun mereka. Hal ini memungkinkan Image Builder bertindak atas nama mereka saat digunakan AWS KMS untuk mengenkripsi dan mendekripsi AMI dengan kunci yang sesuai untuk setiap Wilayah.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRoleToPerformKMSOperationsOnBehalfOfTheDestinationAccount",
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:ListGrants",
        "kms:RevokeGrant"
      ],
      "Resource": "*"
    }
  ]
}
```

Untuk informasi selengkapnya tentang kebijakan sebaris, lihat [Kebijakan Sebaris](#) di AWS Identity and Access Management Panduan Pengguna.

5. Jika Anda menggunakan `launchTemplateConfigurations` untuk menentukan template peluncuran Amazon EC2, Anda juga harus menambahkan kebijakan berikut ke setiap `EC2ImageBuilderDistributionCrossAccountRole` akun tujuan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateLaunchTemplateVersion",

```

```

        "ec2:ModifyLaunchTemplate"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/CreatedBy": "EC2 Image Builder"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeLaunchTemplates"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CreateTags"
    ],
    "Resource": "arn:aws:ec2:*:*:launch-template/*",
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/CreatedBy": "EC2 Image Builder"
        }
    }
}
]
}

```

Batas untuk distribusi lintas akun

Ada beberapa batasan saat mendistribusikan image Image Builder di seluruh akun:

- Akun tujuan dibatasi hingga 50 salinan AMI bersamaan untuk setiap Wilayah tujuan.
- Jika Anda ingin menyalin AMI virtualisasi paravirtual (PV) ke Wilayah lain, Wilayah tujuan harus mendukung AMI virtualisasi PV. Untuk informasi selengkapnya, lihat [Jenis virtualisasi Linux AMI](#).
- Anda tidak dapat membuat salinan snapshot terenkripsi yang tidak terenkripsi. Jika Anda tidak menentukan kunci yang dikelola pelanggan AWS Key Management Service (AWS KMS) untuk KmsKeyId parameter, Image Builder menggunakan kunci default untuk Amazon Elastic Block

Store (Amazon EBS). Untuk informasi lebih lanjut, lihat [Enkripsi Amazon EBS](#) di Panduan Pengguna Amazon Elastic Compute Cloud.

Untuk informasi selengkapnya, lihat [CreateDistributionConfiguration](#) di Referensi API EC2 Image Builder.

Konfigurasi distribusi lintas akun untuk Image Builder AMI (konsol)

Bagian ini menjelaskan cara membuat dan mengonfigurasi setelan distribusi untuk distribusi lintas akun AMI Image Builder Anda menggunakan AWS Management Console. Mengonfigurasi distribusi lintas akun memerlukan izin IAM tertentu. Anda harus menyelesaikan [Prasyarat](#) bagian ini sebelum melanjutkan.

Untuk membuat setelan distribusi di konsol Image Builder, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Pengaturan distribusi dari panel navigasi. Ini menunjukkan daftar pengaturan distribusi yang dibuat di bawah akun Anda.
3. Di bagian atas halaman Pengaturan distribusi, pilih Buat pengaturan distribusi. Ini akan membawa Anda ke halaman Buat pengaturan distribusi.
4. Di bagian Jenis gambar, pilih Amazon Machine Image (AMI) sebagai tipe Output. Ini adalah pengaturan default.
5. Di bagian Umum, masukkan Nama sumber daya pengaturan distribusi yang ingin Anda buat (wajib).
6. Di bagian Pengaturan wilayah, masukkan ID akun 12 digit yang ingin Anda distribusikan AMI Anda di akun Target untuk Wilayah yang dipilih, dan tekan Enter. Ini memeriksa pemformatan yang benar, dan kemudian menampilkan ID akun yang Anda masukkan di bawah kotak. Ulangi proses ini untuk menambahkan lebih banyak akun.

Untuk menghapus akun yang Anda masukkan, pilih X yang ditampilkan di sebelah kanan ID akun.

Masukkan nama Output AMI untuk setiap Wilayah.

7. Lanjutkan menentukan pengaturan tambahan apa pun yang Anda perlukan, dan pilih Buat pengaturan untuk membuat sumber daya pengaturan distribusi baru Anda.

Konfigurasi distribusi lintas akun untuk Image Builder AMI () AWS CLI

Bagian ini menjelaskan cara mengonfigurasi file pengaturan distribusi dan menggunakan `create-image` perintah di dalam AWS CLI untuk membangun dan mendistribusikan Image Builder AMI di seluruh akun.

Mengkonfigurasi distribusi lintas akun memerlukan izin IAM tertentu. Anda harus menyelesaikan [Prasyarat](#) untuk bagian ini sebelum Anda menjalankan `create-image` perintah.

1. Konfigurasi file pengaturan distribusi

Sebelum Anda menggunakan `create-image` perintah dalam AWS CLI untuk membuat Image Builder AMI yang didistribusikan ke akun lain, Anda harus membuat struktur `DistributionConfiguration` JSON yang menentukan ID akun target dalam pengaturan. `AmiDistributionConfiguration` Anda harus menentukan setidaknya satu `AmiDistributionConfiguration` di Wilayah sumber.

File contoh berikut, bernam `create-distribution-configuration.json`, menunjukkan konfigurasi untuk distribusi gambar lintas akun di Wilayah sumber.

```
{
  "name": "cross-account-distribution-example",
  "description": "Cross Account Distribution Configuration Example",
  "distributions": [
    {
      "amiDistributionConfiguration": {
        "targetAccountIds": ["123456789012", "987654321098"],
        "name": "Name {{ imagebuilder:buildDate }}",
        "description": "ImageCopy Ami Copy Configuration"
      },
      "region": "us-west-2"
    }
  ]
}
```

2. Buat pengaturan distribusi

Untuk membuat sumber daya pengaturan distribusi Image Builder menggunakan [create-distribution-configuration](#) perintah di AWS CLI, berikan parameter berikut dalam perintah:

- Masukkan nama distribusi dalam `--name` parameter.

- Lampirkan file JSON konfigurasi distribusi yang Anda buat di `--cli-input-json` parameter.

```
aws imagebuilder create-distribution-configuration --name my distribution name --cli-input-json file://create-distribution-configuration.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

Anda juga dapat memberikan JSON langsung dalam perintah, menggunakan `--distributions` parameter.

Konfigurasi setelan distribusi AMI untuk menggunakan templat peluncuran Amazon EC2

Untuk membantu memastikan pengalaman peluncuran yang konsisten untuk Image Builder AMI di akun target dan Wilayah, Anda dapat menentukan templat peluncuran Amazon EC2 di setelan distribusi, menggunakan `launchTemplateConfigurations`. Ketika `launchTemplateConfigurations` hadir selama proses distribusi, Image Builder membuat versi baru dari templat peluncuran yang mencakup semua pengaturan asli dari template, dan ID AMI baru dari build. Untuk informasi selengkapnya tentang meluncurkan instans EC2 menggunakan template peluncuran, lihat salah satu tautan berikut, tergantung pada sistem operasi target Anda.

- [Luncurkan instance Linux dari template peluncuran](#)
- [Luncurkan instance Windows dari template peluncuran](#)

Tambahkan template peluncuran Amazon EC2 ke pengaturan distribusi AMI Anda (konsol)

Untuk menyediakan template peluncuran dengan AMI keluaran Anda, ikuti langkah-langkah berikut di konsol:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Pengaturan distribusi dari panel navigasi. Ini menunjukkan daftar pengaturan distribusi yang dibuat di bawah akun Anda.
3. Di bagian atas halaman Pengaturan distribusi, pilih Buat pengaturan distribusi. Ini membuka halaman Buat pengaturan distribusi.
4. Di bagian Jenis gambar, pilih jenis Output Amazon Machine Image (AMI). Ini adalah pengaturan default.
5. Di bagian Umum, masukkan Nama sumber daya pengaturan distribusi yang ingin Anda buat (wajib).
6. Di bagian Pengaturan wilayah, pilih nama templat peluncuran EC2 dari daftar. Jika tidak ada template peluncuran di akun Anda, pilih Buat templat peluncuran baru, yang membuka Template Peluncuran di Dasbor EC2.

Pilih kotak centang Setel versi default untuk memperbarui versi default template peluncuran ke versi baru yang dibuat Image Builder dengan AMI keluaran Anda.

Untuk menambahkan templat peluncuran lain ke Wilayah yang dipilih, pilih Tambahkan konfigurasi templat peluncuran.

Untuk menghapus template peluncuran, pilih Hapus.

7. Lanjutkan menentukan pengaturan tambahan apa pun yang Anda perlukan, dan pilih Buat pengaturan untuk membuat sumber daya pengaturan distribusi baru Anda.

Tambahkan template peluncuran Amazon EC2 ke pengaturan distribusi AMI Anda () AWS CLI

Bagian ini menjelaskan cara mengonfigurasi file pengaturan distribusi dengan template peluncuran, dan menggunakan create-image perintah di AWS CLI untuk membangun dan mendistribusikan Image Builder AMI dan versi baru dari template peluncuran yang menggunakannya.

1. Konfigurasi file pengaturan distribusi

Sebelum Anda dapat membuat Image Builder AMI dengan template peluncuran, menggunakan AWS CLI, Anda harus membuat struktur JSON konfigurasi distribusi yang menentukan pengaturan. `launchTemplateConfigurations` Anda harus menentukan setidaknya satu `launchTemplateConfigurations` entri di Wilayah sumber.

File contoh berikut, bernama `create-distribution-config-launch-template.json`, menunjukkan beberapa skenario yang mungkin untuk konfigurasi template peluncuran di Wilayah sumber.

```
{
  "name": "NewDistributionConfiguration",
  "description": "This is just a test",
  "distributions": [
    {
      "region": "us-west-2",
      "amiDistributionConfiguration": {
        "name": "test-{{imagebuilder:buildDate}}-{{imagebuilder:buildVersion}}",
        "description": "description"
      },
      "launchTemplateConfigurations": [
        {
          "launchTemplateId": "lt-0a1bcde2fgh34567",
          "accountId": "935302948087",
          "setDefaultVersion": true
        },
        {
          "launchTemplateId": "lt-0aaa1bcde2ff3456"
        },
        {
          "launchTemplateId": "lt-12345678901234567",
          "accountId": "123456789012"
        }
      ]
    }
  ],
  "clientToken": "clientToken1"
}
```

2. Buat pengaturan distribusi

Untuk membuat sumber daya pengaturan distribusi Image Builder menggunakan [create-distribution-configuration](#) perintah diAWS CLI, berikan parameter berikut dalam perintah:

- Masukkan nama distribusi dalam `--name` parameter.
- Lampirkan file JSON konfigurasi distribusi yang Anda buat di `--cli-input-json` parameter.

```
aws imagebuilder create-distribution-configuration --name my distribution name --cli-input-json file://create-distribution-config-launch-template.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

Anda juga dapat memberikan JSON langsung dalam perintah, menggunakan `--distributions` parameter.

Mengelola kebijakan siklus hidup untuk gambar EC2 Image Builder

Saat Anda membuat gambar khusus, penting bagi Anda untuk memiliki rencana untuk menghentikan gambar tersebut sebelum menjadi usang. Pipeline Image Builder dapat menerapkan pembaruan dan patch keamanan secara otomatis. Namun, setiap build membuat versi baru gambar dan semua sumber daya terkait yang didistribusikannya. Versi sebelumnya tetap ada di akun Anda sampai Anda menghapusnya secara manual, atau membuat skrip untuk melakukan tugas.

Dengan kebijakan manajemen siklus hidup Image Builder, Anda dapat mengotomatiskan proses menghentikan, menonaktifkan, dan menghapus gambar usang dan sumber daya terkait. Sumber daya terkait dapat menyertakan gambar keluaran yang telah Anda distribusikan ke organisasi lain, akun AWS, organisasi, dan unit organisasi (OU) Wilayah AWS. Anda menentukan aturan tentang

bagaimana dan kapan harus mengambil setiap langkah dalam proses siklus hidup, dan langkah-langkah yang harus disertakan dalam kebijakan Anda.

Manfaat manajemen siklus hidup otomatis

Manfaat keseluruhan dari manajemen siklus hidup otomatis meliputi:

- Menyederhanakan manajemen siklus hidup untuk gambar kustom Anda dengan cara otomatis untuk menghentikan gambar dan sumber daya terkait.
- Membantu mencegah risiko kepatuhan yang berasal dari penggunaan gambar yang sudah ketinggalan zaman untuk meluncurkan instance baru.
- Menjaga inventaris gambar tetap segar dengan menghapus gambar yang sudah ketinggalan zaman.
- Dapat mengurangi biaya penyimpanan dan transfer data dengan secara opsional menghapus sumber daya terkait untuk gambar yang dihapus.

Sadarilah penghematan biaya

Tidak ada biaya untuk menggunakan EC2 Image Builder untuk membuat AMI atau gambar kontainer khusus. Namun, harga standar berlaku untuk layanan lain yang digunakan dalam proses tersebut. Ketika Anda menghapus gambar yang tidak terpakai atau usang dan sumber daya terkait dari AndaAkun AWS, Anda dapat mewujudkan penghematan waktu dan biaya dengan cara berikut:

- Kurangi waktu yang diperlukan untuk menambal gambar yang ada saat Anda tidak juga menambal gambar yang tidak terpakai atau ketinggalan zaman.
- Untuk sumber daya gambar AMI yang Anda hapus, Anda dapat memilih untuk juga menghapus AMI terdistribusi dan snapshot terkaitnya. Pendekatan ini dapat menghemat biaya penyimpanan snapshot.
- Untuk sumber daya gambar kontainer yang Anda hapus, Anda dapat memilih untuk menghapus sumber daya yang mendasarinya. Pendekatan ini dapat menghemat biaya penyimpanan Amazon ECR dan kecepatan transfer data untuk Docker gambar Anda yang disimpan di repositori ECR.

Note

Image Builder tidak dapat mengevaluasi dampak potensial untuk semua kemungkinan dependensi hilir, seperti grup Auto Scaling atau template peluncuran. Anda harus

mempertimbangkan dependensi hilir untuk gambar Anda saat mengonfigurasi tindakan kebijakan.

Daftar Isi

- [Prasyarat manajemen siklus hidup untuk gambar EC2 Image Builder](#)
- [Kebijakan manajemen siklus hidup untuk sumber daya gambar EC2 Image Builder](#)
- [Cara kerja aturan manajemen siklus hidup untuk sumber daya gambar EC2 Image Builder](#)

Prasyarat manajemen siklus hidup untuk gambar EC2 Image Builder

Sebelum dapat menentukan kebijakan dan aturan manajemen siklus hidup EC2 Image Builder untuk sumber daya gambar, Anda harus memenuhi prasyarat berikut.

- Buat peran IAM yang memberikan izin kepada Image Builder untuk menjalankan kebijakan siklus hidup. Untuk membuat peran, lihat [Membuat peran IAM untuk manajemen siklus hidup Image Builder](#).
- Buat peran IAM di akun tujuan untuk sumber daya terkait yang didistribusikan di seluruh akun. Peran tersebut memberikan izin kepada Image Builder untuk melakukan tindakan siklus hidup di akun tujuan untuk sumber daya terkait. Untuk membuat peran, lihat [Membuat peran IAM untuk manajemen siklus hidup lintas akun Image Builder](#).

Note

Prasyarat ini tidak berlaku jika Anda telah memberikan izin peluncuran untuk AMI keluaran. Dengan izin peluncuran, akun yang Anda bagikan memiliki instans yang diluncurkan dari AMI bersama, tetapi semua sumber daya AMI tetap ada di akun Anda.

- Untuk gambar kontainer, Anda harus menambahkan tag berikut ke repositori ECR Anda untuk memberikan akses bagi Image Builder untuk menjalankan tindakan siklus hidup pada gambar kontainer yang disimpan dalam repositori: `LifecycleExecutionAccess: EC2 Image Builder`

Membuat peran IAM untuk manajemen siklus hidup Image Builder

Untuk memberikan izin bagi Image Builder untuk menjalankan kebijakan siklus hidup, Anda harus terlebih dahulu membuat peran IAM yang digunakan untuk melakukan tindakan siklus hidup. Ikuti langkah-langkah ini untuk membuat peran layanan yang memberikan izin.

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Pilih Peran dari panel navigasi.
3. Pilih Buat peran. Ini membuka ke langkah pertama dalam proses Pilih entitas tepercaya untuk membuat peran Anda.
4. Pilih opsi Kebijakan kepercayaan khusus untuk jenis entitas Tepercaya.
5. Salin kebijakan kepercayaan JSON berikut dan tempelkan ke area teks kebijakan kepercayaan khusus, ganti teks sampel. Kebijakan kepercayaan ini memungkinkan Image Builder untuk mengambil peran yang Anda buat untuk menjalankan tindakan siklus hidup.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "sts:AssumeRole"
      ],
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "imagebuilder.amazonaws.com"
        ]
      }
    }
  ]
}
```

6. Pilih kebijakan terkelola berikut dari daftar: EC2ImageBuilderLifecycleExecutionPolicy, lalu pilih Berikutnya. Ini membuka halaman Nama, ulasan, dan buat.

Tip

Filter image untuk merampingkan hasil.

7. Masukkan Nama peran.

8. Setelah meninjau pengaturan, pilih Buat peran.

Membuat peran IAM untuk manajemen siklus hidup lintas akun Image Builder

Untuk memberikan izin bagi Image Builder untuk melakukan tindakan siklus hidup di akun tujuan untuk sumber daya terkait, Anda harus terlebih dahulu membuat peran IAM yang digunakan untuk melakukan tindakan siklus hidup di akun tersebut. Anda harus membuat peran di akun tujuan.

Ikuti langkah-langkah ini untuk membuat peran layanan yang memberikan izin di akun tujuan.

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Pilih Peran dari panel navigasi.
3. Pilih Buat peran. Ini membuka ke langkah pertama dalam proses Pilih entitas tepercaya untuk membuat peran Anda.
4. Pilih opsi Kebijakan kepercayaan khusus untuk jenis entitas Tepercaya.
5. Salin kebijakan kepercayaan JSON berikut dan tempelkan ke area teks kebijakan kepercayaan khusus, ganti teks sampel. Kebijakan kepercayaan ini memungkinkan Image Builder untuk mengambil peran yang Anda buat untuk menjalankan tindakan siklus hidup.

Note

Saat Image Builder menggunakan peran ini di akun tujuan untuk bertindak atas sumber daya terkait yang didistribusikan di seluruh akun, itu bertindak atas nama pemilik akun tujuan. Akun AWS yang Anda konfigurasi sebagai kebijakan kepercayaan adalah akun tempat Image Builder mendistribusikan sumber daya tersebut.
`aws:SourceAccount`


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "imagebuilder.amazonaws.com"
        ]
      },
    },
  ],
}
```

```

        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": "444455556666"
            },
            "StringLike": {
                "aws:SourceArn": "arn:*:imagebuilder:*:*:image/**/*/*"
            }
        }
    ]
}


```

- Pilih kebijakan terkelola berikut dari daftar: `EC2ImageBuilderLifecycleExecutionPolicy`, lalu pilih Berikutnya. Ini membuka halaman Nama, ulasan, dan buat.

 Tip

Filter image untuk merampingkan hasil.

- Masukkan `Ec2ImageBuilderCrossAccountLifecycleAccess` sebagai nama Peran.

 Important

`Ec2ImageBuilderCrossAccountLifecycleAccess` harus menjadi nama peran ini.

- Setelah meninjau pengaturan, pilih Buat peran.

Kebijakan manajemen siklus hidup untuk sumber daya gambar EC2 Image Builder

Dengan kebijakan siklus hidup gambar, Anda dapat menentukan strategi pengelolaan sumber daya untuk menghentikan gambar yang sudah ketinggalan zaman dan sumber daya terkait melalui proses menghentikan, menonaktifkan, dan menghapus gambar usang dan sumber daya terkait. Bagian ini menunjukkan cara membuat daftar kebijakan, melihat detail kebijakan, dan membuat kebijakan baru untuk AMI dan gambar kontainer.

Konten

- [Daftar kebijakan manajemen siklus hidup untuk sumber daya gambar Image Builder](#)

- [Lihat detail kebijakan siklus hidup](#)
- [Buat kebijakan siklus hidup](#)

Daftar kebijakan manajemen siklus hidup untuk sumber daya gambar Image Builder

Anda bisa mendapatkan daftar kebijakan pengelolaan siklus hidup gambar yang menyertakan kolom detail utama pada halaman daftar kebijakan Siklus Hidup di AWS Management Console, atau dengan perintah atau tindakan di Image Builder API, SDK, atau AWS CLI

Anda dapat menggunakan salah satu metode berikut untuk mencantumkan sumber daya kebijakan siklus hidup image Builder Image Builder di Akun AWS Untuk tindakan API, lihat [ListLifecyclePolicies](#) di Referensi API EC2 Image Builder. Untuk permintaan SDK terkait, lihat tautan [Lihat Juga](#) di halaman yang sama.

AWS Management Console

Detail berikut ditampilkan di konsol untuk kebijakan yang ada. Anda dapat memilih kolom apa pun untuk mengubah urutan pengurutan hasil Anda. Daftar kebijakan awalnya diurutkan berdasarkan nama Kebijakan. Nama kolom untuk urutan pengurutan saat ini dicetak tebal.

Jika Anda memiliki lebih dari satu halaman hasil, panah paging di sudut kanan atas panel menjadi aktif. Anda dapat memfilter hasil berdasarkan nama kebijakan, status kebijakan, jenis gambar keluaran, dan ARN sumber gambar dengan bilah pencarian.

- Nama kebijakan — Nama kebijakan.
- Status kebijakan — Apakah kebijakan tersebut aktif atau tidak aktif.
- Jenis - Jenis gambar keluaran yang didistribusikan Image Builder saat Anda membuat versi gambar baru (AMI atau gambar kontainer).
- Tanggal eksekusi terakhir — Terakhir kali kebijakan siklus hidup berjalan.
- Tanggal dibuat — Stempel waktu dari pembuatan kebijakan siklus hidup.
- ARN — Nama Sumber Daya Amazon (ARN) dari sumber daya kebijakan siklus hidup.

Untuk mencantumkan kebijakan siklus hidup di AWS Management Console, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).

2. Pilih Kebijakan Siklus Hidup dari panel navigasi. Ini menunjukkan daftar kebijakan siklus hidup gambar di akun Anda.

Tindakan yang tersedia

Anda juga dapat melakukan tindakan berikut untuk kebijakan siklus hidup Anda dari halaman daftar Kebijakan Siklus Hidup.

Untuk membuat kebijakan siklus hidup gambar baru, pilih Buat kebijakan siklus hidup. Untuk informasi selengkapnya tentang cara membuat kebijakan, lihat [Buat kebijakan siklus hidup](#).

Untuk semua tindakan berikut, Anda harus memilih kebijakan terlebih dahulu. Untuk memilih kebijakan, Anda dapat memilih kotak centang di sebelah Nama kebijakan.

- Untuk menonaktifkan atau mengaktifkan kebijakan, pilih Nonaktifkan kebijakan atau Aktifkan kebijakan dari menu Tindakan.
- Untuk mengubah kebijakan, pilih Edit kebijakan dari menu Tindakan.
- Untuk menghapus kebijakan, pilih Hapus kebijakan dari menu Tindakan.
- Untuk membuat kebijakan baru yang menggunakan kebijakan yang dipilih untuk pengaturan dasar, pilih Kebijakan kloning dari menu Tindakan.

AWS CLI

Contoh perintah berikut menunjukkan cara menggunakan kebijakan siklus hidup gambar AWS CLI untuk daftar tertentu. Wilayah AWS Untuk informasi lebih lanjut tentang parameter dan opsi yang dapat Anda gunakan dengan perintah ini, lihat [list-lifecycle-policies](#) perintah di Referensi AWS CLI Perintah.

Contoh:

```
aws imagebuilder list-lifecycle-policies \  
--region us-west-1
```

Output:

```
{  
  "lifecyclePolicySummaryList": [  
    {
```

```

    "arn": "arn:aws:imagebuilder:us-west-2:111122223333:lifecycle-policy/sample-lifecycle-policy1",
    "name": "sample-lifecycle-policy1",
    "status": "DISABLED",
    "executionRole": "arn:aws:iam::111122223333:role/sample-lifecycle-role",
    "resourceType": "AMI_IMAGE",
    "dateCreated": "2023-11-07T14:57:01.603000-08:00",
    "tags": {}
  },
  {
    "arn": "arn:aws:imagebuilder:us-west-2:111122223333:lifecycle-policy/sample-lifecycle-policy2",
    "name": "sample-lifecycle-policy2",
    "status": "ENABLED",
    "executionRole": "arn:aws:iam::111122223333:role/sample-lifecycle-role",
    "resourceType": "AMI_IMAGE",
    "dateCreated": "2023-09-06T10:43:21.436000-07:00",
    "dateLastRun": "2023-11-13T04:43:46.106000-08:00",
    "tags": {}
  },
  {
    "arn": "arn:aws:imagebuilder:us-west-2:111122223333:lifecycle-policy/sample-lifecycle-policy3",
    "name": "sample-lifecycle-policy3",
    "status": "ENABLED",
    "executionRole": "arn:aws:iam::111122223333:role/sample-lifecycle-role",
    "resourceType": "AMI_IMAGE",
    "dateCreated": "2023-10-19T15:16:40.046000-07:00",
    "dateUpdated": "2023-10-21T20:07:15.958000-07:00",
    "dateLastRun": "2023-11-12T09:27:45.830000-08:00"
  }
}]]}

```

Note

Untuk menggunakan default AndaWilayah AWS, jalankan perintah ini tanpa `--region` parameter.

Lihat detail kebijakan siklus hidup

Halaman detail kebijakan siklus hidup di konsol Image Builder menyertakan bagian ringkasan, dengan informasi tambahan dikelompokkan ke dalam tab. Judul halaman adalah nama kebijakan.

Pada halaman detail kebijakan siklus hidup di konsol Image Builder, Anda dapat melihat detail untuk kebijakan siklus hidup tertentu. Anda juga dapat menggunakan perintah atau tindakan dengan Image Builder API, SDK, atau AWS CLI untuk mendapatkan detail kebijakan.

Daftar Isi

- [Melihat detail kebijakan siklus hidup di konsol Image Builder](#)

Melihat detail kebijakan siklus hidup di konsol Image Builder

Halaman detail gambar di konsol Image Builder menyertakan bagian ringkasan, dengan informasi tambahan dikelompokkan ke dalam tab. Judul halaman adalah nama dan versi build dari resep yang membuat gambar.

Bagian dan tab detail konsol

- [Bagian ringkasan](#)
- [Tab Aturan](#)
- [Tab lingkup](#)
- [RunLog tab](#)

Bagian ringkasan

Bagian ringkasan mencakup lebar halaman dan mencakup detail berikut. Detail ini selalu ditampilkan.

Status kebijakan

Apakah kebijakan tersebut aktif atau tidak aktif.

Tipe

Jenis gambar keluaran yang didistribusikan Image Builder saat Anda membuat versi gambar baru (AMI atau image kontainer).

Tanggal dibuat

Stempel waktu dari pembuatan kebijakan siklus hidup.

Tanggal diubah

Terakhir kali kebijakan siklus hidup diperbarui.

Tanggal berjalan terakhir

Terakhir kali kebijakan siklus hidup berjalan.

Peran IAM

Peran IAM yang digunakan Image Builder untuk melakukan tindakan siklus hidup.

ARN

Nama Sumber Daya Amazon (ARN) dari sumber daya kebijakan siklus hidup.

Deskripsi

Deskripsi untuk kebijakan siklus hidup, jika dimasukkan.

Tab Aturan

Tab Aturan menampilkan aturan siklus hidup yang dikonfigurasi untuk kebijakan yang Anda lihat. Tab ini mencakup detail berikut:

- Nama — Nama aturan. Nama-nama ini bersifat statis, berdasarkan tindakan kebijakan yang dapat Anda konfigurasi.
 - `Deprecation rule`
 - `Disable rule`
 - `Deletion rule`
- Aturan — Deskripsi singkat tentang tindakan yang dikonfigurasi untuk aturan.
- Kondisi aturan - Daftar konfigurasi untuk penanganan sumber daya terkait, pengecualian aturan, dan pengaturan penyimpanan, jika berlaku.

Untuk informasi selengkapnya tentang konfigurasi aturan, lihat [Cara kerja aturan siklus hidup](#).

Tab lingkup

Tab Lingkup menampilkan kriteria pemilihan sumber daya yang dikonfigurasi untuk kebijakan yang Anda lihat. Tab ini mencakup detail berikut:

- Filter: ***jenis filter*** - Jenis filter yang Anda gunakan untuk menentukan ruang lingkup. Jenis filter dapat berupa salah satu dari yang berikut:
 - `recipes`— Resep yang digunakan untuk membuat gambar yang berlaku untuk kebijakan siklus hidup.

- **tags**— Kumpulan tag yang digunakan Image Builder untuk memilih sumber daya gambar yang berlaku untuk kebijakan siklus hidup.
- **Bilah pencarian** - Anda dapat memfilter daftar berdasarkan Nama untuk merampingkan hasil yang ditampilkan di tab.
- **Nama** - Setiap baris berisi nama atau tag yang telah Anda konfigurasi untuk kriteria filter.
- **Versi** - Jika Anda telah mengonfigurasi filter resep, Image Builder menampilkan versi resep.

RunLog tab

Setiap kali Anda menjalankan kebijakan untuk sumber daya yang dikonfigurasi, Image Builder menyimpan detail runtime. Setiap baris dalam tabel mewakili contoh runtime tunggal. Tab ini mencakup detail berikut:

- **Execution ID** — Mengidentifikasi instance runtime kebijakan siklus hidup.
- **Status eksekusi** — Status runtime yang melaporkan jika tindakan kebijakan sedang berjalan, berhasil dijalankan, gagal, atau dibatalkan.
- **Sumber daya terpengaruh** - Menunjukkan apakah instance runtime mengidentifikasi sumber daya gambar apa pun untuk tindakan siklus hidup.
- **Tanggal mulai** - Stempel waktu saat instance runtime dimulai.
- **Tanggal akhir** - Stempel waktu saat instance runtime berakhir.

Buat kebijakan siklus hidup

Saat Anda membuat kebijakan siklus hidup EC2 Image Builder baru, konfigurasi akan bergantung pada jenis gambar kebijakan tersebut. Tindakan API untuk membuat kebijakan siklus hidup untuk sumber daya gambar AMI dan sumber daya gambar kontainer adalah sama () [CreateLifecyclePolicy](#). Namun, konfigurasi untuk sumber daya gambar dan sumber daya terkait berbeda. Bagian ini menunjukkan cara membuat kebijakan manajemen siklus hidup untuk keduanya.

Note

Sebelum Anda membuat kebijakan siklus hidup, pastikan bahwa Anda telah memenuhi semua. [Prasyarat](#)

Membuat kebijakan manajemen siklus hidup untuk sumber daya gambar Image Builder AMI

Anda dapat menggunakan salah satu metode berikut untuk membuat kebijakan siklus hidup gambar AMI dengan atau tanpa AWS Management Console atau AWS CLI. Anda juga dapat menggunakan tindakan [CreateLifecyclePolicy](#) API. Untuk permintaan SDK terkait, Anda dapat merujuk ke tautan [Lihat Juga](#) untuk perintah tersebut di Referensi API EC2 Image Builder.

AWS Management Console

Untuk membuat kebijakan siklus hidup untuk sumber daya gambar AMI di AWS Management Console, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Kebijakan Siklus Hidup dari panel navigasi.
3. Pilih Buat kebijakan siklus hidup.
4. Konfigurasi setelan kebijakan yang dijelaskan dalam prosedur berikut.
5. Untuk membuat kebijakan siklus hidup setelah Anda mengonfigurasi setelan, pilih Buat kebijakan.

Konfigurasi setelan Umum untuk kebijakan Anda.

1. Pilih opsi AMI dari Jenis kebijakan.
2. Masukkan nama Kebijakan.
3. Secara opsional masukkan Deskripsi untuk kebijakan siklus hidup Anda.
4. Secara default, Aktifkan dihidupkan. Setelan default mengaktifkan kebijakan siklus hidup dan langsung menambahkannya ke jadwal. Untuk membuat kebijakan yang awalnya dinonaktifkan, Anda dapat menonaktifkan Aktifkan.
5. Pilih peran IAM yang Anda buat untuk izin kebijakan siklus hidup. Jika Anda belum membuat peran ini, lihat [Prasyarat](#) untuk informasi selengkapnya.

Konfigurasi cakupan Aturan untuk kebijakan Anda.

Bagian ini mengonfigurasi pemilihan sumber daya untuk kebijakan siklus hidup Anda, berdasarkan jenis filter yang Anda gunakan.

1. Jenis filter: Resep — Untuk menerapkan aturan siklus hidup ke sumber daya gambar berdasarkan resep yang membuatnya, pilih hingga 50 versi resep untuk kebijakan tersebut.

2. Jenis filter: Tag — Untuk menerapkan aturan siklus hidup ke sumber daya gambar berdasarkan tag sumber daya, masukkan daftar hingga 50 pasangan nilai kunci agar kebijakan cocok.

Aktifkan satu atau beberapa aturan Siklus Hidup berikut agar diterapkan pada sumber daya yang dipilih kebijakan siklus hidup. Jika sumber daya cocok dengan lebih dari satu aturan siklus hidup saat kebijakan berjalan, Image Builder akan melakukan tindakan aturan dalam urutan berikut: 1) Usang, 2) Nonaktifkan, 3) Hapus.

Menghentikan aturan

Menetapkan status sumber daya gambar Image Builder ke`Deprecated`. Pipeline Image Builder masih berjalan untuk gambar yang tidak digunakan lagi. Anda dapat secara opsional menyetel waktu penghentian untuk AMI terkait tanpa memengaruhi kemampuan Anda untuk meluncurkan instance baru.

- Jumlah unit - Tentukan nilai integer untuk periode waktu yang harus dilewati setelah sumber daya gambar dibuat sebelum ditandai sebagai`Deprecated`.
- Unit — Pilih rentang waktu yang akan digunakan. Kisarannya bisa`Days`,`Weeks`,`Months`, atau`Years`.
- Usang AMI — Pilih kotak centang untuk menandai AMI Amazon EC2 terkait dengan tanggal penghentian. AMI tetap tersedia, dan Anda masih dapat meluncurkan instance baru dari mereka.

Nonaktifkan aturan

Menetapkan status sumber daya gambar Image Builder ke`Disabled`. Ini mencegah pipeline Image Builder berjalan untuk gambar ini. Anda dapat menonaktifkan AMI terkait secara opsional untuk mencegah peluncuran instans baru.

- Jumlah unit - Tentukan nilai integer untuk periode waktu yang harus dilewati setelah sumber daya gambar dibuat sebelum ditandai sebagai`Disabled`.
- Unit — Pilih rentang waktu yang akan digunakan. Kisarannya bisa`Days`,`Weeks`,`Months`, atau`Years`.
- Nonaktifkan AMI - Pilih kotak centang untuk menonaktifkan AMI Amazon EC2 terkait. Anda tidak dapat lagi menggunakan AMI atau meluncurkan instance baru dari mereka.

Hapus aturan

Menghapus sumber daya gambar berdasarkan usia atau hitungan. Anda menentukan ambang batas yang memenuhi kebutuhan Anda. Ketika sumber daya gambar Image Builder melewati ambang batas, itu dihapus. Anda dapat secara opsional membatalkan pendaftaran AMI terkait atau menghapus snapshot untuk AMI tersebut. Anda juga dapat menentukan tag untuk sumber daya yang ingin Anda pertahankan melewati ambang batas.

Saat Anda mengonfigurasi aturan Hapus berdasarkan usia, Image Builder menghapus sumber daya gambar setelah periode waktu yang Anda konfigurasi. Misalnya, hapus sumber daya gambar setelah 6 bulan. Saat Anda mengonfigurasi berdasarkan hitungan, Image Builder mempertahankan jumlah gambar terbaru yang Anda tentukan, atau sedekat mungkin dengan angka tersebut, dan menghapus versi sebelumnya.

- Berdasarkan usia
 - Jumlah unit - Tentukan nilai integer untuk periode waktu yang harus dilewati setelah sumber daya gambar dibuat sebelum dihapus.
 - Unit — Pilih rentang waktu yang akan digunakan. Kisarannya bisa `Days`, `Weeks`, `Months`, atau `Years`.
 - Simpan setidaknya satu gambar per resep — Pilih kotak centang untuk menyimpan sumber daya gambar terbaru yang tersedia untuk setiap versi resep yang dipengaruhi aturan ini.

Dengan hitungan

- Jumlah gambar - Tentukan nilai integer untuk jumlah sumber daya gambar terbaru yang akan disimpan untuk setiap versi resep.
- Deregister AMI - Pilih kotak centang untuk membatalkan pendaftaran AMI Amazon EC2 terkait. Anda tidak dapat lagi menggunakan AMI atau meluncurkan instance baru dari mereka.
- Simpan gambar, AMI, dan snapshot dengan tag terkait - Pilih kotak centang untuk memasukkan daftar tag untuk sumber daya gambar yang ingin Anda simpan. Tag berlaku untuk sumber daya gambar dan AMI Amazon EC2. Anda dapat memasukkan hingga 50 pasangan nilai kunci.

Tanda (opsional)

Tambahkan tag ke kebijakan siklus hidup Anda.

AWS CLI

Untuk membuat kebijakan lifecycle Image Builder baru, Anda dapat menggunakan [create-lifecycle-policy](#) perintah di AWS CLI

Membuat kebijakan manajemen siklus hidup untuk sumber daya gambar kontainer Image Builder

Anda dapat menggunakan salah satu metode berikut untuk membuat kebijakan siklus hidup gambar kontainer dengan atau tanpa AWS Management Console. AWS CLI Anda juga dapat menggunakan tindakan [CreateLifecyclePolicy](#) API. Untuk permintaan SDK terkait, Anda dapat merujuk ke tautan [Lihat Juga](#) untuk perintah tersebut di Referensi API EC2 Image Builder.

AWS Management Console

Untuk membuat kebijakan siklus hidup sumber daya gambar kontainer di AWS Management Console, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di](https://console.aws.amazon.com/imagebuilder/) <https://console.aws.amazon.com/imagebuilder/>.
2. Pilih Kebijakan Siklus Hidup dari panel navigasi.
3. Pilih Buat kebijakan siklus hidup.
4. Konfigurasi setelan kebijakan yang dijelaskan dalam prosedur berikut.
5. Untuk membuat kebijakan siklus hidup setelah Anda mengonfigurasi setelan, pilih Buat kebijakan.

Konfigurasi kebijakan: Pengaturan umum

Konfigurasi setelan Umum untuk kebijakan Anda.

1. Pilih opsi AMI dari Jenis kebijakan.
2. Masukkan nama Kebijakan.
3. Secara opsional masukkan Deskripsi untuk kebijakan siklus hidup Anda.
4. Secara default, Aktifkan dihidupkan. Setelan default mengaktifkan kebijakan siklus hidup dan langsung menambahkannya ke jadwal. Untuk membuat kebijakan yang awalnya dinonaktifkan, Anda dapat menonaktifkan Aktifkan.
5. Pilih peran IAM yang Anda buat untuk izin kebijakan siklus hidup. Jika Anda belum membuat peran ini, lihat [Prasyarat](#) untuk informasi selengkapnya.

Konfigurasi cakupan Aturan untuk kebijakan Anda.

Bagian ini mengonfigurasi pemilihan sumber daya untuk kebijakan siklus hidup Anda, berdasarkan jenis filter yang Anda gunakan.

1. Jenis filter: Resep — Untuk menerapkan aturan siklus hidup ke sumber daya gambar berdasarkan resep yang membuatnya, pilih hingga 50 versi resep untuk kebijakan tersebut.
2. Jenis filter: Tag — Untuk menerapkan aturan siklus hidup ke sumber daya gambar berdasarkan tag sumber daya, masukkan daftar hingga 50 pasangan nilai kunci agar kebijakan cocok.

Hapus aturan

Untuk gambar kontainer, aturan ini menghapus sumber daya gambar kontainer Image Builder. Anda secara opsional dapat menghapus gambar Docker yang didistribusikan ke repositori ECR untuk mencegahnya digunakan untuk menjalankan wadah baru.

Saat Anda mengonfigurasi aturan Hapus berdasarkan usia, Image Builder menghapus sumber daya gambar setelah periode waktu yang Anda konfigurasi. Misalnya, hapus sumber daya gambar setelah 6 bulan. Saat Anda mengonfigurasi berdasarkan hitungan, Image Builder mempertahankan jumlah gambar terbaru yang Anda tentukan, atau sedekat mungkin dengan angka tersebut, dan menghapus versi sebelumnya.

- Berdasarkan usia
 - Jumlah unit - Tentukan nilai integer untuk periode waktu yang harus dilewati setelah sumber daya gambar dibuat sebelum dihapus.
 - Unit — Pilih rentang waktu yang akan digunakan. Kisarannya bisa `Days`, `Weeks`, `Months`, atau `Years`.
 - Pertahankan setidaknya satu gambar — Pilih kotak centang untuk menyimpan hanya sumber daya gambar terbaru yang tersedia untuk setiap versi resep yang dipengaruhi aturan ini.

Dengan hitungan

- Jumlah gambar - Tentukan nilai integer untuk jumlah sumber daya gambar terbaru yang akan disimpan untuk setiap versi resep.
- Hapus gambar kontainer ECR - Pilih kotak centang untuk menghapus gambar kontainer terkait yang disimpan dalam repositori ECR. Anda tidak dapat lagi menggunakan gambar kontainer sebagai basis untuk membuat gambar baru, atau untuk menjalankan kontainer baru.

- Mempertahankan gambar dengan tag terkait - Pilih kotak centang untuk memasukkan daftar tag untuk sumber daya gambar yang ingin Anda simpan.

Tanda (opsional)

Tambahkan tag ke kebijakan siklus hidup Anda.

AWS CLI

Untuk membuat kebijakan lifecycle Image Builder baru, Anda dapat menggunakan [create-lifecycle-policy](#) perintah di AWS CLI

Cara kerja aturan manajemen siklus hidup untuk sumber daya gambar EC2 Image Builder

Kebijakan siklus hidup gambar menggunakan aturan siklus hidup yang Anda tentukan untuk menerapkan strategi pengelolaan sumber daya secara keseluruhan. Aturan yang Anda tetapkan membantu memastikan kesegaran gambar yang tersedia dan meminimalkan biaya untuk infrastruktur dasar seperti penyimpanan snapshot untuk AMI keluaran, atau penyimpanan repositori ECR dan kecepatan transfer data untuk gambar kontainer.

Anda dapat mengonfigurasi jenis aturan berikut untuk kebijakan Anda.

Menghentikan aturan

Menetapkan status sumber daya gambar Image Builder ke`Deprecated`. Pipeline Image Builder masih berjalan untuk gambar yang tidak digunakan lagi. Anda dapat secara opsional menyetel waktu penghentian untuk AMI terkait tanpa memengaruhi kemampuan Anda untuk meluncurkan instance baru.

Ketika AMI tidak digunakan lagi, AMI akan diabaikan oleh penelusuran umum. Misalnya, jika Anda menjalankan `describe-images` perintah Amazon EC2 di AWS CLI, itu tidak akan mengembalikan AMI usang dalam set hasil. Namun, Anda masih dapat menemukan AMI yang tidak digunakan lagi dengan ID AMI mereka.

Aturan ini tidak tersedia untuk gambar kontainer.

Nonaktifkan aturan

Menetapkan status sumber daya gambar Image Builder ke `Disabled`. Ini mencegah pipeline Image Builder berjalan untuk gambar ini. Anda dapat menonaktifkan AMI terkait secara opsional untuk mencegah peluncuran instans baru.

Ketika AMI dinonaktifkan, AMI menjadi pribadi dan tidak dapat digunakan untuk meluncurkan instance baru. Jika Anda membagikan AMI dengan akun, organisasi, atau unit organisasi apa pun, mereka kehilangan akses ke AMI Anda saat menjadi pribadi.

Aturan ini tidak tersedia untuk gambar kontainer.

Hapus aturan

Menghapus sumber daya gambar berdasarkan usia atau hitungan. Anda menentukan ambang batas yang memenuhi kebutuhan Anda. Ketika sumber daya gambar Image Builder melewati ambang batas, itu dihapus. Anda dapat secara opsional membatalkan pendaftaran AMI terkait atau menghapus snapshot untuk AMI tersebut. Anda juga dapat menentukan tag untuk sumber daya yang ingin Anda pertahankan melewati ambang batas.

Untuk gambar kontainer, aturan ini menghapus sumber daya gambar kontainer Image Builder. Anda dapat secara opsional menghapus gambar kontainer yang didistribusikan ke repositori ECR untuk mencegahnya digunakan untuk menjalankan kontainer baru.

Daftar Isi

- [Aturan pengecualian \(API/SDK/CLI\)](#)
- [Melihat detail aturan manajemen siklus hidup untuk kebijakan](#)

Aturan pengecualian (API/SDK/CLI)

Aturan pengecualian berikut menentukan pengecualian pada aturan siklus hidup untuk AMI. AMI yang memenuhi kriteria yang ditentukan oleh aturan pengecualian dikecualikan dari tindakan siklus hidup. Aturan pengecualian tidak tersedia di AWS Management Console

Istilah berikut menggunakan notasi API dari tipe [LifecyclePolicyDetailExclusionRules](#) data.

Aturan pengecualian

amis

Berisi pengaturan yang LifecyclePolicyDetailExclusionRulesAmis ditunjukkan dalam daftar berikut.

TagMap

Anda dapat memberikan daftar hingga 50 tag yang melewati tindakan siklus hidup untuk semua jenis sumber daya.

Istilah berikut menggunakan notasi API dari tipe [LifecyclePolicyDetailExclusionRulesAmis](#) data.

Aturan pengecualian AMI

isPublic

Mengkonfigurasi apakah AMI publik dikecualikan dari tindakan siklus hidup.

Terakhir Diluncurkan

Menentukan detail konfigurasi untuk Image Builder untuk mengecualikan sumber daya terbaru dari tindakan siklus hidup.

daerah

Mengkonfigurasi Wilayah AWS yang dikecualikan dari tindakan siklus hidup.

SharedAccounts

Menentukan sumber daya Akun AWS yang dikecualikan dari tindakan siklus hidup.

TagMap

Daftar tag yang harus dikecualikan dari tindakan siklus hidup untuk AMI yang memilikinya.

Melihat detail aturan manajemen siklus hidup untuk kebijakan

Aturan ditentukan dalam kebijakan manajemen siklus hidup yang Anda buat untuk sumber daya gambar Image Builder. Di konsol, halaman detail kebijakan siklus hidup memiliki halaman [Tab Aturan](#) yang menampilkan detail aturan yang dikonfigurasi untuk kebijakan tersebut.

Untuk mendapatkan detail kebijakan di AWS CLI, Anda dapat menjalankan [get-lifecycle-policy](#) perintah. Detail kebijakan dalam respons berisi daftar tindakan (aturan) yang Anda tetapkan untuk kebijakan tersebut, yang mencakup semua setelan yang dikonfigurasi.

Mengelola alur kerja build dan test untuk image EC2 Image Builder

Alur kerja gambar menentukan urutan langkah yang dilakukan EC2 Image Builder selama tahap pembuatan dan pengujian proses pembuatan gambar. Ini adalah bagian dari keseluruhan kerangka kerja Image Builder.

Manfaat alur kerja gambar

- Dengan alur kerja gambar, Anda memiliki lebih banyak fleksibilitas, visibilitas, dan kontrol atas proses pembuatan gambar.
- Anda dapat menambahkan langkah alur kerja yang disesuaikan saat menentukan dokumen alur kerja, atau Anda dapat memilih untuk menggunakan alur kerja default Image Builder.
- Anda dapat mengecualikan langkah-langkah alur kerja yang disertakan dalam alur kerja gambar default.
- Anda dapat membuat alur kerja khusus pengujian yang melewati proses pembuatan sepenuhnya. Anda dapat melakukan hal yang sama untuk membuat alur kerja khusus build.

Note

Anda tidak dapat mengubah alur kerja yang ada, tetapi Anda dapat mengkloningnya atau membuat versi baru.

Kerangka alur kerja: Tahapan

Untuk menyesuaikan alur kerja gambar, penting untuk memahami tahapan alur kerja yang membentuk kerangka kerja alur kerja pembuatan gambar.

Kerangka kerja pembuatan gambar mencakup dua tahapan berbeda berikut.

1. Tahap build (pre-snapshot) — Selama tahap build, Anda membuat perubahan pada instance build Amazon EC2 yang menjalankan image dasar, untuk membuat baseline untuk image baru Anda. Misalnya, resep Anda dapat menyertakan komponen yang menginstal aplikasi atau memodifikasi pengaturan firewall sistem operasi.

Setelah tahap ini selesai dengan sukses, Image Builder membuat snapshot atau image kontainer yang digunakan untuk tahap pengujian dan seterusnya.

2. Tahap pengujian (post-snapshot) — Selama tahap pengujian, ada beberapa perbedaan antara gambar yang membuat AMI dan gambar kontainer. Untuk alur kerja AMI, Image Builder meluncurkan instans EC2 dari snapshot yang dibuatnya sebagai langkah terakhir dari tahap pembuatan. Pengujian dijalankan pada instance baru untuk memvalidasi pengaturan dan memastikan bahwa instance berfungsi seperti yang diharapkan. Untuk alur kerja kontainer, pengujian dijalankan pada instance yang sama yang digunakan untuk membangun.

Kerangka kerja alur kerja juga mencakup tahap distribusi. Namun, Image Builder menangani alur kerja untuk tahap itu.

Akses layanan

Untuk menjalankan alur kerja gambar, Image Builder memerlukan izin untuk melakukan tindakan alur kerja. Anda dapat menentukan peran [AWSServiceRoleForImageBuilder](#) terkait layanan, atau Anda dapat menentukan peran kustom Anda sendiri untuk akses layanan, sebagai berikut.

- Konsol — Di panduan pipeline Langkah 3 Tentukan proses pembuatan gambar, pilih peran terkait layanan atau peran kustom Anda sendiri dari daftar peran IAM di panel akses Layanan.
- Image Builder API - Dalam permintaan [CreateImage](#) tindakan, tentukan peran terkait layanan atau peran kustom Anda sendiri sebagai nilai untuk parameter. `executionRole`

Untuk mempelajari lebih lanjut tentang cara membuat peran layanan, lihat [Membuat peran untuk mendelegasikan izin ke AWS layanan](#) di AWS Identity and Access Management Panduan Pengguna.

Daftar Isi

- [Daftar alur kerja gambar](#)
- [Buat alur kerja gambar](#)
- [Membuat dokumen alur kerja YAMB](#)

Daftar alur kerja gambar

Pada halaman daftar alur kerja Gambar di konsol Image Builder, Anda bisa mendapatkan daftar sumber daya alur kerja gambar yang Anda miliki atau memiliki akses, bersama dengan beberapa detail penting tentang sumber daya ini. Anda juga dapat menggunakan perintah atau tindakan

dengan Image Builder API, SDK, atau AWS CLI untuk membuat daftar alur kerja gambar di akun Anda.

Anda dapat menggunakan salah satu metode berikut untuk membuat daftar sumber daya alur kerja gambar yang Anda miliki atau memiliki akses ke. Untuk tindakan API, lihat [ListWorkflows](#) di Referensi API EC2 Image Builder. Untuk permintaan SDK terkait, lihat tautan [Lihat Juga](#) di halaman yang sama.

Console

Rincian alur kerja

Detail pada halaman daftar alur kerja Image di konsol Image Builder mencakup hal-hal berikut:

- **Workflow** - Nama versi terbaru dari sumber daya alur kerja gambar. Di konsol Image Builder, kolom Workflow menautkan ke halaman detail alur kerja.
- **Versi** - Versi terbaru dari sumber daya alur kerja gambar.
- **Jenis** - Jenis alur kerja: BUILD atau TEST.
- **Pemilik** — Pemilik sumber daya alur kerja.
- **Waktu pembuatan** - Tanggal dan waktu ketika Image Builder membuat versi terbaru dari sumber daya alur kerja gambar.
- **ARN** - Nama Sumber Daya Amazon (ARN) dari versi sumber daya alur kerja gambar saat ini.

Daftar alur kerja gambar

Untuk mencantumkan sumber daya alur kerja gambar di konsol Image Builder, lakukan langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Alur kerja gambar dari panel navigasi.

Filter hasil

Pada halaman daftar alur kerja Gambar, Anda dapat mencari alur kerja gambar tertentu untuk memfilter hasil Anda. Filter berikut tersedia untuk alur kerja gambar:

Workflow

Anda dapat memasukkan semua atau sebagian nama alur kerja untuk merampingkan hasil. Defaultnya adalah menampilkan semua alur kerja dalam daftar.

Version

Anda dapat memasukkan semua atau sebagian dari nomor versi untuk merampingkan hasil. Defaultnya adalah menampilkan semua versi dalam daftar.

Type

Anda dapat memfilter berdasarkan jenis alur kerja atau melihat semua jenis. Standarnya adalah menampilkan semua jenis alur kerja dalam daftar.

- MEMBANGUN
- UJI

Owner

Saat Anda memilih filter pemilik dari bilah pencarian, Image Builder menampilkan daftar pemilik alur kerja gambar di akun Anda. Anda dapat memilih pemilik dari daftar untuk merampingkan hasil. Defaultnya adalah menampilkan semua pemilik dalam daftar.

- Akun AWS— Akun yang memiliki sumber daya alur kerja.
- Amazon — Sumber daya alur kerja yang dimiliki dan dikelola Amazon.

AWS CLI

Ketika Anda menjalankan [list-workflows](#) perintah diAWS CLI, Anda bisa mendapatkan daftar alur kerja gambar yang Anda miliki atau memiliki akses ke.

Contoh perintah berikut menunjukkan cara menggunakan list-workflows perintah tanpa filter untuk mencantumkan semua sumber daya alur kerja gambar Image Builder yang Anda miliki atau memiliki akses ke.

Contoh: daftar semua alur kerja gambar

```
aws imagebuilder list-workflows
```

Output:

```
{  
  "workflowVersionList": [  

```

```
{
  "name": "example-test-workflow",
  "dateCreated": "2023-11-21T22:53:14.347Z",
  "version": "1.0.0",
  "owner": "111122223333",
  "type": "TEST",
  "arn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/test/example-test-workflow/1.0.0"
},
{
  "name": "example-build-workflow",
  "dateCreated": "2023-11-20T12:26:10.425Z",
  "version": "1.0.0",
  "owner": "111122223333",
  "type": "BUILD",
  "arn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/build/example-build-workflow/1.0.0"
}
]
```

Saat Anda menjalankan `list-workflows` perintah, Anda dapat menerapkan filter untuk merampingkan hasil, seperti yang ditunjukkan contoh berikut. Untuk informasi selengkapnya tentang cara memfilter hasil, lihat perintah [daftar-alur kerja](#) di Referensi Perintah. AWS CLI

Contoh: filter untuk alur kerja build

```
aws imagebuilder list-workflows --filters name="type",values="BUILD"
```

Output:

```
{
  "workflowVersionList": [
    {
      "name": "example-build-workflow",
      "dateCreated": "2023-11-20T12:26:10.425Z",
      "version": "1.0.0",
      "owner": "111122223333",
      "type": "BUILD",
      "arn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/build/example-build-workflow/1.0.0"
    }
  ]
}
```

```
]
}
```

Buat alur kerja gambar

Saat Anda membuat alur kerja gambar, Anda memiliki kontrol lebih besar atas proses pembuatan gambar Anda. Anda dapat menentukan alur kerja apa yang berjalan saat Image Builder membuat gambar Anda, dan alur kerja apa yang dijalankan saat menguji gambar. Anda juga dapat menentukan kunci yang dikelola pelanggan untuk mengenkripsi sumber daya alur kerja Anda. Untuk mempelajari lebih lanjut tentang enkripsi sumber daya alur kerja Anda, lihat [Enkripsi dan manajemen kunci di EC2 Image Builder](#).

Untuk pembuatan gambar, Anda dapat menentukan satu alur kerja tahap build, dan satu atau beberapa alur kerja tahap pengujian. Anda bahkan dapat melewati tahap pembuatan atau pengujian sepenuhnya, tergantung pada kebutuhan Anda. Anda mengonfigurasi tindakan yang diambil alur kerja Anda dalam dokumen definisi YAMM yang digunakan alur kerja Anda. Untuk informasi selengkapnya tentang sintaks untuk dokumen YAMM Anda, lihat [Membuat dokumen alur kerja YAMB](#)

Untuk langkah-langkah untuk membuat alur kerja build atau pengujian baru, pilih tab yang cocok dengan lingkungan yang akan Anda gunakan.

AWS Management Console

Anda dapat menggunakan proses berikut untuk membuat alur kerja baru di konsol Image Builder.

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Pilih Alur kerja gambar dari panel navigasi. Ini menampilkan daftar alur kerja gambar yang dimiliki atau diakses oleh akun Anda.

Note

Anda akan selalu melihat sumber daya alur kerja terkelola Amazon yang digunakan Image Builder untuk alur kerja defaultnya dalam daftar Anda. Untuk melihat detail alur kerja tersebut, Anda dapat memilih tautan Alur Kerja.

3. Untuk membuat alur kerja baru, pilih Buat alur kerja gambar. Ini menampilkan halaman alur kerja Buat gambar.

4. Konfigurasi detail untuk alur kerja baru Anda. Untuk membuat alur kerja build, pilih opsi Build di dekat bagian atas formulir. Untuk membuat alur kerja pengujian, pilih opsi Uji di dekat bagian atas formulir. Image Builder mengisi daftar Template berdasarkan opsi ini. Semua langkah lainnya sama untuk membangun dan menguji alur kerja.

Umum

Bagian umum mencakup pengaturan yang berlaku untuk sumber daya alur kerja Anda, seperti nama dan deskripsi. Pengaturan umum meliputi yang berikut:

- Nama alur kerja gambar (wajib) - Nama untuk alur kerja gambar Anda. Nama harus unik di akun Anda. Nama dapat memiliki panjang hingga 128 karakter. Karakter yang valid termasuk huruf, angka, spasi-, dan_.
- Versi (wajib) - Versi semantik untuk sumber daya alur kerja yang akan dibuat (major.minor.patch).
- Deskripsi (opsional) - Secara opsional tambahkan deskripsi untuk alur kerja Anda.
- Kunci KMS (opsional) - Anda dapat mengenkripsi sumber daya alur kerja Anda dengan kunci yang dikelola pelanggan. Untuk informasi selengkapnya, lihat [Enkripsi alur kerja gambar dengan kunci terkelola pelanggan](#).

Dokumen definisi

Dokumen alur kerja YAMM berisi semua konfigurasi untuk alur kerja Anda.

Mulai

- Untuk memulai dengan template default Image Builder sebagai dasar alur kerja Anda, pilih opsi Mulai dari templat. Opsi ini dipilih secara default. Setelah Anda memilih template apa yang akan digunakan dari daftar Template, ini akan menyalin konfigurasi default dari template yang Anda pilih ke dalam Konten untuk dokumen alur kerja baru Anda, di mana Anda dapat membuat perubahan.
- Untuk menentukan dokumen alur kerja Anda dari awal, pilih opsi Mulai dari awal. Ini mengisi Konten dengan garis besar singkat dari beberapa bagian penting dari format dokumen untuk membantu Anda memulai.

Panel Konten menyertakan bilah status di bagian bawah yang menampilkan peringatan atau kesalahan untuk dokumen YAMM Anda. Untuk informasi selengkapnya tentang cara membuat dokumen alur kerja YAMM, lihat [Membuat dokumen alur kerja YAMB](#)

5. Ketika Anda telah menyelesaikan alur kerja Anda, atau jika Anda ingin menyimpan kemajuan dan kembali lagi nanti, pilih Buat alur kerja.

AWS CLI

Sebelum Anda menjalankan [create-workflow](#) perintah diAWS CLI, Anda harus membuat dokumen YAMAL yang berisi semua konfigurasi untuk alur kerja Anda. Untuk informasi selengkapnya, lihat [Membuat dokumen alur kerja YAMB](#).

Contoh berikut menunjukkan cara membuat alur kerja build dengan perintah [AWS CLIcreate-workflow](#). --dataParameter mengacu pada dokumen YAMAL yang berisi konfigurasi build untuk alur kerja yang Anda buat.

Contoh: Buat alur kerja

```
aws imagebuilder create-workflow --name example-build-workflow --semantic-version 1.0.0 --type BUILD --data file://example-build-workflow.yml
```

Output:

```
{
  "workflowBuildVersionArn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/build/example-build-workflow/1.0.0/1",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

Contoh berikut menunjukkan cara membuat alur kerja pengujian dengan perintah [AWS CLIcreate-workflow](#). --dataParameter mengacu pada dokumen YAMAL yang berisi konfigurasi build untuk alur kerja yang Anda buat.

Contoh: Buat alur kerja pengujian

```
aws imagebuilder create-workflow --name example-test-workflow --semantic-version 1.0.0 --type TEST --data file://example-test-workflow.yml
```

Output:

```
{
  "workflowBuildVersionArn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/
test/example-test-workflow/1.0.0/1",
  "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

Membuat dokumen alur kerja YAMB

Dokumen definisi format YAMAL mengonfigurasi langkah input, output, dan alur kerja untuk tahap pembuatan dan pengujian proses pembuatan gambar. Anda dapat memulai dari template yang menyertakan langkah-langkah standar, atau Anda dapat mulai dari awal untuk menentukan alur kerja Anda sendiri. Apakah Anda menggunakan template atau mulai dari awal, Anda dapat menyesuaikan alur kerja agar sesuai dengan kebutuhan Anda.

Struktur dokumen alur kerja YAMB

Dokumen alur kerja YAMAL yang digunakan Image Builder untuk melakukan pembuatan gambar dan tindakan pengujian disusun sebagai berikut.

- [Identifikasi](#)
- [Parameter input](#)
- [Langkah-langkah](#)
- [Output](#)

Identifikasi

Mengidentifikasi alur kerja secara unik. Bagian ini dapat mencakup atribut berikut.

Bidang	Deskripsi	Tipe	Diperlukan
nama	Nama dokumen alur kerja.	String	Tidak

Bidang	Deskripsi	Tipe	Diperlukan
deskripsi	Deskripsi dokumen.	String	Tidak
schemaVersion	Versi skema dokumen, saat ini 1.0.	Tali	Ya

Contoh

```

---
name: sample-test-image
description: Workflow for a sample image, with extra configuration options exposed
  through workflow parameters.
schemaVersion: 1.0

```

Parameter input

Bagian dari dokumen alur kerja ini mendefinisikan parameter input yang dapat ditentukan oleh pemanggil. Jika Anda tidak memiliki parameter apa pun, Anda dapat meninggalkan bagian ini. Jika Anda menentukan parameter, setiap parameter dapat menyertakan atribut berikut.

Bidang	Deskripsi	Tipe	Diperlukan	Batasan
nama	Nama parameter	Tali	Ya	
deskripsi	Deskripsi parameter.	String	Tidak	
default	Nilai default parameter, jika tidak ada nilai yang disediakan. Jika Anda	Cocokkan dengan tipe data parameter.	Tidak	

Bidang	Deskripsi	Tipe	Diperlukan	Batasan
	tidak menyertakan nilai default dalam definisi parameter, nilai parameter diperlukan saat runtime.			
jenis	Tipe data parameter. Jika Anda tidak menyertakan tipe data dalam definisi parameter, tipe parameter default ke nilai string yang diperlukan saat runtime.	Tali	Ya	Tipe data parameter harus salah satu dari yang berikut: <ul style="list-style-type: none"> • string • integer • boolean • stringList

Contoh

Tentukan parameter dalam dokumen alur kerja.

```
parameters:
  - name: waitForActionAtEnd
    type: boolean
    default: true
    description: "Wait for an external action at the end of the workflow"
```

Gunakan nilai parameter dalam dokumen alur kerja.

```
$.parameters.waitForActionAtEnd
```

Langkah-langkah

Menentukan hingga 15 langkah tindakan untuk alur kerja. Langkah-langkah berjalan dalam urutan yang didefinisikan dalam dokumen alur kerja. Jika terjadi kegagalan, rollback berjalan dalam urutan terbalik, dimulai dengan langkah yang gagal, dan bekerja mundur melalui langkah-langkah sebelumnya.

Setiap langkah dapat merujuk ke output dari setiap tindakan langkah sebelumnya. Ini dikenal sebagai rantai, atau referensi. Untuk merujuk ke output dari tindakan langkah sebelumnya, Anda dapat menggunakan pemilih JsonPath. Sebagai contoh:

```
$.stepOutputs.step-name.output-name
```

Untuk informasi selengkapnya, lihat [Gunakan variabel dinamis dalam dokumen alur kerja Anda](#).

Note

Meskipun langkah itu sendiri tidak memiliki atribut output, output apa pun dari tindakan langkah disertakan `stepOutput` untuk langkah tersebut.

Setiap langkah dapat mencakup atribut berikut.

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default	Batasan
tindakan	Tindakan alur kerja yang dilakukan langkah ini.	Tali	Ya		Harus merupakan langkah tindakan yang didukung untuk dokumen alur kerja Image Builder.
		Dikte	Tidak		

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default	Batasan
<code>if</code> , diikuti oleh serangkaian pernyataan bersyarat yang memodifikasi <code>if</code> operator.	Pernyataan bersyarat menambahkan alur titik keputusan kontrol ke badan langkah alur kerja Anda.				<p>Image Builder mendukung pernyataan bersyarat berikut sebagai pengubah operator: <code>if</code></p> <ul style="list-style-type: none"> Kondisi percabangan dan pengubah: <code>if</code>, <code>and</code>, <code>or</code>, <code>not</code>. Kondisi percabangan ditentukan pada garis sendiri. Operator perbandingan: <code>booleanEquals</code>, <code>numberEquals</code>, <code>numberGreaterThan</code>, <code>numberGreaterThanEquals</code>, <code>numberLessThan</code>, <code>numberLessThanEquals</code>

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default	Batasan
					ls ,stringEquals .
deskripsi	Deskripsi langkah.	String	Tidak		String kosong tidak diperbolehkan. Jika disertakan, panjangnya harus 1-1024 karakter.
masukan	Berisi parameter yang perlu dijalankan oleh tindakan langkah. Anda dapat menentukan nilai kunci sebagai nilai statis, atau dengan variabel JsonPath yang menyelesaikan ke tipe data yang benar.	Dikte	Ya		

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default	Batasan
nama	Nama langkah. Nama ini harus unik dalam dokumen alur kerja.	Tali	Ya		Panjang harus antara 3-128 karakter. Dapat menyertakan karakter alfanumerik dan _ Tidak ada spasi.

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default	Batasan
onFailure	<p>Mengkonfigurasi tindakan yang akan diambil jika langkah gagal, sebagai berikut.</p> <p>Perilaku</p> <ul style="list-style-type: none"> Abort—Gagal langkah, gagal alur kerja, dan tidak menjalankan langkah yang tersisa setelah langkah yang gagal. Jika rollback diaktifkan, rollback dimulai dengan langkah yang 	String	Tidak	Abort	Abort Continue

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default	Batasan
	<p>gagal, dan berlanjut hingga semua langkah yang mungkin kannya diputar kembali.</p> <ul style="list-style-type: none">• Continue—Gagal langkah, tetapi terus menjalankan langkah-langkah yang tersisa setelah langkah yang gagal. Dalam hal ini, tidak ada rollback.				

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default	Batasan
RollBackEnabled	Mengkonfigurasi apakah langkah akan diputar kembali jika terjadi kegagalan. Anda dapat menggunakan nilai Boolean statis atau variabel JSONPath dinamis yang menyelesaikan nilai Boolean.	Boolean	Tidak	true	true false atau variabel JSONPath yang menyelesaikan ke true atau false.

Bidang	Deskripsi	Tipe	Diperlukan	Nilai default	Batasan
timeoutSeconds	Waktu maksimum, dalam hitungan detik, langkah berjalan sebelum gagal dan mencoba lagi, jika percobaan ulang berlaku.	Bulat	Tidak	Tergantung pada default yang ditentukan untuk tindakan langkah, jika berlaku.	Antara 1-86400 detik (maksimum 24 jam)

Contoh

```

steps:
  - name: LaunchTestInstance
    action: LaunchInstance
    onFailure: Abort
    inputs:
      waitFor: "ssmAgent"

  - name: ApplyTestComponents
    action: ExecuteComponents
    onFailure: Abort
    inputs:
      instanceId.$: "$.stepOutputs.LaunchTestInstance.instanceId"

  - name: TerminateTestInstance
    action: TerminateInstance
    onFailure: Continue
    inputs:
      instanceId.$: "$.stepOutputs.LaunchTestInstance.instanceId"

  - name: WaitForActionAtEnd

```

```

action: WaitForAction
if:
  booleanEquals: true
  value: "$.parameters.waitForActionAtEnd"

```

Output

Mendefinisikan output untuk alur kerja. Setiap output adalah pasangan nilai kunci yang secara spesifik nama output dan nilainya. Anda dapat menggunakan output untuk mengeksport data saat runtime yang dapat digunakan alur kerja berikutnya. Bagian ini opsional.

Setiap output yang Anda tentukan mencakup atribut berikut.

Bidang	Deskripsi	Tipe	Diperlukan
nama	Nama output. Nama harus unik di seluruh alur kerja yang Anda sertakan dalam pipeline Anda.	Tali	Ya
nilai	Nilai untuk output. Nilai string dapat berupa variabel dyanmic, seperti file output dari tindakan langkah. Untuk informasi selengkapnya, lihat Gunakan variabel dinamis dalam dokumen alur kerja Anda .	Tali	Ya

Contoh

Buat ID gambar keluaran untuk dokumen alur kerja dengan output langkah dari `createProdImage` langkah.

```
outputs:  
  - name: 'outputImageId'  
    value: '$.stepOutputs.createProdImage.imageId'
```

Lihat output alur kerja di alur kerja berikutnya.

```
$.workflowOutputs.outputImageId
```

Tindakan langkah yang didukung untuk dokumen alur kerja Anda

Bagian ini mencakup detail untuk tindakan langkah yang didukung Image Builder.

Istilah yang digunakan dalam bagian ini

AMI

Gambar Mesin Amazon

ARN

Nama Sumber Daya Amazon

Tindakan yang didukung

- [BootstrapInstanceForContainer](#)
- [CollectImageMetadata](#)
- [CollectImageScanFindings](#)
- [CreateImage](#)
- [ExecuteComponents](#)
- [LaunchInstance](#)
- [RunCommand](#)
- [RunSysPrep](#)
- [SanitizeInstance](#)
- [TerminateInstance](#)

- [WaitForAction](#)

BootstrapInstanceForContainer

Tindakan langkah ini menjalankan skrip layanan untuk mem-bootstrap instance dengan persyaratan minimum untuk menjalankan alur kerja kontainer. Image Builder menggunakan API Systems Manager untuk menjalankan skrip ini. `sendCommand` Untuk informasi selengkapnya, lihat [AWS Systems ManagerMenjalankan Perintah](#).

Note

Skrip bootstrap menginstal paket AWS CLI dan Docker yang merupakan prasyarat bagi Image Builder agar berhasil membangun wadah Docker. Jika Anda tidak menyertakan tindakan langkah ini, pembuatan gambar bisa gagal.

Batas Waktu Default: 60 menit

Rollback: Tidak ada rollback untuk tindakan langkah ini.

Input: Tabel berikut mencakup input yang didukung untuk tindakan langkah ini.

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
instanceld	ID dari instance untuk bootstrap.	Tali	Ya		Ini harus berupa ID instance keluaran dari langkah alur kerja yang meluncurkan instance untuk alur kerja ini.

Output: Tabel berikut mencakup output untuk tindakan langkah ini.

Nama output	Deskripsi	Tipe
runCommandId	ID Systems Manager sendCommand yang menjalankan skrip bootstrap pada instance.	String
status	Status dikembalikan dari Systems ManagersendCommand.	String
output	Output dikembalikan dari Systems ManagersendCommand.	String

Contoh

Tentukan tindakan langkah dalam dokumen alur kerja.

```
- name: ContainerBootstrapStep
  action: BootstrapInstanceForContainer
  onFailure: Abort
  inputs:
    instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Gunakan output dari nilai tindakan langkah dalam dokumen alur kerja.

```
$.stepOutputs.ContainerBootstrapStep.status
```

CollectImageMetadata

Tindakan langkah ini hanya berlaku untuk alur kerja build.

EC2 Image Builder [AWS Systems Manager](#) menjalankan Agen ([Systems Manager](#)) pada instans EC2 yang diluncurkan untuk membangun dan menguji gambar Anda. Image Builder mengumpulkan informasi tambahan tentang instance yang digunakan selama fase build dengan [Systems Manager Inventory](#). Informasi ini mencakup nama dan versi sistem operasi (OS), serta daftar paket dan versinya masing-masing seperti yang dilaporkan oleh sistem operasi Anda.

Note

Tindakan langkah ini hanya berfungsi untuk gambar yang membuat AMI.

Batas Waktu Default: 30 menit

Rollback: Image Builder mengembalikan semua sumber daya Systems Manager yang dibuat selama langkah ini.

Input: Tabel berikut mencakup input yang didukung untuk tindakan langkah ini.

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
instanceld	Instans build untuk menerapkan pengaturan metadata ke.	Tali	Ya		Ini harus berupa ID instance keluaran dari langkah alur kerja yang meluncurkan instance build untuk alur kerja ini.

Output: Tabel berikut mencakup output untuk tindakan langkah ini.

Nama output	Deskripsi	Tipe
osVersion	Nama dan versi sistem operasi yang dikumpulkan dari instance build.	String
AssociationID	ID asosiasi Systems Manager yang digunakan untuk pengumpulan inventaris.	String

Contoh

Tentukan tindakan langkah dalam dokumen alur kerja.

```
- name: CollectMetadataStep
  action: CollectImageMetadata
  onFailure: Abort
  inputs:
    instanceId: $.stepOutputs.LaunchStep.instanceId
```

Gunakan output dari tindakan langkah dalam dokumen alur kerja.

```
$.stepOutputs.CollectMetadataStep.osVersion
```

CollectImageScanFindings

Jika Amazon Inspector diaktifkan untuk akun Anda dan pemindaian gambar diaktifkan untuk pipeline Anda, tindakan langkah ini mengumpulkan temuan pemindaian gambar yang dilaporkan oleh Amazon Inspector untuk instance pengujian Anda. Tindakan langkah ini tidak tersedia untuk alur kerja build.

Batas Waktu Default: 120 menit

Rollback: Tidak ada rollback untuk tindakan langkah ini.

Input: Tabel berikut mencakup input yang didukung untuk tindakan langkah ini.

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
instanceld	ID untuk contoh pemindaian berjalan.	Tali	Ya		Ini harus berupa ID instance keluaran dari langkah alur kerja yang meluncurkan instance untuk alur kerja ini.

Output: Tabel berikut mencakup output untuk tindakan langkah ini.

Nama output	Deskripsi	Tipe
runCommandId	ID Systems Manager sendCommand yang menjalankan skrip untuk mengumpulkan temuan.	String
status	Status dikembalikan dari Systems ManagersendCommand.	String
output	Output dikembalikan dari Systems ManagersendCommand.	String

Contoh

Tentukan tindakan langkah dalam dokumen alur kerja.

```
- name: CollectFindingsStep
  action: CollectImageScanFindings
  onFailure: Abort
  inputs:
    instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Gunakan output dari nilai tindakan langkah dalam dokumen alur kerja.

```
$.stepOutputs.CollectFindingsStep.status
```

CreateImage

Tindakan langkah ini membuat gambar dari instance yang sedang berjalan dengan Amazon EC2 API `CreateImage`. Selama proses pembuatan, tindakan langkah menunggu seperlunya untuk memverifikasi bahwa sumber daya telah mencapai keadaan yang benar sebelum dilanjutkan.

Batas Waktu Default: 720 menit

Rollback: Tidak ada rollback untuk tindakan langkah ini.

Input: Tabel berikut mencakup input yang didukung untuk tindakan langkah ini.

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
instanceld	Contoh untuk membuat gambar baru dari.	Tali	Ya		Instance untuk ID instance yang disediakan harus dalam <code>running</code> keadaan saat langkah ini dimulai.

Output: Tabel berikut mencakup output untuk tindakan langkah ini.

Nama output	Deskripsi	Tipe
imageId	ID AMI dari gambar yang dibuat.	String

Contoh

Tentukan tindakan langkah dalam dokumen alur kerja.

```
- name: CreateImageFromInstance
  action: CreateImage
  onFailure: Abort
  inputs:
    instanceId.$: "i-1234567890abcdef0"
```

Gunakan output dari nilai tindakan langkah dalam dokumen alur kerja.

```
$.stepOutputs.CreateImageFromInstance.imageId
```

ExecuteComponents

Tindakan langkah ini menjalankan komponen yang ditentukan dalam resep untuk gambar saat ini yang sedang dibangun. Alur kerja build menjalankan komponen build pada instance build. Alur kerja pengujian hanya menjalankan komponen pengujian pada instance pengujian.

Image Builder menggunakan API Systems Manager untuk menjalankan komponen. `sendCommand` Untuk informasi selengkapnya, lihat [AWS Systems ManagerMenjalankan Perintah](#).

Batas Waktu Default: 720 menit

Rollback: Tidak ada rollback untuk tindakan langkah ini.

Input: Tabel berikut mencakup input yang didukung untuk tindakan langkah ini.

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
instanceId	ID untuk contoh bahwa komponen harus berjalan.	Tali	Ya		Ini harus berupa ID instance keluaran dari langkah alur kerja yang meluncurkan instance untuk alur kerja ini.

Output: Tabel berikut mencakup output untuk tindakan langkah ini.

Nama output	Deskripsi	Tipe
runCommandId	ID Systems Manager <code>sendCommand</code> yang menjalankan komponen pada instance.	String

Nama output	Deskripsi	Tipe
status	Status dikembalikan dari Systems ManagersendCommand.	String
output	Output dikembalikan dari Systems ManagersendCommand.	String

Contoh

Tentukan tindakan langkah dalam dokumen alur kerja.

```
- name: ExecComponentsStep
  action: ExecuteComponents
  onFailure: Abort
  inputs:
    instanceId: $.stepOutputs.LaunchStep.instanceId
```

Gunakan output dari tindakan langkah dalam dokumen alur kerja.

```
$.stepOutputs.ExecComponentsStep.status
```

LaunchInstance

Tindakan langkah ini meluncurkan instance di Anda Akun AWS dan menunggu hingga agen Systems Manager berjalan pada instance sebelum melanjutkan ke langkah berikutnya. Tindakan peluncuran menggunakan pengaturan dari sumber daya konfigurasi resep dan infrastruktur yang terkait dengan gambar Anda. Misalnya, jenis instance yang akan diluncurkan berasal dari konfigurasi infrastruktur. Outputnya adalah ID instance dari instance yang diluncurkan.

`waitForInput` mengkonfigurasi kondisi yang memenuhi persyaratan penyelesaian langkah.

Batas Waktu Default: 60 menit

Rollback: Untuk instance build, rollback melakukan tindakan yang telah dikonfigurasi di sumber daya konfigurasi infrastruktur. Secara default, instance build dihentikan jika pembuatan gambar

gagal. Namun, ada pengaturan dalam konfigurasi infrastruktur untuk menjaga instance build untuk pemecahan masalah.

Input: Tabel berikut mencakup input yang didukung untuk tindakan langkah ini.

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
Menunggu	Kondisi untuk menunggu sebelum menyelesaikan langkah alur kerja dan melanjutkan ke langkah berikutnya.	Tali	Ya		Image Builder saat ini mendukung ssmAgent.

Output: Tabel berikut mencakup output untuk tindakan langkah ini.

Nama output	Deskripsi	Tipe
instanceId	ID instance dari instance yang diluncurkan.	String

Contoh

Tentukan tindakan langkah dalam dokumen alur kerja.

```
- name: LaunchStep
  action: LaunchInstance
  onFailure: Abort
  inputs:
    waitFor: ssmAgent
```

Gunakan output dari tindakan langkah dalam dokumen alur kerja.

```
$.stepOutputs.LaunchStep.instanceId
```

RunCommand

Tindakan langkah ini menjalankan dokumen perintah untuk alur kerja Anda. Image Builder menggunakan API Systems Manager untuk menjalankannya untuk Anda. `sendCommand` Untuk informasi selengkapnya, lihat [AWS Systems ManagerMenjalankan Perintah](#).

Batas Waktu Default: 12 jam

Rollback: Tidak ada rollback untuk tindakan langkah ini.

Input: Tabel berikut mencakup input yang didukung untuk tindakan langkah ini.

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
instanceId	ID dari instance untuk menjalankan dokumen perintah pada.	Tali	Ya		Ini harus berupa ID instance keluaran dari langkah alur kerja yang meluncurkan instance untuk alur kerja ini.
documentName	Nama dokumen perintah Systems Manager untuk dijalankan.	Tali	Ya		
parameter	Daftar pasangan nilai kunci	<string>kamus <string, daftar>	Bersyarat		

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
	untuk parameter apa pun yang diperlukan dokumen perintah.				
DocumentVersion	Versi dokumen perintah untuk dijalankan.	String	Tidak	\$DEFAULT	

Output: Tabel berikut mencakup output untuk tindakan langkah ini.

Nama output	Deskripsi	Tipe
runCommandId	ID Systems Manager sendCommand yang menjalankan dokumen perintah pada instance.	String
status	Status dikembalikan dari Systems ManagersendCommand.	String
output	Output dikembalikan dari Systems ManagersendCommand.	Daftar string

Contoh

Tentukan tindakan langkah dalam dokumen alur kerja.

```

- name: RunCommandDoc
  action: RunCommand
  onFailure: Abort
  inputs:
    documentName: SampleDocument
    parameters:
      osPlatform:
        - "linux"
    instanceId.$: $.stepOutputs.LaunchStep.instanceId

```

Gunakan output dari nilai tindakan langkah dalam dokumen alur kerja.

```
$.stepOutputs.RunCommandDoc.status
```

RunSysPrep

Tindakan langkah ini menggunakan API Systems Manager untuk menjalankan AWSEC2-RunSysprep dokumen untuk instance Windows sebelum instance build dimatikan untuk snapshot. sendCommand Tindakan ini mengikuti [praktik AWS terbaik untuk pengerasan dan pembersihan gambar](#).

Batas Waktu Default: 60 menit

Rollback: Tidak ada rollback untuk tindakan langkah ini.

Input: Tabel berikut mencakup input yang didukung untuk tindakan langkah ini.

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
instanceId	ID dari instance untuk menjalankan AWSEC2-RunSysprep dokumen pada.	Tali	Ya		Ini harus berupa ID instance keluaran dari langkah alur kerja yang meluncurkan instance

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
					untuk alur kerja ini.

Output: Tabel berikut mencakup output untuk tindakan langkah ini.

Nama output	Deskripsi	Tipe
runCommandId	ID Systems Manager sendCommand yang menjalankan AWSEC2-RunSysprep dokumen pada instance.	String
status	Status dikembalikan dari Systems ManagersendCommand.	String
output	Output dikembalikan dari Systems ManagersendCommand.	String

Contoh

Tentukan tindakan langkah dalam dokumen alur kerja.

```
- name: RunSysprep
  action: RunSysPrep
  onFailure: Abort
  inputs:
    instanceId.$: $.stepOutputs.LaunchStep.instanceId
```

Gunakan output dari nilai tindakan langkah dalam dokumen alur kerja.

```
$.stepOutputs.RunSysprep.status
```

SanitizeInstance

Tindakan langkah ini menjalankan skrip sanitasi yang direkomendasikan untuk instance Linux sebelum instance build dimatikan untuk snapshot. Skrip sanitasi membantu memastikan bahwa gambar akhir mengikuti praktik terbaik keamanan, dan artefak atau pengaturan build yang tidak boleh dibawa ke snapshot Anda dihapus. Untuk informasi lebih lanjut tentang skrip, lihat [Diperlukan pembersihan pasca-build](#). Tindakan langkah ini tidak berlaku untuk gambar kontainer.

Image Builder menggunakan API Systems Manager untuk menjalankan skrip ini. `sendCommand` Untuk informasi selengkapnya, lihat [AWS Systems ManagerMenjalankan Perintah](#).

Batas Waktu Default: 60 menit

Rollback: Tidak ada rollback untuk tindakan langkah ini.

Input: Tabel berikut mencakup input yang didukung untuk tindakan langkah ini.

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
<code>instanceId</code>	ID dari instance untuk membersihkan.	Tali	Ya		Ini harus berupa ID instance keluaran dari langkah alur kerja yang meluncurkan instance untuk alur kerja ini.

Output: Tabel berikut mencakup output untuk tindakan langkah ini.

Nama output	Deskripsi	Tipe
<code>runCommandId</code>	ID Systems Manager <code>sendCommand</code> yang	String

Nama output	Deskripsi	Tipe
	menjalankan skrip sanitasi pada instance.	
status	Status dikembalikan dari Systems Managers sendCommand.	String
output	Output dikembalikan dari Systems Managers sendCommand.	String

Contoh

Tentukan tindakan langkah dalam dokumen alur kerja.

```
- name: SanitizeStep
  action: SanitizeInstance
  onFailure: Abort
  inputs:
    instanceId: $.stepOutputs.LaunchStep.instanceId
```

Gunakan output dari nilai tindakan langkah dalam dokumen alur kerja.

```
$.stepOutputs.SanitizeStep.status
```

TerminateInstance

Tindakan langkah ini mengakhiri instance dengan id instance yang diteruskan sebagai input.

Batas Waktu Default: 30 menit

Rollback: Tidak ada rollback untuk tindakan langkah ini.

Input: Tabel berikut mencakup input yang didukung untuk tindakan langkah ini.

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
instanceId	ID dari instance untuk mengakhiri.	Tali	Ya		

Output: Tidak ada output untuk tindakan langkah ini.

Contoh

Tentukan tindakan langkah dalam dokumen alur kerja.

```
- name: TerminateInstance
  action: TerminateInstance
  onFailure: Continue
  inputs:
    instanceId.$: i-1234567890abcdef0
```

WaitForAction

Tindakan langkah ini menghentikan sementara alur kerja yang sedang berjalan dan menunggu untuk menerima tindakan eksternal dari tindakan Image Builder API. `SendWorkflowStepAction` Langkah ini menerbitkan acara ke bus EventBridge EventBridge acara default Anda dengan tipe `EC2 Image Builder Workflow Step Waiting detail`. Langkah ini juga dapat mengirim pemberitahuan SNS jika Anda memberikan SNS Topic ARN.

Batas Waktu Default: 3 hari

Rollback: Tidak ada rollback untuk tindakan langkah ini.

Input: Tabel berikut mencakup input yang didukung untuk tindakan langkah ini.

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
snsTopicArn	Topik SNS opsional	String	Tidak		

Nama masukan	Deskripsi	Tipe	Diperlukan	Default	Batasan
	ARN untuk mengirim pemberitahuan saat langkah alur kerja tertunda.				

Output: Tabel berikut mencakup output untuk tindakan langkah ini.

Nama output	Deskripsi	Tipe
tindakan	Tindakan yang dikembalikan oleh tindakan <code>SendWorkflowStepAction</code> API.	String (RESUME atau STOP)
akal budi	Alasan untuk tindakan yang dikembalikan.	String

Contoh

Tentukan tindakan langkah dalam dokumen alur kerja.

```
- name: SendEventAndWait
  action: WaitForAction
  onFailure: Abort
  inputs:
    snsTopicArn: arn:aws:sns:us-west-2:111122223333:ExampleTopic
```

Gunakan output dari nilai tindakan langkah dalam dokumen alur kerja.

```
$.stepOutputs.SendEventAndWait.reason
```

Gunakan variabel dinamis dalam dokumen alur kerja Anda

Anda dapat menggunakan variabel dinamis dalam dokumen alur kerja Anda untuk mewakili nilai yang bervariasi saat runtime untuk proses pembuatan gambar Anda. Nilai variabel dinamis direpresentasikan sebagai pemilih JSONPath dengan node struktural yang secara unik mengidentifikasi variabel target.

Struktur variabel alur kerja dinamis JsonPath

```
$.<document structure>.[<step name>].<variable name>
```

Node pertama setelah root (\$) mengacu pada struktur dokumen alur kerja, seperti `stepOutputs`, atau dalam kasus variabel sistem Image Builder, `imageBuilder`. Daftar berikut berisi node struktur dokumen alur kerja JSONPath yang didukung.

Node struktur dokumen

- `parameter` - Parameter alur kerja
- `StepOutputs` - Output dari langkah dalam dokumen alur kerja yang sama
- `WorkflowOutputs` - Output dari dokumen alur kerja yang sudah berjalan
- `imagebuilder` - variabel sistem Image Builder

Node struktur `parameters` dan `stepOutputs` dokumen menyertakan node opsional untuk nama langkah. Ini membantu memastikan nama variabel unik di semua langkah.

Node terakhir di JsonPath adalah nama variabel target, seperti `instanceId`

Setiap langkah dapat merujuk ke output dari setiap tindakan langkah sebelumnya dengan variabel dinamis JSONPath ini. Ini juga dikenal sebagai rantai, atau referensi. Untuk merujuk ke output dari tindakan langkah sebelumnya, Anda dapat menggunakan variabel dinamis berikut.

```
$.stepOutputs.step-name.output-name
```

Contoh

```
- name: ApplyTestComponents
  action: ExecuteComponents
  onFailure: Abort
  inputs:
```



```
instanceId.$: "$.stepOutputs.LaunchTestInstance.instanceId"
```

Gunakan variabel sistem Image Builder

Image Builder menyediakan variabel sistem berikut yang dapat Anda gunakan dalam dokumen alur kerja Anda:

Nama variabel	Deskripsi	Tipe	Nilai contoh
cloudWatchLogKelompok	Nama grup CloudWatch Log untuk log keluaran. Format: /aws/imagebuilder/ <i><recipe-name></i>	String	/aws/imagebuilder/ <i>sampleImageRecipe</i>
cloudWatchLogStreaming	Nama aliran CloudWatch Log untuk log keluaran.	String	<i>1.0.0/1</i>
collectImageMetadata	Pengaturan yang mengarahkan Image Builder apakah akan mengumpulkan metadata instance.	Boolean	true false
collectImageScanTemuan	Nilai setelan saat ini yang memungkinkan Image Builder mengumpulkan temuan pemindaian gambar.	Boolean	true false
imageBuildNumber		Bulat	<i>1</i>

Nama variabel	Deskripsi	Tipe	Nilai contoh
	Nomor versi build gambar.		
imageId	Id AMI dari gambar dasar.	String	<i>ami-1234567890abcdef1</i>
imageName	Nama gambar.	String	<i>SampleImage</i>
imageType	Jenis output gambar.	String	AMI Docker
imageVersionNumber	Nomor versi gambar.	String	<i>1.0.0</i>
instanceProfileName	Nama peran profil instance yang digunakan Image Builder untuk meluncurkan instance build dan test.	String	<i>SampleImageBuilderInstanceProfileRole</i>
platform	Platform sistem operasi dari gambar yang dibangun.	String	Linux Windows MacOS
S3log	Objek JSON yang berisi konfigurasi untuk log S3 yang ditulis Image Builder.	Objek JSON	<i>{'s3logs': {'s3': 'sample-bucket', 'BucketName': 'sample-bucket', 'KeyPrefix': 'ib-logs'}}</i>

Nama variabel	Deskripsi	Tipe	Nilai contoh
securityGroups	ID grup keamanan yang berlaku untuk membangun dan menguji instance.	Daftar [String]	<i>[sg-1234567890abcd ef1, sg-11112222333344445]</i>
SourceImagearn	Nama Sumber Daya Amazon (ARN) dari sumber daya image Image Builder yang digunakan alur kerja untuk tahap pembuatan dan pengujian.	String	<i>arn:aws:imagebuilder:us-east-1:111122223333:image/sampleImage/1.0.0/1</i>
subnetId	ID subnet untuk meluncurkan instance build dan test ke dalam.	String	<i>subnet-1234567890abcdef1</i>
terminateInstanceOnKegagalan	Nilai setelan saat ini yang mengarahkan Image Builder untuk menghentikan instance saat gagal atau menyimpannya untuk pemecahan masalah.	Boolean	true false
WorkflowPhase	Tahap saat ini yang berjalan untuk eksekusi alur kerja.	String	Build Test

Nama variabel	Deskripsi	Tipe	Nilai contoh
<code>workingDirectory</code>	Jalur ke direktori kerja.	String	<code>/tmp</code>

Gunakan pernyataan bersyarat dalam langkah alur kerja Anda

Pernyataan bersyarat dimulai dengan atribut dokumen `if` pernyataan. Tujuan akhir dari `if` pernyataan ini adalah untuk menentukan apakah akan menjalankan tindakan langkah atau melewatinya. Jika `if` pernyataan diselesaikan `true`, maka tindakan langkah berjalan. Jika diselesaikan `false`, Image Builder melewatkan tindakan langkah dan mencatat status langkah `SKIPPED` di log.

`if` Pernyataan ini mendukung pernyataan percabangan (`and`, `or`) dan pengubah bersyarat (`!`). `not` Ini juga mendukung pernyataan bersyarat berikut yang melakukan perbandingan nilai (sama, kurang dari, lebih besar dari) berdasarkan tipe data yang dibandingkannya (string atau angka).

Pernyataan bersyarat yang didukung

- `booleanEquals`
- `numberEquals`
- `numberGreaterThan`
- `numberGreaterThanEquals`
- `numberLessThan`
- `numberLessThanEquals`
- `stringEquals`

Aturan untuk pernyataan percabangan dan pengubah bersyarat

Aturan berikut berlaku untuk pernyataan percabangan (`and`, `or`) dan pengubah bersyarat (`!`). `not`

- Pernyataan percabangan dan pengubah bersyarat harus muncul pada baris sendiri.
- Pernyataan percabangan dan pengubah bersyarat harus mengikuti aturan level.
 - Hanya ada satu pernyataan di tingkat induk.

- Setiap cabang anak atau pengubah memulai level baru.

Untuk informasi lebih lanjut tentang level, lihat [Tingkat bersarang](#).

- Setiap pernyataan percabangan harus memiliki setidaknya satu pernyataan bersyarat anak, tetapi tidak lebih dari sepuluh.
- Pengubah bersyarat beroperasi hanya pada satu pernyataan bersyarat anak.

Tingkat bersarang

Pernyataan bersyarat beroperasi pada beberapa tingkatan di bagian mereka sendiri. Misalnya, atribut `if` pernyataan muncul pada tingkat yang sama dalam dokumen alur kerja Anda sebagai nama langkah dan tindakan. Ini adalah dasar dari pernyataan bersyarat.

Anda dapat menentukan hingga empat tingkat pernyataan bersyarat, tetapi hanya satu pernyataan yang dapat muncul di tingkat induk. Semua pernyataan percabangan lainnya, pengubah bersyarat, atau operator bersyarat diindentasi dari sana, satu indentasi per level.

Garis besar berikut menunjukkan jumlah maksimum level bersarang untuk pernyataan bersyarat.

```
base:
  parent:
    - child (level 2)
      - child (level 3)
        child (level 4)
```

ifatribut

`ifAtribut` menentukan pernyataan bersyarat sebagai atribut dokumen. Ini adalah level nol.

Tingkat orang tua

Ini adalah tingkat pertama bersarang untuk pernyataan bersyarat. Hanya ada satu pernyataan di level ini. Jika Anda tidak memerlukan percabangan atau pengubah, ini bisa menjadi operator bersyarat tanpa pernyataan anak. Level ini tidak menggunakan notasi dasbor, kecuali untuk operator bersyarat.

Tingkat anak

Tingkat dua hingga empat dianggap tingkat anak. Pernyataan anak dapat mencakup pernyataan percabangan, pengubah bersyarat, atau operator bersyarat.

Contoh: Level bersarang

Contoh berikut menunjukkan jumlah maksimum level dalam pernyataan bersyarat.

```
if:
  and:
    #first level
    - stringEquals: 'my_string' #second level
      value: 'my_string'
    - and:
      #also second level
      - numberEquals: '1' #third level
        value: 1
      - not:
        #also third level
        stringEquals: 'second_string' #fourth level
        value: "diff_string"
```

Aturan bersarang

- Setiap cabang atau pengubah di tingkat anak memulai level baru.
- Setiap level menjorok.
- Ada maksimal empat level, termasuk satu pernyataan, pengubah, atau operator di tingkat induk, dan hingga tiga level tambahan.

Contoh

Kelompok contoh ini menunjukkan berbagai aspek pernyataan bersyarat.

Percabangan: dan

Pernyataan `and` percabangan beroperasi pada daftar ekspresi yang merupakan anak-anak dari cabang, yang semuanya harus dievaluasi. `true` Image Builder mengevaluasi ekspresi dalam urutan yang muncul dalam daftar. Jika ada ekspresi yang dievaluasi `false`, maka pemrosesan berhenti dan cabang dipertimbangkan `false`.

Contoh berikut mengevaluasi `true`, karena kedua ekspresi mengevaluasi `true`

```
if:
  and:
    - stringEquals: 'test_string'
      value: 'test_string'
    - numberEquals: 1
```

```
value: 1
```

Percabangan: atau

Pernyataan `or` percabangan beroperasi pada daftar ekspresi yang merupakan anak-anak dari cabang, setidaknya salah satunya harus dievaluasi. `true` Image Builder mengevaluasi ekspresi dalam urutan yang muncul dalam daftar. Jika ada ekspresi yang dievaluasi `true`, maka pemrosesan berhenti dan cabang dipertimbangkan `true`.

Contoh berikut mengevaluasi `true`, meskipun ekspresi pertama adalah `false`.

```
if:
  or:
    - stringEquals: 'test_string'
      value: 'test_string_not_equal'
    - numberEquals: 1
      value: 1
```

Pengubah bersyarat: tidak

Pengubah `not` bersyarat meniadakan pernyataan bersyarat yang merupakan turunan dari cabang.

Contoh berikut mengevaluasi `true` kapan `not` pengubah meniadakan pernyataan bersyarat. `stringEquals`

```
if:
  not:
    - stringEquals: 'test_string'
      value: 'test_string_not_equal'
```

Pernyataan bersyarat: `BooleanEquals`

Pernyataan `booleanEquals` kondisional membandingkan nilai Boolean dan mengembalikan nilai `true` jika nilai Boolean sama persis.

Contoh berikut menentukan apakah `collectImageScanFindings` diaktifkan.

```
if:
  - booleanEquals: true
    value: '$.imagebuilder.collectImageScanFindings'
```

Pernyataan bersyarat: StringEquals

Pernyataan `stringEquals` kondisional membandingkan dua string dan mengembalikan `true` jika string adalah sama persis. Jika salah satu nilai bukan string, Image Builder mengubahnya menjadi string sebelum dibandingkan.

Contoh berikut membandingkan variabel sistem platform untuk menentukan apakah alur kerja berjalan pada platform Linux.

```
if:
  - stringEquals: 'Linux'
    value: '$.imagebuilder.Platform'
```

Pernyataan bersyarat: NumberEquals

Pernyataan `numberEquals` kondisional membandingkan dua angka dan mengembalikan nilai `true` jika angkanya sama. Angka-angka untuk membandingkan harus menjadi salah satu format berikut.

- Bulat
- Desimal
- String yang cocok dengan pola regex berikut: `^-?[0-9]+(\.)?[0-9]+$`

Contoh berikut `comparisonx` semua mengevaluasi untuk `true`

```
if:
  # Value provider as a number
  numberEquals: 1
  value: '1'

  # Comparison value provided as a string
  numberEquals: '1'
  value: 1

  # Value provided as a string
  numberEquals: 1
  value: '1'

  # Floats are supported
  numberEquals: 5.0
  value: 5.0
```



```
# Negative values are supported
numberEquals: -1
value: -1
```

Impor dan ekspor gambar mesin virtual (VM) dengan EC2 Image Builder

Saat Anda mengekspor VM Anda dari lingkungan virtualisasi, proses itu membuat satu set file wadah disk atau lebih yang bertindak sebagai snapshot dari lingkungan, pengaturan, dan data VM Anda. Anda dapat menggunakan file-file ini untuk mengimpor VM Anda, dan menggunakannya sebagai gambar dasar untuk resep gambar Anda.

Image Builder mendukung format file berikut untuk wadah disk VM Anda:

- Buka Arsip Virtualisasi (OVA)
- Disk Mesin Virtual (VMDK)
- Hard Disk Virtual (VHD/VHDX)
- Mentah

Impor menggunakan disk untuk membuat Amazon Machine Image (AMI) dan sumber daya gambar Image Builder, yang keduanya dapat berfungsi sebagai gambar dasar untuk resep gambar kustom Anda. Disk VM harus disimpan dalam ember S3 untuk impor. Atau, Anda dapat mengimpor dari snapshot EBS yang ada.

Di konsol Image Builder, Anda dapat mengimpor gambar secara langsung, lalu menggunakan gambar keluaran atau AMI di resep Anda, atau Anda dapat menentukan parameter impor saat membuat versi resep atau resep. Untuk informasi selengkapnya tentang mengimpor secara langsung, lihat [Impor VM \(konsol\)](#). Untuk informasi selengkapnya tentang mengimpor sebagai bagian dari resep gambar Anda, lihat [Konfigurasi impor VM](#).

Impor VM ke Image Builder () AWS CLI

Untuk mengimpor VM dari disk ke AMI dan membuat sumber daya gambar Image Builder yang dapat Anda referensikan segera, ikuti langkah-langkah berikut dari: AWS CLI

1. Memulai impor VM, dengan perintah Impor/Ekspor VM `import-image` Amazon EC2 di. AWS CLI Catat ID tugas yang dikembalikan dalam respons perintah. Anda akan membutuhkannya

untuk langkah selanjutnya. Untuk informasi selengkapnya, lihat [Mengimpor VM sebagai gambar menggunakan Impor/Ekspor VM di Panduan Pengguna Impor/Ekspor VM](#).

2. Buat file JSON input CLI

Untuk merampingkan import-vm-image perintah Image Builder yang digunakan dalam AWS CLI, kita membuat file JSON yang berisi semua konfigurasi impor yang ingin kita masukkan ke dalam perintah.

Note

Konvensi penamaan untuk nilai data dalam file JSON mengikuti pola yang ditentukan untuk parameter permintaan tindakan Image Builder API. Untuk meninjau parameter permintaan perintah API, lihat [ImportVmImage](#) perintah di Referensi API EC2 Image Builder.

Untuk memberikan nilai data sebagai parameter baris perintah, lihat nama parameter yang ditentukan dalam Referensi AWS CLI Perintah. ke import-vm-image perintah Image Builder sebagai opsi.

Berikut adalah ringkasan parameter yang kami tentukan dalam contoh ini:

- name (string, required) - Nama untuk sumber daya gambar Image Builder untuk dibuat sebagai output dari impor.
- SemanticVersion <major>(string, required) - Versi semantik untuk gambar keluaran yang menentukan versi dalam format berikut, dengan nilai numerik di setiap posisi untuk menunjukkan versi tertentu: <minor>. <patch>. Sebagai contoh, 1.0.0. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. [Versi semantik](#)
- deskripsi (string) — Deskripsi resep gambar.
- platform (string, diperlukan) — Platform sistem operasi untuk VM yang diimpor.
- vmImportTaskId (string, required) - The ImportTaskId (AWS CLI) dari proses impor Amazon EC2 VM. Image Builder memantau proses impor untuk menarik AMI yang dibuatnya dan membangun sumber daya gambar Image Builder yang dapat langsung digunakan dalam resep.

- ClientToken (string, required) - Pengidentifikasi unik dan peka huruf besar/kecil yang Anda berikan untuk memastikan idempotensi permintaan. Untuk informasi selengkapnya, lihat [Memastikan idempotensi di Referensi](#) API Amazon EC2.
- tag (peta string) — Tag adalah pasangan nilai kunci yang dilampirkan ke sumber daya impor. Hingga 50 pasangan nilai kunci diperbolehkan.

Simpan file sebagai `import-vm-image.json`, untuk digunakan dalam `import-vm-image` perintah Image Builder.

```
{
  "name": "example-request",
  "semanticVersion": "1.0.0",
  "description": "vm-import-test",
  "platform": "Linux",
  "vmImportTaskId": "import-ami-01ab234567890cd1e",
  "clientToken": "asz1231231234cs3z",
  "tags": {
    "Usage": "VMIE"
  }
}
```

3. Impor gambar

Jalankan [import-vm-image](#) perintah, dengan file yang Anda buat sebagai input:

```
aws imagebuilder import-vm-image --cli-input-json file://import-vm-image.json
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

Mendistribusikan disk VM dari image build () Anda AWS CLI

Anda dapat mengatur distribusi file format disk VM yang didukung ke bucket S3 di Wilayah target sebagai bagian dari proses pembuatan gambar biasa, menggunakan konfigurasi distribusi Image Builder di file. AWS CLI Untuk informasi selengkapnya, lihat [Buat pengaturan distribusi untuk output VM disk \(\) AWS CLI](#).

Bagikan sumber daya EC2 Image Builder

EC2 Image Builder terintegrasi AWS Resource Access Manager dengan AWS RAM () untuk memungkinkan Anda berbagi sumber daya tertentu dengan Akun AWS salah satu atau melalui. AWS Organizations Sumber daya EC2 Image Builder yang dapat dibagikan adalah:

- Komponen-komponen
- Citra
- Resep

Bagian ini memberikan informasi untuk membantu Anda membagikan sumber daya EC2 Image Builder ini.

Isi bagian

- [Bekerja dengan komponen, gambar, dan resep bersama di EC2 Image Builder](#)
- [Prasyarat untuk berbagi komponen, gambar, dan resep](#)
- [Layanan-layanan terkait](#)
- [Berbagi di seluruh Wilayah](#)
- [Berbagi komponen, gambar, atau resep](#)
- [Membatalkan berbagi komponen, gambar, atau resep bersama](#)
- [Mengidentifikasi komponen, gambar, atau resep bersama](#)
- [Izin komponen, gambar, dan resep bersama](#)
- [Tagihan dan pengukuran](#)
- [Batas sumber daya](#)

Bekerja dengan komponen, gambar, dan resep bersama di EC2 Image Builder

Komponen, gambar, dan berbagi resep memungkinkan pemilik sumber daya untuk berbagi konfigurasi perangkat lunak dengan orang lain Akun AWS atau dalam AWS organisasi. Anda dapat mengelola berbagi sumber daya secara terpusat, dan menentukan satu set akun yang dapat digunakan untuk berbagi konfigurasi.

Dalam model ini, Akun AWS yang memiliki komponen, gambar, atau resep (pemilik) membagikannya dengan yang lain Akun AWS (konsumen). Konsumen dapat mengaitkan komponen bersama dengan pipeline gambar mereka untuk secara otomatis menggunakan pembaruan ke komponen, gambar, atau resep bersama.

Pemilik komponen, gambar, atau resep dapat berbagi sumber daya ini dengan:

- Khusus Akun AWS di dalam atau di luar organisasinya di AWS Organizations.
- Unit organisasi (OU) di dalam organisasinya di AWS Organizations.
- Seluruh organisasinya di AWS Organizations
- AWS Organizations atau OU di luar organisasinya.

Prasyarat untuk berbagi komponen, gambar, dan resep

Untuk berbagi komponen, gambar, atau resep Image Builder:

- Anda harus memiliki komponen, gambar, atau resep di Akun AWS. Anda tidak dapat berbagi sumber daya yang telah dibagikan dengan Anda.
- Kunci AWS Key Management Service (AWS KMS) yang terkait dengan sumber daya terenkripsi harus dibagikan secara eksplisit dengan akun target, organisasi, atau OU.
- Untuk berbagi sumber daya Image Builder dengan AWS Organizations dan OU AWS RAM, Anda harus mengaktifkan berbagi. Untuk informasi selengkapnya, lihat [Aktifkan Berbagi dengan AWS Organizations](#) dalam Panduan Pengguna AWS RAM.
- Jika Anda mendistribusikan gambar yang dienkripsi dengan AWS KMS seluruh akun di Wilayah yang berbeda, Anda harus membuat kunci KMS dan alias di setiap Wilayah target. Selain itu, orang-orang yang akan meluncurkan instans di Wilayah tersebut akan memerlukan akses ke kunci KMS yang ditentukan melalui Kebijakan Utama.

Sumber daya berikut yang dibuat Image Builder dari build pipeline Anda tidak dianggap sebagai sumber daya Image Builder, melainkan sumber daya eksternal yang didistribusikan Image Builder di akun Anda, dan keWilayah AWS, akun, dan organisasi atau unit organisasi (OU) yang Anda tentukan dalam konfigurasi distribusi.

- Amazon Machine Image (AMI)
- Gambar kontainer yang berada di Amazon ECR

Untuk informasi selengkapnya tentang setelan distribusi untuk AMI Anda, lihat [Membuat dan memperbarui konfigurasi distribusi AMI](#). Untuk informasi selengkapnya tentang setelan distribusi untuk gambar kontainer Anda di Amazon ECR, lihat [Membuat dan memperbarui pengaturan distribusi untuk gambar kontainer](#).

Untuk informasi selengkapnya tentang berbagi AMI dengan AWS Organizations dan OU, lihat [Berbagi AMI dengan organisasi atau OU](#).

Layanan-layanan terkait

AWS Resource Access Manager

Komponen, gambar, dan berbagi resep terintegrasi dengan AWS Resource Access Manager (AWS RAM). AWS RAM adalah layanan yang memungkinkan Anda untuk berbagi AWS sumber daya Anda dengan AWS akun apa pun atau melalui AWS Organizations. Dengan AWS RAM, Anda berbagi sumber daya yang Anda miliki dengan membuat pembagian sumber daya. Pembagian sumber daya menentukan sumber daya untuk dibagikan dan konsumen yang akan dibagikan. Konsumen dapat berupa individu akun AWS, unit organisasi, atau seluruh organisasi di AWS Organizations.

Untuk informasi selengkapnya tentang AWS RAM, lihat [Panduan Pengguna AWS RAM](#).

Berbagi di seluruh Wilayah

Komponen, gambar, dan resep bersama hanya dapat dibagikan di AWS Wilayah tertentu. Ketika Anda berbagi sumber daya ini, mereka tidak akan mereplikasi di seluruh Wilayah.

Berbagi komponen, gambar, atau resep

Untuk membagikan komponen, gambar, atau resep Image Builder, Anda harus menambahkannya ke pembagian sumber daya. Berbagi sumber daya adalah sumber daya AWS RAM yang memungkinkan

Anda berbagi sumber daya di seluruh akun AWS. Pembagian sumber daya menentukan sumber daya untuk dibagikan dan konsumen dengan siapa mereka dibagikan. Untuk menambahkan komponen, gambar, atau resep ke pembagian sumber daya baru, Anda harus terlebih dahulu membuat pembagian sumber daya menggunakan AWS RAM konsol.

Jika Anda adalah bagian dari organisasi AWS Organizations dan berbagi dalam organisasi Anda diaktifkan, konsumen di organisasi Anda secara otomatis diberikan akses ke komponen, gambar, atau resep bersama. Jika tidak, konsumen menerima undangan untuk bergabung dengan pembagian sumber daya dan diberikan akses ke sumber daya bersama setelah menerima undangan.

Opsi berikut tersedia untuk berbagi sumber daya Anda:

Opsi 1: Buat berbagi sumber daya RAM

Saat Anda membuat pembagian sumber daya RAM, Anda dapat membagikan komponen, gambar, atau resep yang Anda miliki dalam satu langkah. Gunakan salah satu metode berikut untuk membuat pembagian sumber daya Anda:

- Konsol

Untuk membuat pembagian sumber daya menggunakan AWS RAM konsol, lihat [Berbagi AWS sumber daya yang dimiliki oleh Anda](#) di Panduan AWS RAM Pengguna.

- AWS CLI

Untuk membuat pembagian sumber daya Anda menggunakan antarmuka baris AWS RAM perintah, jalankan [create-resource-share](#) perintah di fileAWS CLI.

Opsi 2: Terapkan kebijakan sumber daya dan promosikan ke pembagian sumber daya RAM

Opsi kedua untuk berbagi sumber daya Anda melibatkan dua langkah, menjalankan perintah di AWS CLI untuk keduanya. Langkah pertama menggunakan perintah Image Builder AWS CLI untuk menerapkan kebijakan berbasis sumber daya ke sumber daya bersama. Langkah kedua mempromosikan sumber daya ke berbagi sumber daya RAM menggunakan [promote-resource-share-created-from-policy](#) AWS RAM perintah di AWS CLI untuk memastikan bahwa sumber daya terlihat oleh semua kepala sekolah dengan siapa Anda telah membagikannya.

1. Menerapkan kebijakan sumber daya

Agar berhasil menerapkan kebijakan sumber daya, Anda harus memastikan bahwa akun yang Anda bagikan memiliki izin untuk mengakses sumber daya apa pun yang mendasarinya.

Pilih tab yang cocok dengan jenis sumber daya Anda untuk perintah yang berlaku.

Image

Anda dapat menerapkan kebijakan sumber daya ke gambar, untuk memungkinkan orang lain menggunakannya sebagai gambar dasar dalam resep mereka.

Jalankan perintah [put-image-policy](#) Image Builder diAWS CLI, untuk mengidentifikasi AWS prinsipal untuk berbagi gambar dengan.

```
aws imagebuilder put-image-policy --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-image/2019.12.03/1 --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action": [ "imagebuilder:GetImage", "imagebuilder:ListImages" ], "Resource": [ "arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-image/2019.12.03/1" ] } ] }'
```

Component

Anda dapat menerapkan kebijakan sumber daya ke komponen build atau pengujian untuk mengaktifkan berbagi lintas akun. Perintah ini memberikan izin akun lain untuk menggunakan komponen Anda dalam resep mereka. Agar berhasil menerapkan kebijakan sumber daya, Anda harus memastikan bahwa akun yang Anda bagikan memiliki izin untuk mengakses sumber daya apa pun yang direferensikan oleh komponen bersama, seperti file yang dihosting di repositori pribadi.

Jalankan perintah [put-component-policy](#) Image Builder diAWS CLI, untuk mengidentifikasi AWS prinsipal untuk berbagi komponen dengan.

```
aws imagebuilder put-component-policy --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-component/2019.12.03/1 --policy '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action": [ "imagebuilder:GetComponent", "imagebuilder:ListComponents" ], "Resource": [ "arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-component/2019.12.03/1" ] } ] }'
```


Image recipe

Anda dapat menerapkan kebijakan sumber daya ke resep gambar untuk mengaktifkan berbagi lintas akun. Perintah ini memberikan izin akun lain untuk menggunakan resep Anda untuk membuat gambar di akun mereka. Agar berhasil menerapkan kebijakan sumber daya, Anda harus memastikan bahwa akun yang Anda bagikan memiliki izin untuk mengakses sumber daya apa pun yang direferensikan resep, seperti gambar dasar, atau komponen yang dipilih.

Jalankan perintah [put-image-recipe-policy](#) Image Builder diAWS CLI, untuk mengidentifikasi AWS prinsipal untuk berbagi gambar dengan.

```
aws imagebuilder put-image-recipe-policy --image-recipe-arn
arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-
image-recipe/2019.12.03 --policy '{ "Version": "2012-10-17", "Statement":
[ { "Effect": "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":
[ "imagebuilder:GetImageRecipe", "imagebuilder:ListImageRecipes" ], "Resource":
[ "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-image-
recipe/2019.12.03" ] } ] }'
```

Container recipe

Anda dapat menerapkan kebijakan sumber daya ke resep kontainer untuk mengaktifkan berbagi lintas akun. Perintah ini memberikan izin akun lain untuk menggunakan resep Anda untuk membuat gambar di akun mereka. Agar berhasil menerapkan kebijakan sumber daya, Anda harus memastikan bahwa akun yang Anda bagikan memiliki izin untuk mengakses sumber daya apa pun yang direferensikan resep, seperti gambar dasar, atau komponen yang dipilih.

Jalankan perintah [put-container-recipe-policy](#) Image Builder diAWS CLI, untuk mengidentifikasi AWS prinsipal untuk berbagi gambar dengan.

```
aws imagebuilder put-container-recipe-policy --container-recipe-arn
arn:aws:imagebuilder:us-west-2:123456789012:container-recipe/my-example-
container-recipe/2021.12.03 --policy '{ "Version": "2012-10-17", "Statement":
[ { "Effect": "Allow", "Principal": { "AWS": [ "123456789012" ] }, "Action":
[ "imagebuilder:GetContainerRecipe", "imagebuilder:ListContainerRecipes" ],
"Resource": [ "arn:aws:imagebuilder:us-west-2:123456789012:container-recipe/my-
example-container-recipe/2021.12.03" ] } ] }'
```

Note

Untuk menetapkan kebijakan yang benar untuk berbagi dan membatalkan berbagi sumber daya, pemilik sumber daya harus memiliki `imagebuilder:put*` izin.

2. Promosikan sebagai pembagian sumber daya RAM

Untuk memastikan bahwa sumber daya terlihat oleh semua prinsipal dengan siapa Anda telah berbagi, jalankan [promote-resource-share-created-from-policy](#) AWS RAM perintah di. AWS CLI

Membatalkan berbagi komponen, gambar, atau resep bersama

Untuk membatalkan berbagi komponen, gambar, atau resep bersama yang Anda miliki, Anda harus menghapusnya dari pembagian sumber daya. Anda dapat melakukan ini menggunakan AWS Resource Access Manager konsol atau AWS CLI.

Note

Untuk membatalkan pembagian komponen, gambar, atau resep, konsumen tidak dapat memiliki ketergantungan apa pun padanya. Konsumen harus menghapus dependensi apa pun pada sumber daya bersama sebelum pemilik dapat membatalkan pembagiannya.

Untuk membatalkan berbagi komponen, gambar, atau resep bersama yang Anda miliki menggunakan konsol AWS Resource Access Manager

Lihat [Memperbarui Sumber Daya Bersama](#) di Panduan Pengguna AWS RAM.

Untuk membatalkan berbagi komponen, gambar, atau resep bersama yang Anda miliki menggunakan AWS CLI

Gunakan [disassociate-resource-share](#) perintah untuk berhenti berbagi sumber daya.

Mengidentifikasi komponen, gambar, atau resep bersama

Pemilik dan konsumen dapat mengidentifikasi komponen bersama, gambar, dan resep gambar menggunakan perintah Image Builder di file AWS CLI.

Identifikasi komponen bersama

Jalankan perintah [list-components](#) untuk mendapatkan daftar komponen yang Anda miliki dan komponen yang dibagikan dengan Anda. Perintah [get-component](#) menunjukkan Akun AWS ID pemilik komponen.

Identifikasi gambar bersama

Jalankan perintah [list-images](#) untuk mendapatkan daftar gambar yang Anda miliki dan gambar yang dibagikan dengan Anda. Perintah [get-image](#) menunjukkan Akun AWS ID pemilik gambar.

Identifikasi gambar kontainer bersama

Jalankan perintah [list-images](#) untuk mendapatkan daftar gambar yang Anda miliki dan gambar yang dibagikan dengan Anda. Perintah [get-image](#) menunjukkan Akun AWS ID pemilik gambar.

Identifikasi resep gambar bersama

Jalankan [list-image-recipes](#) perintah untuk mendapatkan daftar resep gambar yang Anda miliki dan resep gambar yang dibagikan dengan Anda. [get-image-recipe](#) Perintah menunjukkan Akun AWS ID pemilik resep gambar.

Identifikasi resep wadah bersama

Jalankan [list-container-recipes](#) perintah untuk mendapatkan daftar resep wadah yang Anda miliki dan resep wadah yang dibagikan dengan Anda. [get-container-recipe](#) Perintah menunjukkan Akun AWS ID pemilik resep kontainer.

Izin komponen, gambar, dan resep bersama

Izin untuk pemilik

Pemilik tidak dapat menghapus resep komponen, gambar, atau gambar bersama hingga tidak lagi dibagikan. Pemilik tidak dapat membatalkan pembagian sumber daya ini sampai tidak ada konsumen yang bergantung padanya.

Izin untuk konsumen

Konsumen dapat membaca komponen, gambar, atau resep gambar, tetapi tidak dapat memodifikasinya dengan cara apa pun. Mereka tidak dapat melihat atau memodifikasi sumber daya ini jika dimiliki oleh konsumen lain atau pemilik sumber daya. Konsumen dapat menggunakan komponen dan gambar bersama dalam resep gambar untuk membuat gambar khusus. Konsumen dapat menggunakan resep gambar bersama untuk membuat gambar kustom mereka sendiri.

Tagihan dan pengukuran

Tidak ada biaya untuk menggunakan EC2 Image Builder.

Batas sumber daya

Komponen bersama, gambar, dan resep gambar dihitung terhadap batas sumber daya yang sesuai dari pemilik saja. Batas sumber daya konsumen tidak terpengaruh oleh sumber daya yang telah dibagikan dengan mereka.

Menandai sumber daya EC2 Image Builder

Menandai sumber daya Anda dapat berguna untuk memfilter dan melacak biaya sumber daya, atau kategori lainnya. Anda juga dapat mengontrol akses berdasarkan tag. Untuk informasi selengkapnya tentang otorisasi berbasis tag, lihat [Otorisasi berdasarkan tag Image Builder](#)

Image Builder mendukung tag dinamis berikut:

- - `{{imagebuilder:buildDate}}`

Menyelesaikan ke tanggal/waktu pembuatan pada waktu pembuatan.

- - `{{imagebuilder:buildVersion}}`

Menyelesaikan ke versi build, yang merupakan nomor yang terletak di akhir Image Builder ARN. Misalnya, "arn:aws:imagebuilder:us-west-2:123456789012:component/myexample-component/2019.12.02/1" menampilkan versi build sebagai 1.

Daftar Isi

- [Menandai sumber daya \(AWS CLI\)](#)
- [Membatalkan tag sumber daya \(\) AWS CLI](#)
- [Daftar semua tag untuk sumber daya tertentu \(AWS CLI\)](#)

Menandai sumber daya (AWS CLI)

Contoh berikut menunjukkan cara menggunakan perintah imagebuilder CLI untuk menambahkan dan menandai sumber daya di EC2 Image Builder. Anda harus memberikan `resourceArn` dan tag untuk diterapkan padanya.

Contoh `tag-resource.json` isi adalah sebagai berikut:

```
{
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-  
example-pipeline",
  "tags": {
    "KeyName": "KeyValue"
  }
}
```

Jalankan perintah berikut, yang mereferensikan file sebelumnya. `tag-resource.json`

```
aws imagebuilder tag-resource --cli-input-json file://tag-resource.json
```

Membatalkan tag sumber daya () AWS CLI

Contoh berikut menunjukkan cara menggunakan perintah `imagebuilder CLI` untuk menghapus tag dari sumber daya. Anda harus memberikan `resourceArn` dan kunci untuk menghapus tag.

Contoh `untag-resource.json` isi adalah sebagai berikut:

```
{
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-  
example-pipeline",
  "tagKeys": [
    "KeyName"
  ]
}
```

Jalankan perintah berikut, yang mereferensikan file sebelumnya. `untag-resource.json`

```
aws imagebuilder untag-resource --cli-input-json file://untag-resource.json
```

Daftar semua tag untuk sumber daya tertentu (AWS CLI)

Contoh berikut menunjukkan cara menggunakan perintah `imagebuilder CLI` untuk daftar semua tag untuk sumber daya tertentu.

```
aws imagebuilder list-tags-for-resource --resource-arn arn:aws:imagebuilder:us-  
west-2:123456789012:image-pipeline/my-example-pipeline
```

Hapus sumber EC2 Image Builder

Lingkungan Image Builder Anda, seperti rumah Anda, membutuhkan perawatan rutin untuk membantu Anda menemukan apa yang Anda butuhkan, dan menyelesaikan tugas Anda tanpa mengarungi kekacauan. Pastikan untuk secara teratur membersihkan sumber daya sementara yang Anda buat untuk pengujian. Jika tidak, Anda mungkin lupa tentang sumber daya itu, dan kemudian, tidak ingat untuk apa sumber daya itu digunakan. Pada saat itu, mungkin tidak jelas apakah Anda dapat menyingkirkannya dengan aman.

Menghapus sumber daya tidak menghapus AMI Amazon EC2 atau image container Amazon ECR yang dibuat selama proses pembuatan gambar. Anda harus membersihkannya secara terpisah, menggunakan tindakan konsol Amazon EC2 atau Amazon ECR yang sesuai, atau API atau perintah AWS CLI

Tip

Untuk mencegah kesalahan ketergantungan saat Anda menghapus sumber daya, pastikan untuk menghapus sumber daya Anda dalam urutan berikut:

1. Pipa gambar
2. Resep gambar
3. Semua sumber daya yang tersisa

Hapus sumber daya menggunakan AWS Management Console

Untuk menghapus pipeline gambar dan sumber dayanya, ikuti langkah-langkah berikut:

Hapus pipa

1. Untuk melihat daftar pipeline build yang dibuat di bawah akun Anda, pilih Pipeline gambar dari panel navigasi.
2. Pilih kotak centang di samping nama Pipeline untuk memilih pipeline yang ingin Anda hapus.
3. Di bagian atas panel pipa gambar, pada menu Tindakan, pilih Hapus.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** di dalam kotak, dan pilih Hapus.

Hapus resepnya

1. Untuk melihat daftar resep yang dibuat di bawah akun Anda, pilih Resep gambar dari panel navigasi.
2. Pilih kotak centang di sebelah Nama resep untuk memilih resep yang ingin Anda hapus.
3. Di bagian atas panel Resep gambar, pada menu Tindakan, pilih Hapus resep.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** di dalam kotak, dan pilih Hapus.

Hapus konfigurasi infrastruktur

1. Untuk melihat daftar konfigurasi infrastruktur yang dibuat di bawah akun Anda, pilih Konfigurasi infrastruktur dari panel navigasi.
2. Pilih kotak centang di sebelah Nama konfigurasi untuk memilih konfigurasi infrastruktur yang ingin Anda hapus.
3. Di bagian atas panel konfigurasi Infrastruktur, pilih Hapus.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** di dalam kotak, dan pilih Hapus.

Hapus pengaturan distribusi

1. Untuk melihat daftar setelan distribusi yang dibuat di bawah akun Anda, pilih Pengaturan distribusi dari panel navigasi.
2. Pilih kotak centang di sebelah Nama konfigurasi untuk memilih pengaturan distribusi yang Anda buat untuk tutorial ini.
3. Di bagian atas panel Pengaturan distribusi, pilih Hapus.
4. Untuk mengonfirmasi penghapusan, masukkan **Delete** di dalam kotak, dan pilih Hapus.

Hapus citra

1. Untuk melihat daftar gambar yang dibuat di bawah akun Anda, pilih Gambar dari panel navigasi.
2. Pilih Versi gambar untuk gambar yang ingin Anda hapus. Ini membuka halaman Image build version.
3. Pilih kotak centang di sebelah Versi untuk gambar apa pun yang ingin Anda hapus. Anda dapat memilih lebih dari satu versi gambar sekaligus.
4. Di bagian atas panel Image build version, pilih Delete version.

5. Untuk mengonfirmasi penghapusan, masukkan **Delete** di dalam kotak, dan pilih Hapus.

Hapus pipeline gambar menggunakan AWS CLI

Contoh berikut menunjukkan cara menghapus sumber daya Image Builder menggunakan file AWS CLI. Seperti disebutkan sebelumnya, sumber daya harus dihapus dalam urutan berikut untuk menghindari kesalahan ketergantungan:

1. Pipa gambar
2. Resep gambar
3. Semua sumber daya yang tersisa

Hapus pipa gambar (AWS CLI)

Contoh berikut menunjukkan cara menghapus pipeline gambar dengan menentukan ARN nya.

```
aws imagebuilder delete-image-pipeline --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-example-pipeline
```

Hapus resep gambar (AWS CLI)

Contoh berikut menunjukkan cara menghapus resep gambar dengan menentukan ARN nya.

```
aws imagebuilder delete-image-recipe --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-recipe/2019.12.03
```

Hapus konfigurasi infrastruktur

Contoh berikut menunjukkan cara menghapus sumber daya konfigurasi infrastruktur dengan menentukan ARN nya.

```
aws imagebuilder delete-infrastructure-configuration --infrastructure-configuration-arn arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/my-example-infrastructure-configuration
```

Hapus pengaturan distribusi

Contoh berikut menunjukkan cara menghapus sumber daya pengaturan distribusi dengan menentukan ARN nya.


```
aws imagebuilder delete-distribution-configuration --distribution-configuration-arn
arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-example-
distribution-configuration
```

Hapus citra

Contoh berikut menunjukkan cara menghapus versi build gambar dengan menentukan ARN nya.

```
aws imagebuilder delete-image --image-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:image/my-example-image/2019.12.02/1
```

Hapus komponen

Contoh berikut menunjukkan cara menggunakan perintah imagebuilder CLI untuk menghapus versi build komponen dengan menentukan ARN-nya.

```
aws imagebuilder delete-component --component-build-version-arn
arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-
component/2019.12.02/1
```

Important

Pastikan tidak ada resep yang mereferensikan versi pembuatan komponen dengan cara apa pun sebelum Anda menghapusnya. Gagal melakukannya dapat menyebabkan kegagalan pipa.

Kelola pipeline EC2 Image Builder menggunakan konsol

Image Builder image pipelines menyediakan kerangka kerja otomatisasi untuk membuat dan memelihara AMI kustom dan gambar kontainer. Pipa memberikan fungsionalitas berikut:

- Merakit gambar dasar, komponen untuk membangun dan menguji, konfigurasi infrastruktur, dan pengaturan distribusi.
- Memfasilitasi penjadwalan untuk proses pemeliharaan otomatis menggunakan wizard `Schedule builder` di konsol, atau memasukkan ekspresi cron untuk pembaruan berulang pada gambar Anda.
- Aktifkan deteksi perubahan untuk gambar dasar dan komponen, untuk secara otomatis melewati build terjadwal saat tidak ada perubahan.
- Aktifkan otomatisasi berbasis aturan melalui Amazon. EventBridge

Note

Untuk informasi selengkapnya tentang menggunakan EventBridge API untuk melihat atau mengubah aturan, lihat [Referensi Amazon EventBridge API](#). Untuk informasi selengkapnya tentang penggunaan EventBridge events perintah dalam AWS CLI untuk melihat atau mengubah aturan, lihat [peristiwa](#) di Referensi AWS CLI Perintah.

Daftar Isi

- [Daftar dan lihat detail saluran pipa](#)
- [Buat dan perbarui pipeline gambar AMI](#)
- [Buat dan perbarui pipeline gambar kontainer](#)
- [Konfigurasi alur kerja gambar untuk pipeline EC2 Image Builder](#)
- [Jalankan pipeline gambar Anda](#)
- [Gunakan ekspresi cron di EC2 Image Builder](#)
- [Menggunakan EventBridge aturan dengan pipeline Image Builder](#)

Daftar dan lihat detail saluran pipa

Bagian ini menjelaskan berbagai cara agar Anda dapat menemukan informasi dan melihat detail untuk pipeline gambar EC2 Image Builder Anda.

Detail pipa

- [Daftar pipa gambar \(\) AWS CLI](#)
- [Dapatkan detail pipa gambar \(AWS CLI\)](#)

Daftar pipa gambar () AWS CLI

Contoh berikut menunjukkan cara menggunakan `list-image-pipelines` perintah di AWS CLI daftar semua pipeline gambar Anda.

```
aws imagebuilder list-image-pipelines
```

Dapatkan detail pipa gambar (AWS CLI)

Contoh berikut menunjukkan bagaimana menggunakan `get-image-pipeline` perintah di AWS CLI untuk mendapatkan rincian tentang pipa gambar melalui ARN nya.

```
aws imagebuilder get-image-pipeline --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-example-pipeline
```

Buat dan perbarui pipeline gambar AMI

Anda dapat mengatur, mengonfigurasi, dan mengelola pipeline image AMI dari konsol Image Builder, melalui Image Builder API, atau dengan `imagebuilder` perintah di AWS CLI file. Anda dapat menggunakan wizard Create image pipeline console untuk memandu Anda melalui langkah-langkah berikut:

- Tentukan detail pipeline seperti nama, deskripsi, dan tag sumber daya.
- Pilih resep gambar AMI yang menyertakan gambar dasar dari gambar dikelola mulai cepat, atau gambar yang Anda buat atau yang dibagikan dengan Anda. Resep ini juga mencakup komponen yang melakukan tugas berikut pada instans EC2 yang digunakan Image Builder untuk membangun image Anda:
 - Tambah dan hapus perangkat lunak

- Sesuaikan pengaturan dan skrip
- Jalankan tes yang dipilih
- Tentukan alur kerja untuk mengonfigurasi pembuatan gambar dan langkah-langkah pengujian yang dijalankan pipeline Anda.
- Tentukan konfigurasi infrastruktur untuk pipeline Anda dengan pengaturan default atau pengaturan yang Anda konfigurasikan sendiri. Konfigurasi mencakup jenis instans dan key pair yang akan digunakan untuk pengaturan gambar, keamanan dan jaringan, penyimpanan log dan pengaturan pemecahan masalah, dan notifikasi SNS.

Ini adalah langkah opsional. Image Builder menggunakan pengaturan default untuk konfigurasi infrastruktur Anda jika Anda tidak menentukan konfigurasi sendiri.

- Tentukan pengaturan distribusi untuk mengirimkan gambar Anda ke AWS Wilayah dan akun tujuan. Anda dapat menentukan kunci KMS untuk enkripsi, mengonfigurasi berbagi AMI atau konfigurasi lisensi, atau mengonfigurasi templat peluncuran untuk AMI yang Anda distribusikan.

Ini adalah langkah opsional. Jika Anda tidak menentukan konfigurasi sendiri, Image Builder menggunakan penamaan default untuk AMI keluaran Anda, dan mendistribusikan AMI ke Wilayah sumber. Wilayah sumber adalah Wilayah tempat Anda menjalankan pipa.

Untuk informasi selengkapnya dan step-by-step tutorial tentang penggunaan wizard Create image pipeline console dengan nilai default jika disediakan, lihat [Buat pipeline gambar menggunakan wizard konsol EC2 Image Builder](#).

Daftar Isi

- [Buat pipeline gambar AMI \(AWS CLI\)](#)
- [Perbarui saluran pipa gambar AMI \(konsol\)](#)
- [Perbarui saluran pipa gambar AMI \(\) AWS CLI](#)

Buat pipeline gambar AMI (AWS CLI)

Anda dapat membuat pipeline gambar AMI dengan file JSON yang berisi detail konfigurasi sebagai input ke create-image-pipeline perintah di AWS CLI file.

Seberapa sering pipeline Anda membuat gambar baru untuk menggabungkan pembaruan yang tertunda dari gambar dasar dan komponen Anda tergantung pada `schedule` yang telah Anda konfigurasikan. A `schedule` memiliki atribut berikut:

- `scheduleExpression`— Tetapkan jadwal kapan pipeline Anda berjalan untuk mengevaluasi `pipelineExecutionStartCondition` dan menentukan apakah itu harus memulai pembangunan. Jadwal dikonfigurasi dengan ekspresi cron. Untuk informasi selengkapnya tentang cara memformat ekspresi cron di Image Builder, lihat [Gunakan ekspresi cron di EC2 Image Builder](#).
- `pipelineExecutionStartCondition`— Menentukan apakah pipeline Anda harus memulai pembangunan. Nilai yang valid meliputi:
 - `EXPRESSION_MATCH_ONLY`— pipeline Anda akan membangun gambar baru setiap kali ekspresi cron cocok dengan waktu saat ini.
 - `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`— pipeline Anda tidak akan memulai pembuatan gambar baru kecuali ada perubahan yang tertunda pada image atau komponen dasar Anda.

Ketika Anda menjalankan `create-image-pipeline` perintah di AWS CLI, banyak sumber daya konfigurasi adalah opsional. Namun, beberapa sumber daya memiliki persyaratan bersyarat, tergantung pada jenis gambar apa yang dibuat pipa. Sumber daya berikut diperlukan untuk pipeline gambar AMI:

- Resep gambar ARN
- Konfigurasi infrastruktur ARN

1. Buat file JSON input CLI

Gunakan alat pengeditan file favorit Anda untuk membuat file JSON dengan kunci berikut, ditambah nilai yang valid untuk lingkungan Anda. Contoh ini menggunakan file bernama `create-image-pipeline.json`:

```
{
  "name": "MyWindows2019Pipeline",
  "description": "Builds Windows 2019 Images",
  "enhancedImageMetadataEnabled": true,
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-recipe/2020.12.03",
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/my-example-infrastructure-configuration",
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-example-distribution-configuration",
}
```

```
"imageTestsConfiguration": {
  "imageTestsEnabled": true,
  "timeoutMinutes": 60
},
"schedule": {
  "scheduleExpression": "cron(0 0 * * SUN *)",
  "pipelineExecutionStartCondition":
  "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
},
"status": "ENABLED"
}
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input.

```
aws imagebuilder create-image-pipeline --cli-input-json file://create-image-pipeline.json
```

Perbarui saluran pipa gambar AMI (konsol)

Setelah membuat pipeline image Image Builder untuk image AMI, Anda dapat membuat perubahan pada konfigurasi infrastruktur dan pengaturan distribusi dari konsol Image Builder.

Untuk memperbarui pipeline gambar dengan resep gambar baru, Anda harus menggunakan AWS CLI. Untuk informasi selengkapnya, silakan lihat [Perbarui saluran pipa gambar AMI \(\) AWS CLI](#) dalam panduan ini.

Pilih pipeline Image Builder yang sudah ada

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).

2. Untuk melihat daftar pipeline gambar yang dibuat di bawah akun Anda, pilih Pipeline gambar dari panel navigasi.

Note

Daftar pipeline gambar mencakup indikator untuk jenis gambar keluaran yang dibuat oleh pipeline — AMI atau Docker.

3. Untuk melihat detail atau mengedit pipeline, pilih tautan Nama saluran pipa. Ini membuka tampilan detail untuk pipa.

Note

Anda juga dapat memilih kotak centang di sebelah nama Pipeline, lalu pilih Lihat detail.

Detail saluran pipa

Halaman detail pipeline mencakup bagian-bagian berikut:

Ringkasan

Bagian di bagian atas halaman merangkum detail utama untuk pipeline yang terlihat dengan tab detail apa pun yang terbuka. Detail yang ditampilkan di bagian ini hanya dapat diedit pada tab detailnya masing-masing.

Tab detail

- Gambar keluaran - Menampilkan gambar keluaran yang dihasilkan pipa.
- Resep gambar — Menampilkan detail resep. Setelah Anda membuat resep, Anda tidak dapat mengeditnya. Anda harus membuat versi baru resep dari halaman Resep gambar di konsol Image Builder, atau dengan menggunakan perintah Image Builder diAWS CLI. Untuk informasi selengkapnya, lihat [Kelola resep](#).
- Konfigurasi infrastruktur — Menampilkan informasi yang dapat diedit untuk mengonfigurasi infrastruktur pipeline build Anda.
- Pengaturan distribusi - Menampilkan informasi yang dapat diedit untuk distribusi AMI.
- EventBridge rules — Untuk Event Bus yang dipilih, menampilkan EventBridge aturan yang menargetkan pipeline saat ini. Termasuk Buat bus acara dan Buat tindakan aturan yang ditautkan

ke EventBridge konsol. Untuk informasi selengkapnya tentang tab ini, lihat [Gunakan EventBridge aturan](#).

Edit konfigurasi infrastruktur untuk pipeline Anda

Konfigurasi infrastruktur mencakup detail berikut yang dapat Anda edit setelah membuat pipeline:

- Deskripsi untuk konfigurasi infrastruktur Anda.
- Peran IAM untuk dikaitkan dengan profil instance.
- AWS infrastruktur, termasuk jenis Instance dan topik SNS untuk notifikasi.
- VPC, subnet, dan grup keamanan.
- Pengaturan pemecahan masalah, termasuk Instance Terminate saat gagal, pasangan Kunci untuk menghubungkan, dan lokasi bucket S3 opsional untuk log instance.

Untuk mengedit konfigurasi infrastruktur dari halaman detail pipeline, ikuti langkah-langkah berikut:

1. Pilih tab konfigurasi Infrastruktur.
2. Pilih Edit dari sudut kanan atas panel detail Konfigurasi.
3. Ketika Anda siap untuk menyimpan pembaruan yang telah Anda buat untuk konfigurasi infrastruktur Anda, pilih Simpan perubahan.

Edit setelah distribusi untuk pipeline Anda

Pengaturan distribusi mencakup detail berikut yang dapat Anda edit setelah membuat pipeline:

- Deskripsi untuk konfigurasi distribusi Anda.
- Pengaturan wilayah untuk Wilayah tempat Anda mendistribusikan gambar Anda. Wilayah 1 default ke Wilayah tempat Anda membuat pipeline. Anda dapat menambahkan Wilayah untuk distribusi dengan tombol Tambah Wilayah, dan Anda dapat menghapus semua Wilayah kecuali Wilayah 1.

Pengaturan wilayah meliputi:

- Wilayah Target
- Nama Output AMI
- Luncurkan izin, dan akun untuk dibagikan
- Lisensi terkait (Konfigurasi lisensi asosiasi)

Note

Pengaturan License Manager tidak akan mereplikasi seluruh AWS Wilayah yang harus diaktifkan di akun Anda, misalnya, antara Wilayah ap-east-1 (Hong Kong) dan me-south-1 (Bahrain).

Untuk mengedit setelan distribusi Anda dari halaman detail pipeline, ikuti langkah-langkah berikut:

1. Pilih tab Pengaturan distribusi.
2. Pilih Edit dari sudut kanan atas panel Detail distribusi.
3. Saat Anda siap untuk menyimpan pembaruan, pilih Simpan perubahan.

Edit jadwal pembuatan untuk pipeline Anda

Halaman Edit pipeline mencakup detail berikut yang dapat Anda edit setelah membuat pipeline:

- Deskripsi untuk pipeline Anda.
- Koleksi metadata yang disempurnakan. Ini dihidupkan secara default. Untuk mematikannya, kosongkan kotak centang Aktifkan koleksi metadata yang disempurnakan.
- Jadwal Build untuk pipeline Anda. Anda dapat mengubah opsi Jadwal dan semua pengaturan di sini.

Untuk mengedit pipeline Anda dari halaman detail pipeline, ikuti langkah-langkah berikut:

1. Di sudut kanan atas halaman detail pipeline, pilih Tindakan, lalu Edit pipeline.
2. Saat Anda siap untuk menyimpan pembaruan, pilih Simpan perubahan.

Note

Untuk informasi selengkapnya tentang penjadwalan build menggunakan ekspresi cron, lihat [Gunakan ekspresi cron di EC2 Image Builder](#)

Perbarui saluran pipa gambar AMI () AWS CLI

Anda dapat memperbarui pipeline gambar AMI menggunakan file JSON sebagai input ke `update-image-pipeline` perintah di AWS CLI file. Untuk mengonfigurasi file JSON, Anda harus memiliki Nama Sumber Daya Amazon (ARN) untuk mereferensikan sumber daya yang ada berikut:

- Pipeline gambar untuk diperbarui
- Resep gambar
- Konfigurasi infrastruktur
- Pengaturan distribusi

Anda dapat memperbarui pipeline gambar AMI dengan `update-image-pipeline` perintah AWS CLI sebagai berikut:

Note

`UpdateImagePipeline` tidak mendukung pembaruan selektif untuk pipeline. Anda harus menentukan semua properti yang diperlukan dalam permintaan pembaruan, bukan hanya properti yang telah berubah.

1. Buat file JSON input CLI

Gunakan alat pengeditan file favorit Anda untuk membuat file JSON dengan kunci berikut, ditambah nilai yang valid untuk lingkungan Anda. Contoh ini menggunakan file bernama `create-component.json`:

```
{
  "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-
pipeline/my-example-pipeline",
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-
example-recipe/2019.12.08",
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/my-example-infrastructure-
configuration",
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/my-example-distribution-
configuration",
  "imageTestsConfiguration": {
```

```
"imageTestsEnabled": true,
"timeoutMinutes": 120
},
"schedule": {
"scheduleExpression": "cron(0 0 * * MON *)",
"pipelineExecutionStartCondition":
"EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
},
"status": "DISABLED"
}
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (`\`) sedangkan Linux menggunakan garis miring (`/`) untuk merujuk ke jalur direktori.

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input.

```
aws imagebuilder update-image-pipeline --cli-input-json file://update-image-pipeline.json
```

Buat dan perbarui pipeline gambar kontainer

Anda dapat mengatur, mengonfigurasi, dan mengelola pipeline image container menggunakan konsol Image Builder, melalui Image Builder API, atau dengan `imagebuilder` perintah di AWS CLI file. Wizard konsol Create image pipeline menyediakan artefak awal, dan memandu Anda melalui langkah-langkah untuk:

- Pilih gambar dasar dari gambar terkelola mulai cepat, Amazon ECR, atau repositori Docker Hub
- Tambah dan hapus perangkat lunak
- Sesuaikan pengaturan dan skrip
- Jalankan tes yang dipilih
- Buat Dockerfile menggunakan variabel waktu pembuatan yang telah dikonfigurasi sebelumnya.

- Mendistribusikan gambar ke AWS Wilayah

Untuk informasi selengkapnya dan step-by-step tutorial tentang penggunaan wizard Create image pipeline console, lihat [Buat pipeline gambar kontainer menggunakan wizard konsol EC2 Image Builder](#).

Daftar Isi

- [Buat pipeline gambar kontainer \(AWS CLI\)](#)
- [Perbarui pipeline gambar kontainer \(konsol\)](#)
- [Perbarui pipa gambar kontainer \(\) AWS CLI](#)

Buat pipeline gambar kontainer (AWS CLI)

Anda dapat membuat pipeline gambar kontainer menggunakan file JSON sebagai input ke [create-image-pipeline](#) perintah di AWS CLI file.

Seberapa sering pipeline Anda membuat gambar baru untuk menggabungkan pembaruan yang tertunda dari gambar dasar dan komponen Anda tergantung pada `schedule` yang telah Anda konfigurasi. A `schedule` memiliki atribut berikut:

- `scheduleExpression`— Tetapkan jadwal kapan pipeline Anda berjalan untuk mengevaluasi `pipelineExecutionStartCondition` dan menentukan apakah itu harus memulai pembangunan. Jadwal dikonfigurasi dengan ekspresi cron. Untuk informasi selengkapnya tentang cara memformat ekspresi cron di Image Builder, lihat [Gunakan ekspresi cron di EC2 Image Builder](#).
- `pipelineExecutionStartCondition`— Menentukan apakah pipeline Anda harus memulai pembangunan. Nilai yang valid meliputi:
 - `EXPRESSION_MATCH_ONLY`— pipeline Anda akan membangun gambar baru setiap kali ekspresi cron cocok dengan waktu saat ini.
 - `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`— pipeline Anda tidak akan memulai pembuatan gambar baru kecuali ada perubahan yang tertunda pada image atau komponen dasar Anda.

Ketika Anda menjalankan `create-image-pipeline` perintah di AWS CLI, banyak sumber daya konfigurasi adalah opsional. Namun, beberapa sumber daya memiliki persyaratan bersyarat,

tergantung pada jenis gambar apa yang dibuat pipa. Sumber daya berikut diperlukan untuk pipeline gambar kontainer:

- Resep kontainer ARN
- Konfigurasi infrastruktur ARN

Jika Anda tidak menyertakan sumber daya konfigurasi distribusi saat menjalankan `create-image-pipeline` perintah, gambar keluaran disimpan di repositori ECR yang Anda tentukan sebagai repositori target dalam resep kontainer Anda di Wilayah tempat Anda menjalankan perintah. Jika Anda menyertakan sumber daya konfigurasi distribusi untuk pipeline Anda, repositori target yang telah Anda tentukan untuk Wilayah pertama dalam distribusi akan digunakan.

1. Buat file JSON input CLI

Gunakan alat pengeditan file favorit Anda untuk membuat file JSON dengan kunci berikut, ditambah nilai yang valid untuk lingkungan Anda. Contoh ini menggunakan file bernama `create-image-pipeline.json`:

```
{
  "name": "MyWindows2019Pipeline",
  "description": "Builds Windows 2019 Images",
  "enhancedImageMetadataEnabled": true,
  "containerRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:container-recipe/my-example-recipe/2020.12.03",
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/my-example-infrastructure-configuration",
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-example-distribution-configuration",
  "imageTestsConfiguration": {
    "imageTestsEnabled": true,
    "timeoutMinutes": 60
  },
  "schedule": {
    "scheduleExpression": "cron(0 0 * * SUN *)",
    "pipelineExecutionStartCondition": "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
  },
  "status": "ENABLED"
}
```

```
}
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input.

```
aws imagebuilder create-image-pipeline --cli-input-json file://create-image-pipeline.json
```

Perbarui pipeline gambar kontainer (konsol)

Setelah membuat pipeline image kontainer Image Builder untuk image Docker, Anda dapat membuat perubahan pada konfigurasi infrastruktur dan pengaturan distribusi dari konsol Image Builder.

Untuk memperbarui pipeline gambar kontainer dengan resep wadah baru, Anda harus menggunakan AWS CLI. Untuk informasi selengkapnya, silakan lihat [Perbarui pipa gambar kontainer \(\) AWS CLI](#) dalam panduan ini.

Pilih pipeline image Image Builder Docker yang sudah ada

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Untuk melihat daftar pipeline gambar yang dibuat di bawah akun Anda, pilih Pipeline gambar dari panel navigasi.

Note

Daftar pipeline gambar mencakup indikator untuk jenis gambar keluaran yang dibuat oleh pipeline — AMI atau Docker.

3. Untuk melihat detail atau mengedit pipeline, pilih tautan Nama saluran pipa. Ini membuka tampilan detail untuk pipa.

 Note

Anda juga dapat memilih kotak centang di sebelah nama Pipeline, lalu pilih Lihat detail.

Detail saluran pipa

Halaman detail pipeline EC2 Image Builder mencakup bagian-bagian berikut:

Ringkasan

Bagian di bagian atas halaman merangkum detail utama untuk pipeline yang terlihat dengan tab detail apa pun yang terbuka. Detail yang ditampilkan di bagian ini hanya dapat diedit pada tab detailnya masing-masing.

Tab detail

- Gambar keluaran - Menampilkan gambar keluaran yang dihasilkan pipa.
- Resep kontainer — Menampilkan detail resep. Setelah Anda membuat resep, Anda tidak dapat mengeditnya. Anda harus membuat versi baru resep dari halaman Resep kontainer. Untuk informasi selengkapnya, lihat [Buat versi baru resep wadah](#).
- Konfigurasi infrastruktur — Menampilkan informasi yang dapat diedit untuk mengonfigurasi infrastruktur pipeline build Anda.
- Pengaturan distribusi - Menampilkan informasi yang dapat diedit untuk distribusi gambar Docker.
- EventBridge rules — Untuk Event Bus yang dipilih, menampilkan EventBridge aturan yang menargetkan pipeline saat ini. Termasuk Buat bus acara dan Buat tindakan aturan yang ditautkan ke EventBridge konsol. Untuk informasi selengkapnya tentang tab ini, lihat [Gunakan EventBridge aturan](#).

Edit konfigurasi infrastruktur untuk pipeline Anda

Konfigurasi infrastruktur mencakup detail berikut yang dapat Anda edit setelah membuat pipeline:

- Deskripsi untuk konfigurasi infrastruktur Anda.
- Peran IAM untuk dikaitkan dengan profil instance.
- AWS infrastruktur, termasuk jenis Instance dan topik SNS untuk notifikasi.
- VPC, subnet, dan grup keamanan.

- Pengaturan pemecahan masalah, termasuk Instance Terminate saat gagal, pasangan Kunci untuk menghubungkan, dan lokasi bucket S3 opsional untuk log instance.

Untuk mengedit konfigurasi infrastruktur dari halaman detail pipeline, ikuti langkah-langkah berikut:

1. Pilih tab konfigurasi Infrastruktur.
2. Pilih Edit dari sudut kanan atas panel detail Konfigurasi.
3. Ketika Anda siap untuk menyimpan pembaruan yang telah Anda buat untuk konfigurasi infrastruktur Anda, pilih Simpan perubahan.

Mengedit setelah distribusi untuk pipeline

Pengaturan distribusi mencakup detail berikut yang dapat Anda edit setelah membuat pipeline:

- Deskripsi untuk pengaturan distribusi Anda.
- Pengaturan wilayah untuk Wilayah tempat Anda mendistribusikan gambar Anda. Region 1 default ke Wilayah tempat Anda membuat pipeline. Anda dapat menambahkan Wilayah untuk distribusi dengan tombol Tambah Wilayah, dan Anda dapat menghapus semua Wilayah kecuali Wilayah 1.

Pengaturan wilayah meliputi:

- Wilayah Target
- Layanan default ke “ECR”, dan tidak dapat diedit.
- Nama repositori — nama repositori target Anda (tidak termasuk lokasi Amazon ECR). Misalnya, nama repositori dengan lokasi akan terlihat seperti pola berikut:

```
<account-id>.dkr.ecr.<region>.amazonaws.com/<repository-name>
```

Note

Jika Anda mengubah nama Repositori, hanya gambar yang dibuat setelah perubahan nama yang akan ditambahkan di bawah nama baru. Gambar sebelumnya yang dibuat pipeline Anda tetap berada di repositori aslinya.

Untuk mengedit setelah distribusi Anda dari halaman detail pipeline, ikuti langkah-langkah berikut:

1. Pilih tab Pengaturan distribusi.

2. Pilih Edit dari sudut kanan atas panel Detail distribusi.
3. Saat Anda siap untuk menyimpan pembaruan yang telah Anda buat ke pengaturan distribusi, pilih Simpan perubahan.

Edit jadwal pembuatan untuk pipeline Anda

Halaman Edit pipeline mencakup detail berikut yang dapat Anda edit setelah membuat pipeline:

- Deskripsi untuk pipeline Anda.
- Koleksi metadata yang disempurnakan. Ini dihidupkan secara default. Untuk mematikannya, kosongkan kotak centang Aktifkan koleksi metadata yang disempurnakan.
- Jadwal Build untuk pipeline Anda. Anda dapat mengubah opsi Jadwal dan semua pengaturan di bagian ini.

Untuk mengedit pipeline Anda dari halaman detail pipeline, ikuti langkah-langkah berikut:

1. Di sudut kanan atas halaman detail pipeline, pilih Tindakan, lalu Edit pipeline.
2. Saat Anda siap untuk menyimpan pembaruan, pilih Simpan perubahan.

Note

Untuk informasi selengkapnya tentang penjadwalan build menggunakan ekspresi cron, lihat.

[Gunakan ekspresi cron di EC2 Image Builder](#)

Perbarui pipa gambar kontainer () AWS CLI

Anda dapat memperbarui pipeline gambar kontainer menggunakan file JSON sebagai input ke [update-image-pipeline](#) perintah di AWS CLI file. Untuk mengonfigurasi file JSON, Anda harus memiliki Nama Sumber Daya Amazon (ARN) untuk mereferensikan sumber daya yang ada berikut:

- Pipeline gambar untuk diperbarui
- Resep kontainer
- Konfigurasi infrastruktur
- Pengaturan distribusi (jika termasuk dalam pipa saat ini)

Note

Jika sumber daya pengaturan distribusi disertakan, maka repositori ECR yang ditentukan sebagai repositori target dalam pengaturan distribusi untuk Wilayah tempat perintah berjalan (Wilayah 1) lebih diutamakan daripada repositori target yang ditentukan dalam resep wadah.

Ikuti langkah-langkah ini untuk memperbarui pipeline image container menggunakan update-image-pipeline perintah diAWS CLI:

Note

UpdateImagePipeline tidak mendukung pembaruan selektif untuk pipeline. Anda harus menentukan semua properti yang diperlukan dalam permintaan pembaruan, bukan hanya properti yang telah berubah.

1. Buat file JSON input CLI

Gunakan alat pengeditan file favorit Anda untuk membuat file JSON dengan kunci berikut, ditambah nilai yang valid untuk lingkungan Anda. Contoh ini menggunakan file bernama `create-component.json`:

```
{
  "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-
pipeline/my-example-pipeline",
  "containerRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:container-
recipe/my-example-recipe/2020.12.08",
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/my-example-infrastructure-
configuration",
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/my-example-distribution-
configuration",
  "imageTestsConfiguration": {
    "imageTestsEnabled": true,
    "timeoutMinutes": 120
  },
  "schedule": {
    "scheduleExpression": "cron(0 0 * * MON *)",
```

```
"pipelineExecutionStartCondition":  
"EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"  
},  
"status": "DISABLED"  
}
```

Note

- Anda harus menyertakan notasi `file://` pada awal jalur file JSON.
- Jalur untuk file JSON harus mengikuti konvensi yang sesuai untuk sistem operasi dasar tempat Anda menjalankan perintah. Sebagai contoh, Windows menggunakan garis miring terbalik (\) sedangkan Linux menggunakan garis miring (/) untuk merujuk ke jalur direktori.

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input.

```
aws imagebuilder update-image-pipeline --cli-input-json file://update-image-pipeline.json
```

Konfigurasi alur kerja gambar untuk pipeline EC2 Image Builder

Dengan alur kerja gambar, Anda dapat menyesuaikan alur kerja yang dijalankan pipeline untuk membangun dan menguji gambar sesuai dengan kebutuhan Anda. Alur kerja yang Anda tentukan berjalan dalam konteks kerangka kerja alur kerja Image Builder. Untuk informasi selengkapnya tentang tahapan yang membentuk kerangka kerja alur kerja, lihat [Mengelola alur kerja build dan test untuk image EC2 Image Builder](#).

Membangun alur kerja

Membangun alur kerja berjalan selama `Build` tahap kerangka kerja alur kerja. Anda hanya dapat menentukan satu alur kerja build untuk pipeline Anda. Atau Anda dapat melewati build sepenuhnya untuk mengonfigurasi pipeline khusus pengujian.

Alur kerja uji

Alur kerja pengujian dijalankan selama `Test` tahap kerangka kerja alur kerja. Anda dapat menentukan hingga sepuluh alur kerja pengujian untuk pipeline Anda. Anda juga dapat melewati pengujian sepenuhnya jika Anda hanya ingin pipeline Anda dibangun.

Tentukan grup uji untuk alur kerja pengujian

Alur kerja pengujian didefinisikan dalam kelompok uji. Anda dapat menjalankan hingga sepuluh alur kerja pengujian untuk pipeline Anda. Anda memutuskan apakah akan menjalankan alur kerja pengujian dalam urutan tertentu atau menjalankan sebanyak mungkin pada saat yang bersamaan. Bagaimana mereka berjalan tergantung pada bagaimana Anda mendefinisikan kelompok pengujian Anda. Skenario berikut menunjukkan beberapa cara agar Anda dapat menentukan alur kerja pengujian Anda.

Note

Jika Anda menggunakan konsol untuk membuat alur kerja, sebaiknya luangkan waktu untuk merencanakan cara menjalankan alur kerja pengujian sebelum menentukan grup pengujian. Di konsol, Anda dapat menambah atau menghapus alur kerja dan grup pengujian, tetapi Anda tidak dapat menyusunnya kembali.

Skenario 1: Jalankan satu alur kerja pengujian pada satu waktu

Untuk menjalankan semua alur kerja pengujian Anda satu per satu, Anda dapat mengonfigurasi hingga sepuluh grup pengujian, masing-masing dengan satu alur kerja pengujian di dalamnya. Grup pengujian dijalankan satu per satu, dalam urutan Anda menambahkannya ke pipeline Anda. Ini adalah salah satu cara untuk memastikan bahwa alur kerja pengujian Anda berjalan satu per satu dalam urutan tertentu.

Skenario 2: Jalankan beberapa alur kerja pengujian secara bersamaan

Jika urutannya tidak penting, dan Anda ingin menjalankan alur kerja pengujian sebanyak mungkin secara bersamaan, Anda dapat mengonfigurasi satu grup pengujian dan memasukkan jumlah maksimum alur kerja pengujian di dalamnya. Image Builder memulai hingga lima alur kerja pengujian secara bersamaan, dan memulai alur kerja pengujian tambahan saat yang lain selesai. Jika tujuan Anda adalah menjalankan alur kerja pengujian secepat mungkin, ini adalah salah satu cara untuk melakukannya.

Skenario 3: Campur dan cocokkan

Jika Anda memiliki skenario campuran, dengan beberapa alur kerja pengujian yang dapat berjalan pada saat yang sama dan beberapa yang harus dijalankan satu per satu, Anda dapat mengonfigurasi

grup pengujian untuk mencapai tujuan ini. Satu-satunya batasan cara Anda mengonfigurasi grup pengujian adalah jumlah maksimum alur kerja pengujian yang dapat dijalankan untuk pipeline

Mengatur parameter alur kerja di pipeline Image Builder (konsol)

Parameter alur kerja berfungsi dengan cara yang sama untuk membangun alur kerja dan menguji alur kerja. Saat membuat atau memperbarui pipeline, pilih alur kerja build dan uji yang ingin disertakan. Jika Anda menentukan parameter dalam dokumen alur kerja untuk alur kerja yang Anda pilih, Image Builder akan menampilkannya di panel Parameter. Panel disembunyikan untuk alur kerja yang tidak memiliki parameter yang ditentukan.

Setiap parameter menampilkan atribut berikut yang ditentukan oleh dokumen alur kerja Anda:

- Nama (tidak dapat diedit) — Nama parameter.
- Jenis (tidak dapat diedit) - Tipe data untuk nilai parameter.
- Nilai — Nilai untuk parameter. Anda dapat mengedit nilai parameter untuk mengaturnya untuk pipeline Anda.

Tentukan peran layanan IAM yang digunakan Image Builder untuk menjalankan tindakan alur kerja

Akses layanan

Untuk menjalankan alur kerja gambar, Image Builder memerlukan izin untuk melakukan tindakan alur kerja. Anda dapat menentukan peran [AWSServiceRoleForImageBuilder](#) terkait layanan, atau Anda dapat menentukan peran kustom Anda sendiri untuk akses layanan, sebagai berikut.

- Konsol — Di panduan pipeline Langkah 3 Tentukan proses pembuatan gambar, pilih peran terkait layanan atau peran kustom Anda sendiri dari daftar peran IAM di panel akses Layanan.
- Image Builder API - Dalam permintaan [CreateImage](#) tindakan, tentukan peran terkait layanan atau peran kustom Anda sendiri sebagai nilai untuk parameter. `executionRole`

Untuk mempelajari lebih lanjut tentang cara membuat peran layanan, lihat [Membuat peran untuk mendelegasikan izin ke AWS layanan](#) di AWS Identity and Access Management Panduan Pengguna.

Jalankan pipeline gambar Anda

Jika Anda memilih opsi jadwal manual untuk pipeline Anda, itu hanya akan berjalan saat Anda memulai build secara manual. Jika Anda memilih salah satu opsi penjadwalan otomatis, Anda juga dapat menjalankannya secara manual, di antara jadwal yang dijadwalkan secara teratur. Misalnya, jika Anda memiliki pipeline yang biasanya berjalan sebulan sekali, tetapi Anda perlu memasukkan pembaruan ke salah satu komponen Anda dua minggu setelah proses sebelumnya, Anda dapat memilih untuk menjalankan pipeline secara manual.

Console

Untuk menjalankan pipeline Anda dari halaman detail pipeline di konsol Image Builder, pilih Jalankan pipeline dari menu Tindakan di bagian atas halaman. Pesan status muncul di bagian atas halaman untuk memberi tahu Anda bahwa pipeline Anda telah dimulai, atau jika ada kesalahan.

1. Di sudut kiri atas halaman rincian pipeline, pilih Run pipeline, dari menu Actions.
2. Anda dapat melihat status pipeline Anda saat ini di tab Gambar keluaran, di kolom Status.

AWS CLI

Contoh berikut menunjukkan bagaimana menggunakan [start-image-pipeline-execution](#) perintah di AWS CLI untuk memulai pipa gambar secara manual. Saat Anda menjalankan perintah ini, pipeline membangun dan mendistribusikan gambar baru.

```
aws imagebuilder start-image-pipeline-execution --image-pipeline-arn
arn:aws:imagebuilder:us-west-2:111122223333:image-pipeline/my-example-pipeline
```

Untuk melihat sumber daya apa yang dibuat saat pipeline build berjalan, lihat [Sumber daya dibuat](#).

Gunakan ekspresi cron di EC2 Image Builder

Gunakan ekspresi cron untuk EC2 Image Builder untuk menyiapkan jendela waktu untuk menyegarkan gambar Anda dengan pembaruan yang berlaku untuk gambar dasar dan komponen pipeline Anda. Jendela waktu untuk penyegaran pipeline dimulai dengan waktu yang Anda atur dalam ekspresi cron. Anda dapat mengatur waktu dalam ekspresi cron Anda ke menit. Pembuatan pipeline Anda dapat berjalan pada atau setelah waktu mulai.

Terkadang butuh beberapa detik, atau hingga satu menit agar build Anda mulai berjalan.

Note

Ekspresi cron menggunakan Universal Coordinated Time (UTC). Untuk informasi selengkapnya tentang waktu UTC, dan untuk menemukan offset untuk zona waktu Anda, lihat [Singkatan Zona Waktu — Daftar Seluruh Dunia](#).

Nilai yang didukung untuk ekspresi cron di Image Builder

EC2 Image Builder menggunakan format cron yang terdiri dari enam bidang wajib. Masing-masing dipisahkan dari yang lain oleh ruang di antaranya, tanpa ruang depan atau belakang:

<Minute> <Hour> <Day> <Month> <Day of the week> <Year>

Tabel berikut menunjukkan nilai support untuk entri cron diperlukan.

Nilai support untuk ekspresi cron

Bidang	Nilai	Wildcard
Menit	0-59	, - * /
Jam	0-23	, - * /
Hari	1-31	, - * ? / L W
Bulan	1-12 atau jan-dec	, - * /
Hari dalam seminggu	1-7 atau sun-sat	, - * ? L #
Tahun	1970-2199	, - * /

Wildcard

Tabel berikut menjelaskan bagaimana Image Builder menggunakan wildcard untuk ekspresi cron. Ingatlah bahwa ini dapat memakan waktu hingga satu menit setelah waktu yang Anda tentukan untuk memulai build.

Wildcard support untuk ekspresi cron

Wildcard	Deskripsi
,	Wildcard , (koma) mencakup nilai tambahan. Di bidang Bulan, jan, feb, mar termasuk Januari, Februari, dan Maret.
-	Wildcard - (tanda hubung) menentukan rentang. Di bidang hari bulan, 1-15 termasuk hari 1 hingga 15 dari bulan yang ditentukan.
*	Wildcard * (asterisk) mencakup semua nilai yang valid untuk bidang tersebut.
?	Wildcard ? (tanda tanya) menentukan bahwa nilai bidang tergantung pada pengaturan lain. Dalam kasus ay-of-week bidang Hari dan D, ketika satu ditentukan atau mencakup semua nilai yang mungkin (*), yang lain harus berupa a?. Anda tidak dapat menentukan keduanya. Misalnya, jika Anda memasukkan kolom 7 in the Day (jalankan build pada hari ketujuh dalam sebulan), ay-of-week posisi D harus berisi a?.
/	Wildcard / (garis miring) menentukan tambahan. Misalnya, jika Anda ingin build Anda berjalan setiap hari, masukkan */2 di kolom hari.
L	Wildcard L di salah satu bidang hari, menentukan hari terakhir: 28-31 untuk hari dalam sebulan, tergantung pada bulan apa, atau Minggu, untuk hari dalam seminggu.
W	Wildcard W di ay-of-month bidang D menentukan hari kerja. Di ay-of-month bidang D, jika Anda memasukkan nomor sebelumW, itu berarti Anda ingin menargetkan hari kerja

Wildcard	Deskripsi
	yang paling dekat dengan hari itu. Misalnya, jika Anda menentukan <code>3W</code> , Anda ingin build Anda berjalan pada hari kerja yang paling dekat dengan hari ketiga setiap bulan.
#	# (hash) hanya diperbolehkan untuk bidang hari dalam seminggu, dan harus diikuti oleh angka antara 1 dan 5. Angka tersebut menentukan minggu mana di bulan tertentu yang berlaku untuk menjalankan build. Misalnya, jika Anda ingin build Anda berjalan pada hari Jumat kedua setiap bulan, gunakan <code>fri#2</code> untuk bidang hari dalam seminggu.

Pembatasan

- Anda tidak dapat menentukan ay-of-week bidang D ay-of-month dan D dalam ekspresi cron yang sama. Jika Anda menentukan nilai atau * di salah satu bidang ini, Anda harus menggunakan ? di bidang lainnya.
- Ekspresi cron yang mengarah ke tingkat lebih cepat dari satu menit tidak didukung.

Contoh ekspresi cron di EC2 Image Builder

Ekspresi cron dimasukkan secara berbeda untuk konsol Image Builder, dibandingkan untuk API atau CLI. Untuk melihat contoh, pilih tab yang berlaku untuk Anda.

Image Builder console

Contoh berikut menunjukkan ekspresi cron yang dapat Anda masukkan ke konsol untuk jadwal build Anda. Waktu UTC ditentukan menggunakan jam 24 jam.

Jalankan setiap hari pada pukul 10:00 (UTC)

```
0 10 * * ? *
```

Jalankan setiap hari pukul 12:15 (UTC)

```
15 12 * * ? *
```

Jalankan setiap hari pada tengah malam (UTC)

```
0 0 * * ? *
```

Jalankan pukul 10:00 pagi (UTC) setiap pagi hari kerja

```
0 10 ? * 2-6 *
```

Jalankan pukul 6 sore (UTC) setiap malam hari kerja

```
0 18 ? * mon-fri *
```

Jalankan pukul 8:00 pagi (UTC) pada hari pertama setiap bulan

```
0 8 1 * ? *
```

Jalankan pada hari Selasa kedua setiap bulan pukul 22:30 (UTC)

```
30 22 ? * tue#2 *
```

Tip

Jika Anda tidak ingin pekerjaan pipeline Anda diperpanjang ke hari berikutnya saat sedang berjalan, pastikan Anda memperhitungkan waktu untuk build saat menentukan waktu mulai.

API/CLI

Contoh berikut menunjukkan ekspresi cron yang dapat Anda masukkan untuk jadwal build menggunakan perintah CLI atau permintaan API. Hanya ekspresi cron yang ditampilkan.

Jalankan setiap hari pada pukul 10:00 (UTC)

```
cron(0 10 * * ? *)
```

Jalankan setiap hari pukul 12:15 (UTC)

```
cron(15 12 * * ? *)
```

Jalankan setiap hari pada tengah malam (UTC)

```
cron(0 0 * * ? *)
```

Jalankan pukul 10:00 pagi (UTC) setiap pagi hari kerja

```
cron(0 10 ? * 2-6 *)
```

Jalankan pukul 18:00 (UTC) setiap malam hari kerja

```
cron(0 18 ? * mon-fri *)
```

Jalankan pukul 8:00 pagi (UTC) pada hari pertama setiap bulan

```
cron(0 8 1 * ? *)
```

Jalankan pada hari Selasa kedua setiap bulan pukul 22:30 (UTC)

```
cron(30 22 ? * tue#2 *)
```

Tip

Jika Anda tidak ingin pekerjaan pipeline Anda diperpanjang ke hari berikutnya saat sedang berjalan, pastikan Anda memperhitungkan waktu untuk build saat menentukan waktu mulai.

Nilai ekspresi di EC2 Image Builder

Ekspresi rate dimulai ketika Anda membuat aturan peristiwa terjadwal, dan kemudian aturan berjalan pada jadwal yang ditetapkan.

Ekspresi rate memiliki dua field wajib berikut. Field dipisahkan oleh white space.

Sintaks

```
rate(value unit)
```

nilai

Bilangan positif

unit

Unit waktu. Unit yang berbeda diperlukan untuk nilai 1, seperti `minute`, dan nilai lebih dari 1, seperti `minutes`.

Nilai yang valid: `menit` | `menit-menit` | `jam` | `jam-jam` | `hari` | `hari-hari`

Pembatasan

Jika nilai sama dengan 1, maka unit harus tunggal. Demikian pula, untuk nilai lebih besar dari 1, unit harus jamak. Misalnya, `rate(1 hours)` dan `rate(5 hour)` tidak valid, tetapi `rate(1 hour)` dan `rate(5 hours)` valid.

Menggunakan EventBridge aturan dengan pipeline Image Builder

Acara dari berbagai layanan AWS dan mitra dialirkan ke bus EventBridge acara Amazon dalam waktu dekat. Anda juga dapat membuat acara khusus, dan mengirim acara dari aplikasi Anda sendiri ke EventBridge. Bus acara menggunakan aturan untuk menentukan ke mana harus merutekan data acara.

Pipeline Image Builder tersedia sebagai target EventBridge aturan, yang berarti Anda dapat menjalankan pipeline Image Builder berdasarkan aturan yang Anda buat untuk merespons peristiwa di bus, atau sesuai jadwal.

Note

Bus acara khusus untuk suatu Wilayah. Aturan dan target harus berada di Wilayah yang sama.

Daftar Isi

- [EventBridge istilah](#)
- [Lihat EventBridge aturan untuk pipeline Image Builder](#)
- [Gunakan EventBridge aturan untuk menjadwalkan pembuatan pipeline](#)

EventBridge istilah

Bagian ini berisi ringkasan istilah untuk membantu Anda memahami cara EventBridge terintegrasi dengan pipeline Image Builder Anda.

Peristiwa

Menjelaskan perubahan dalam lingkungan yang mungkin memengaruhi satu atau lebih sumber daya aplikasi. Lingkungan dapat berupa AWS lingkungan, layanan atau aplikasi mitra SaaS, atau salah satu aplikasi atau layanan Anda. Anda juga dapat mengatur acara terjadwal pada timeline.

Bus peristiwa

Pipeline yang menerima data peristiwa dari aplikasi dan layanan.

Sumber

Layanan atau aplikasi yang mengirimkan acara ke bus acara.

Target

Sumber daya atau titik akhir yang EventBridge dipanggil saat cocok dengan aturan, mengirimkan data dari acara ke target.

Aturan

Aturan mencocokkan peristiwa yang masuk dan mengirimkannya ke target untuk pemrosesan. Aturan tunggal dapat mengirim acara ke beberapa target, yang kemudian dapat berjalan secara paralel. Aturan didasarkan baik pada pola peristiwa atau jadwal.

Pola

Pola peristiwa mendefinisikan struktur acara dan bidang yang cocok dengan aturan untuk memulai tindakan target.

Jadwal

Aturan jadwal melakukan tindakan sesuai jadwal, seperti menjalankan pipeline Image Builder untuk menyegarkan gambar setiap tiga bulan. Ada dua jenis ekspresi jadwal:

- Ekspresi cron — Cocokkan kriteria penjadwalan tertentu menggunakan sintaks cron yang dapat menguraikan kriteria sederhana; misalnya, berjalan mingguan pada hari tertentu. Anda juga dapat menetapkan kriteria yang lebih kompleks, seperti berjalan setiap triwulanan pada hari kelima setiap bulan, antara jam 2 pagi dan 4 pagi.
- Ekspresi nilai - Tentukan interval reguler saat target dipanggil, seperti setiap 12 jam.

Lihat EventBridge aturan untuk pipeline Image Builder

Tab EventBridge aturan di halaman detail saluran pipa Image Builder Image menampilkan bus EventBridge acara yang dapat diakses akun Anda, dan aturan untuk bus acara yang dipilih yang berlaku untuk pipeline saat ini. Tab ini juga menautkan langsung ke EventBridge konsol untuk membuat sumber daya baru.

Tindakan yang menautkan ke EventBridge konsol

- Buat bus acara
- Buat aturan

Untuk mempelajari selengkapnya EventBridge, lihat topik berikut di Panduan EventBridge Pengguna Amazon.

- [Apa itu Amazon EventBridge](#)
- [Bus EventBridge acara Amazon](#)
- [EventBridge Acara Amazon](#)
- [EventBridge Aturan Amazon](#)

Gunakan EventBridge aturan untuk menjadwalkan pembuatan pipeline

Untuk contoh ini, kita membuat aturan jadwal baru untuk bus acara default, menggunakan ekspresi tingkat. Aturan dalam contoh ini menghasilkan acara di bus acara setiap 90 hari. Acara memulai pembuatan pipeline untuk menyegarkan gambar.

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Untuk melihat daftar pipeline gambar yang dibuat di bawah akun Anda, pilih Pipeline gambar dari panel navigasi.

Note

Daftar pipeline gambar mencakup indikator untuk jenis gambar keluaran yang dibuat oleh pipeline — AMI atau Docker.


3. Untuk melihat detail atau mengedit pipeline, pilih tautan Nama saluran pipa. Ini membuka tampilan detail untuk pipa.

Note

Anda juga dapat memilih kotak centang di sebelah nama Pipeline, lalu pilih Lihat detail.

4. Buka tab EventBridge aturan.
5. Simpan bus acara default yang telah dipilih sebelumnya di panel Event Bus.

6. Pilih Buat aturan. Ini akan membawa Anda ke halaman aturan Buat di EventBridge konsol Amazon.
7. Masukkan nama dan deskripsi untuk aturan. Nama aturan harus unik di dalam bus acara untuk Wilayah yang dipilih.
8. Di panel Tentukan pola, pilih opsi Jadwal. Ini memperluas panel, dengan tarif Tetap setiap opsi dipilih.
9. Masukkan 90 di kotak pertama, dan pilih Hari dari daftar drop-down.
10. Lakukan tindakan berikut di panel Pilih target:
 - a. Pilih EC2 Image Builder dari Target daftar drop-down.
 - b. Untuk menerapkan aturan ke pipeline Image Builder, pilih pipeline target dari daftar drop-down Image Pipeline.
 - c. EventBridge memerlukan izin untuk memulai build untuk pipeline yang dipilih. Untuk contoh ini, pertahankan opsi default untuk Membuat peran baru untuk sumber daya khusus ini.
 - d. Pilih Tambahkan target.
11. Pilih Buat

 Note

Untuk mempelajari setelan aturan ekspresi laju yang tidak tercakup dalam contoh ini, lihat [Nilai ekspresi](#) di Panduan EventBridge Pengguna Amazon.

Integrasikan produk dan layanan di EC2 Image Builder

EC2 Image Builder terintegrasi AWS Marketplace dengan dan aplikasi Layanan AWS lainnya untuk membantu Anda membuat gambar mesin kustom yang kuat dan aman.

Produk

Resep Image Builder dapat menggabungkan produk gambar dari AWS Marketplace dan komponen yang dikelola Image Builder untuk menyediakan fungsionalitas build dan pengujian khusus, sebagai berikut.

- **AWS Marketplace produk gambar** — Gunakan produk gambar dari AWS Marketplace sebagai gambar dasar dalam resep Anda untuk memenuhi standar organisasi, seperti CIS Hardening. Saat membuat resep dari konsol Image Builder, Anda dapat memilih dari langganan yang ada, atau mencari produk tertentu. AWS Marketplace Saat membuat resep dari Image Builder API, CLI, atau SDK, Anda dapat menentukan produk gambar Amazon Resource Name (ARN) yang akan digunakan sebagai gambar dasar.
- **AWSTOE komponen** — Komponen yang Anda tentukan dalam resep dapat melakukan tindakan pembuatan dan pengujian, misalnya, untuk menginstal perangkat lunak atau melakukan validasi kepatuhan. Beberapa produk gambar yang Anda berlangganan dari AWS Marketplace mungkin termasuk komponen pendamping yang dapat Anda gunakan dalam resep Anda. Gambar CIS Hardened menyertakan AWSTOE komponen yang cocok yang dapat Anda gunakan dalam resep Anda untuk menerapkan pedoman CIS Benchmarks Level 1 untuk konfigurasi Anda.

Note


Untuk informasi selengkapnya tentang produk terkait kepatuhan, lihat. [Produk kepatuhan untuk gambar Image Builder](#)

Jasa

Image Builder terintegrasi dengan yang berikut ini Layanan AWS untuk memberikan metrik peristiwa, pencatatan, dan pemantauan terperinci. Informasi ini membantu Anda melacak aktivitas, memecahkan masalah pembuatan gambar, dan membuat otomatisasi berdasarkan pemberitahuan peristiwa.

- AWS CloudTrail— Monitor peristiwa Image Builder yang dikirim ke CloudTrail. Untuk informasi lebih lanjut tentang CloudTrail, lihat [Apa itu AWS CloudTrail?](#) dalam AWS CloudTrail User Guide.
- Amazon CloudWatch Logs — Pantau, simpan, dan akses file log Image Builder Anda. Secara opsional, Anda dapat menyimpan log Anda ke ember S3. Untuk informasi selengkapnya tentang CloudWatch Log, lihat [Apa itu CloudWatch Log Amazon?](#) di Panduan Pengguna CloudWatch Log Amazon.
- Amazon EventBridge - Hubungkan ke aliran data peristiwa real-time dari aktivitas Image Builder di akun Anda. Untuk informasi selengkapnya EventBridge, lihat [Apa itu Amazon EventBridge?](#) di Panduan EventBridge Pengguna Amazon.
- Amazon Inspector — Temukan kerentanan dalam pengaturan perangkat lunak dan jaringan Anda dengan pemindaian otomatis untuk instance pengujian EC2 yang diluncurkan Image Builder buat gambar baru. Image Builder menyimpan temuan untuk sumber daya gambar keluaran sehingga Anda dapat menyelidiki dan memperbaiki setelah instance pengujian Anda dihentikan. Untuk informasi selengkapnya tentang pemindaian dan harga, lihat [Apa itu Amazon Inspector?](#) di Panduan Pengguna Amazon Inspector.

Amazon Inspector juga dapat memindai repositori ECR Anda jika Anda mengonfigurasi pemindaian yang disempurnakan. Untuk informasi selengkapnya, lihat [Memindai gambar penampung Amazon ECR](#) di Panduan Pengguna Amazon Inspector.

 Note

Amazon Inspector adalah fitur berbayar.

- AWS Marketplace— Lihat daftar langganan AWS Marketplace produk Anda saat ini, dan cari produk gambar langsung dari Image Builder. Anda juga dapat menggunakan produk gambar yang telah Anda berlangganan sebagai gambar dasar untuk resep Image Builder. Untuk informasi selengkapnya tentang mengelola AWS Marketplace langganan, lihat [Panduan AWS Marketplace Pembeli](#).
- Amazon Simple Notification Service (Amazon SNS) - Jika dikonfigurasi, publikasikan pesan terperinci tentang status gambar Anda ke topik SNS yang Anda berlangganan. Untuk informasi selengkapnya tentang Amazon SNS, lihat [Apa itu Amazon SNS?](#) di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

Topik integrasi produk dan layanan

- [AWS CloudTrail integrasi dalam Image Builder](#)

- [Integrasi Amazon CloudWatch Logs di Image Builder](#)
- [EventBridge Integrasi Amazon di Image Builder](#)
- [Integrasi Amazon Inspector di Image Builder](#)
- [AWS Marketplace integrasi dalam Image Builder](#)
- [Integrasi Amazon SNS di Image Builder](#)
- [Produk kepatuhan untuk gambar Image Builder](#)

AWS CloudTrail integrasi dalam Image Builder

Layanan ini mendukung AWS CloudTrail. CloudTrail adalah layanan yang merekam AWS panggilan untuk Anda Akun AWS dan mengirimkan file log ke bucket Amazon S3. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan apa yang berhasil dibuat Layanan AWS, siapa yang membuat permintaan, kapan dibuat, dan sebagainya. Untuk informasi selengkapnya tentang CloudTrail integrasi dengan Image Builder, lihat [Pencatatan panggilan EC2 Image Builder API menggunakan AWS CloudTrail](#).

Untuk mempelajari selengkapnya CloudTrail, termasuk cara mengaktifkannya dan menemukan file log Anda, lihat [Panduan AWS CloudTrail Pengguna](#).

Integrasi Amazon CloudWatch Logs di Image Builder

CloudWatch Dukungan log diaktifkan secara default. Log disimpan pada instance selama proses pembuatan, dan dialirkan ke CloudWatch Log. Log instance dihapus dari instance sebelum pembuatan gambar.

Log build dialirkan untuk mengikuti grup dan streaming Image Builder CloudWatch Logs:

LogGroup:

```
/aws/imagebuilder/ImageName
```

LogStream (x.x.x/x):

```
ImageVersion/ImageBuildVersion
```

Anda dapat memilih keluar dari streaming CloudWatch Log dengan menghapus izin berikut yang terkait dengan profil instans.

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "logs:CreateLogStream",  
      "logs:CreateLogGroup",  
      "logs:PutLogEvents"  
    ],  
    "Resource": "arn:aws:logs:*:*:log-group:/aws/imagebuilder/*"  
  }  
]
```

[Untuk pemecahan masalah lanjutan, Anda dapat menjalankan perintah dan skrip yang telah ditentukan menggunakan Run Command. AWS Systems Manager](#) Untuk informasi selengkapnya, lihat [Memecahkan Masalah EC2 Image Builder](#).

EventBridge Integrasi Amazon di Image Builder

Amazon EventBridge adalah layanan bus acara tanpa server yang dapat Anda gunakan untuk menghubungkan aplikasi Image Builder Anda dengan data terkait dari yang lain. Layanan AWS Di EventBridge, aturan cocok dengan peristiwa yang masuk dan mengirimkannya ke target untuk diproses. Aturan tunggal dapat mengirim acara ke beberapa target, dan peristiwa ini kemudian berjalan secara paralel.

Dengan EventBridge, Anda dapat mengotomatiskan Layanan AWS dan merespons secara otomatis peristiwa sistem seperti masalah ketersediaan aplikasi atau perubahan sumber daya. Acara dari Layanan AWS dikirim ke EventBridge dalam waktu dekat. Anda dapat mengatur aturan yang bereaksi terhadap peristiwa masuk untuk memulai tindakan misalnya, mengirim peristiwa ke fungsi Lambda saat status instans EC2 berubah dari pending menjadi berjalan. Ini disebut pola. Untuk membuat aturan berdasarkan pola peristiwa, lihat [Membuat EventBridge aturan Amazon yang bereaksi terhadap peristiwa](#) di Panduan EventBridge Pengguna Amazon.

Tindakan yang dapat dimulai secara otomatis meliputi:

- Menginvokasi fungsi AWS Lambda
- Menginvokasi Amazon EC2 Run Command
- Menyampaikan peristiwa ke Amazon Kinesis Data Streams
- Mengaktifkan mesin status AWS Step Functions

- Beri tahu topik Amazon SNS atau antrean Amazon SQS

Anda juga dapat mengatur aturan penjadwalan untuk bus acara default untuk melakukan tindakan secara berkala, seperti menjalankan pipeline Image Builder untuk menyegarkan gambar setiap tiga bulan. Ada dua jenis ekspresi jadwal:

- ekspresi cron - Contoh ekspresi cron berikut menjadwalkan tugas untuk dijalankan setiap hari pada siang hari UTC+0:

```
cron(0 12 * * ? *)
```

Untuk informasi selengkapnya tentang menggunakan ekspresi cron dengan EventBridge, lihat [Ekspresi cron](#) di EventBridge Panduan Pengguna Amazon.

- ekspresi tingkat - Contoh berikut dari ekspresi laju menjadwalkan tugas untuk dijalankan setiap 12 jam:

```
rate(12 hour)
```

Untuk informasi selengkapnya tentang menggunakan ekspresi [laju dengan EventBridge](#), lihat [Nilai ekspresi](#) di Panduan EventBridge Pengguna Amazon.

Untuk informasi selengkapnya tentang cara EventBridge mengintegrasikan dengan pipeline image Image Builder, lihat [Menggunakan EventBridge aturan dengan pipeline Image Builder](#)

Integrasi Amazon Inspector di Image Builder

Saat Anda mengaktifkan pemindaian keamanan dengan Amazon Inspector, ini terus memindai gambar mesin dan menjalankan instance di akun Anda untuk mencari kerentanan sistem operasi dan bahasa pemrograman. Jika diaktifkan, pemindaian keamanan otomatis, dan Image Builder dapat menyimpan snapshot temuan dari instance pengujian saat Anda membuat gambar baru. Amazon Inspector adalah layanan berbayar.

Saat Amazon Inspector menemukan kerentanan di perangkat lunak atau pengaturan jaringan Anda, diperlukan tindakan berikut:

- Memberitahu Anda bahwa ada temuan.
- Nilai tingkat keparahan temuan. Peringkat keparahan mengkategorikan kerentanan untuk membantu Anda memprioritaskan temuan Anda, dan mencakup nilai-nilai berikut:

- Tidak digarisbawahi
 - Informasi
 - Rendah
 - Sedang
 - Tinggi
 - Kritis
- Memberikan informasi tentang temuan, dan tautan ke sumber daya tambahan untuk lebih detail.
 - Menawarkan panduan remediasi untuk membantu Anda menyelesaikan masalah yang menghasilkan temuan.

Konfigurasi pemindaian keamanan

Jika Anda telah mengaktifkan Amazon Inspector untuk akun Anda, Amazon Inspector secara otomatis memindai instans EC2 yang diluncurkan Image Builder untuk membuat dan menguji gambar baru. Contoh tersebut memiliki umur yang pendek selama proses pembuatan dan pengujian, dan temuan mereka biasanya akan kedaluwarsa segera setelah instance tersebut ditutup. Untuk membantu Anda menyelidiki dan memulihkan temuan untuk image baru Anda, Image Builder secara opsional dapat menyimpan temuan apa pun yang diidentifikasi oleh Amazon Inspector pada instance pengujian Anda selama proses pembuatan sebagai snapshot.

Untuk mengonfigurasi pemindaian keamanan untuk pipeline Anda, lihat [Konfigurasi pemindaian keamanan untuk gambar Image Builder di AWS Management Console](#).

Tinjau temuan keamanan

Di konsol Image Builder, Anda dapat melihat temuan keamanan untuk semua resource Image Builder di satu tempat. Anda dapat melihat semua temuan di halaman Temuan Keamanan di bagian Ikhtisar Keamanan, atau Anda dapat mengelompokkan temuan berdasarkan kerentanan, berdasarkan pipeline gambar, atau berdasarkan gambar. Konsol default untuk menampilkan semua temuan keamanan. Panel ringkasan untuk opsi Semua temuan keamanan menunjukkan jumlah temuan yang Anda miliki untuk setiap tingkat keparahan. Untuk informasi selengkapnya, lihat [Mengelola temuan keamanan untuk image Image Builder di AWS Management Console](#).

Untuk mempelajari selengkapnya tentang temuan kerentanan Amazon Inspector, lihat [Memahami temuan di Amazon Inspector di Panduan Pengguna Amazon Inspector](#).

AWS Marketplace integrasi dalam Image Builder

AWS Marketplace adalah katalog digital yang dikuratori tempat Anda dapat menemukan dan berlangganan perangkat lunak, data, dan layanan pihak ketiga yang membantu Anda membangun solusi agar sesuai dengan kebutuhan bisnis Anda. AWS Marketplace menyatukan pembeli dan penjual terdaftar yang diautentikasi bersama dengan daftar perangkat lunak dari kategori populer seperti keamanan, jaringan, penyimpanan, pembelajaran mesin, dan banyak lagi.

AWS Marketplace Penjual dapat berupa vendor perangkat lunak independen (ISV), pengecer, atau individu yang memiliki sesuatu untuk ditawarkan yang bekerja dengan AWS produk dan layanan. Saat penjual mengirimkan produk di AWS Marketplace, mereka menentukan harga produk, serta syarat dan ketentuan penggunaan. Pembeli menyetujui harga, syarat, dan ketentuan yang ditetapkan untuk penawaran. Untuk mempelajari lebih lanjut tentang AWS Marketplace, lihat [Apa itu AWS Marketplace?](#)

Note

Penyedia produk data harus memenuhi persyaratan kelayakan AWS Data Exchange. Untuk informasi selengkapnya, lihat [Menyediakan Produk Data di AWS Data Exchange](#) di Panduan Pengguna AWS Data Exchange.

Fitur integrasi AWS Marketplace

Image Builder terintegrasi dengan AWS Marketplace untuk menyediakan kemampuan berikut langsung dari konsol Image Builder:

- Cari produk gambar yang tersedia di AWS Marketplace.
- Lihat daftar langganan AWS Marketplace produk Anda saat ini.
- Gunakan produk AWS Marketplace gambar sebagai gambar dasar untuk resep Image Builder.

Untuk produk yang menyertakan komponen associated AWS Task Orchestrator and Executor (AWSTOE), Anda dapat memfilter pemilik produk di konsol dan di API, SDK, dan CLI. Untuk informasi selengkapnya, lihat [Daftar AWSTOE komponen](#).

Temukan produk AWS Marketplace gambar dari konsol Image Builder

Image Builder terintegrasi dengan AWS Marketplace untuk menampilkan langganan produk gambar Anda langsung dari AWS Marketplace bagian di konsol Image Builder. Anda juga dapat mencari produk AWS Marketplace gambar dari halaman produk Gambar tanpa meninggalkan konsol Image Builder.

Untuk menemukan produk AWS Marketplace gambar dari konsol Image Builder, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder [di https://console.aws.amazon.com/imagebuilder/](https://console.aws.amazon.com/imagebuilder/).
2. Dari panel navigasi, pilih Produk gambar di AWS Marketplace bagian.
3. Halaman Produk gambar menunjukkan ringkasan produk gambar yang telah Anda berlangganan di tab Langganan, atau Anda dapat mencari produk gambar di AWS Marketplace tab.

Image Builder melakukan pra-filter produk AWS Marketplace untuk fokus pada gambar mesin yang dapat Anda gunakan dalam resep Image Builder Anda. Untuk informasi selengkapnya tentang AWS Marketplace integrasi dengan Image Builder, pilih tab yang cocok dengan apa yang ingin Anda lihat.

AWS Marketplace

Tab ini berisi dua panel. Di sebelah kiri, panel Perbaiki hasil membantu Anda memfilter hasil Anda untuk menemukan produk yang ingin Anda berlangganan. Di sebelah kanan, panel Cari produk menunjukkan produk yang memenuhi kriteria filter Anda, dan juga memberi Anda opsi untuk mencari berdasarkan nama produk.

Sempurnakan hasil

Daftar berikut menunjukkan hanya beberapa filter yang dapat Anda terapkan untuk pencarian produk Anda:

- Pilih satu atau beberapa kategori produk, seperti perangkat lunak infrastruktur atau pembelajaran mesin.
- Pilih sistem operasi untuk produk gambar Anda atau pilih semua produk untuk platform sistem operasi tertentu, misalnya Semua Linux/Unix.
- Pilih satu atau lebih penerbit untuk menampilkan produk mereka yang tersedia. Pilih tautan Tampilkan Semua untuk menampilkan semua penayang yang memiliki produk yang sesuai dengan filter yang telah Anda terapkan.

Note

Nama penerbit tidak dalam urutan abjad. Jika Anda mencari penerbit tertentu, seperti `Center for Internet Security`, Anda dapat memasukkan bagian dari nama di kotak pencarian di bagian atas dialog Semua penerbit. Anda harus mengeja nama, sebagai singkatan, seperti CIS mungkin tidak menghasilkan hasil yang Anda cari.

Anda juga dapat menelusuri halaman nama penerbit demi halaman.

Pilihan filter bersifat dinamis. Setiap pilihan yang Anda buat memengaruhi opsi Anda untuk semua kategori lainnya. Ada ribuan produk yang tersedia AWS Marketplace, jadi semakin banyak Anda dapat memfilter, semakin besar kemungkinan Anda menemukan apa yang Anda inginkan.

Cari produk

Untuk menemukan produk tertentu berdasarkan nama, Anda dapat memasukkan bagian dari nama di bilah pencarian di bagian atas panel ini. Setiap hasil produk mencakup rincian berikut:

- Nama produk dan logo. Keduanya ditautkan ke halaman detail produk di AWS Marketplace. Halaman detail terbuka di tab baru di browser Anda. Dari sana, Anda dapat berlangganan produk gambar jika Anda ingin menggunakannya dalam resep Image Builder. Untuk informasi selengkapnya, lihat [Membeli produk](#) di Panduan AWS Marketplace Pembeli.

Jika Anda berlangganan produk gambar di AWS Marketplace, beralih kembali ke tab Image Builder di browser Anda, dan segarkan daftar produk gambar berlangganan Anda untuk melihatnya.

Note

Mungkin perlu beberapa menit sebelum langganan baru Anda tersedia.

- Nama penerbit. Ini ditautkan ke halaman detail penerbit di AWS Marketplace. Halaman detail penerbit terbuka di tab baru di browser Anda.
- Versi produk.

- Peringkat bintang produk, dan tautan langsung ke bagian ulasan halaman detail produk diAWS Marketplace. Halaman detail terbuka di tab baru di browser Anda.
- Beberapa baris pertama dari deskripsi produk.

Tepat di bawah bilah pencarian, Anda dapat melihat berapa banyak hasil yang dihasilkan pencarian Anda dan bagian dari hasil tersebut yang saat ini ditampilkan. Anda dapat menggunakan kontrol tambahan di sisi kanan panel untuk menyesuaikan pengaturan Anda untuk jumlah produk yang akan ditampilkan pada satu waktu, dan urutan pengurutan untuk diterapkan pada hasil Anda. Anda juga dapat menggunakan kontrol pagination ke halaman melalui hasil Anda.

Subscriptions

Tab ini menunjukkan kepada Anda daftar produk gambar yang telah Anda berlangganan. AWS Marketplace Setiap produk berlangganan menunjukkan detail berikut:

- Nama produk. Ini terkait dengan halaman detail produk diAWS Marketplace. Halaman detail produk untuk produk berlangganan Anda terbuka di tab baru di browser Anda.
- Nama penerbit. Ini ditautkan ke halaman detail penerbit diAWS Marketplace. Halaman detail penerbit terbuka di tab baru di browser Anda.
- Versi produk yang Anda berlangganan.
- Jika ada komponen Terkait yang disertakan dengan produk berlangganan Anda, Image Builder menampilkan tautan ke detail AWSTOE komponen.

Di bagian atas halaman, Anda dapat mencari produk tertentu berdasarkan nama, atau Anda dapat halaman melalui hasil Anda dengan kontrol pagination. Untuk menggunakan produk berlangganan sebagai gambar dasar untuk resep baru, pilih produk berlangganan dan pilih Buat resep baru. Image Builder pra-memilih produk pertama dalam daftar Anda secara default.

Note

Jika Anda mencari produk yang baru saja Anda berlangganan, dan Anda tidak melihatnya dalam daftar, gunakan tombol segarkan di bagian atas tab untuk menyegarkan hasil Anda. Mungkin perlu beberapa menit untuk langganan baru muncul dalam daftar.

Gunakan produk AWS Marketplace gambar dalam resep Image Builder

Di konsol Image Builder, ada dua cara untuk membuat resep gambar baru berdasarkan salah satu produk gambar berlangganan Anda.

1. Anda dapat mulai dari halaman produk Gambar sebagai berikut:

1. Buka konsol EC2 Image Builder [di](https://console.aws.amazon.com/imagebuilder/) <https://console.aws.amazon.com/imagebuilder/>.
2. Dari panel navigasi, pilih Produk gambar di AWS Marketplace bagian.
3. Buka tab Langganan.
4. Pilih produk gambar berlangganan untuk digunakan sebagai gambar dasar dalam resep Anda.
5. Pilih Buat resep baru. Ini membuka halaman Buat resep dengan opsi AWS Marketplace gambar dan produk gambar berlangganan Anda yang telah dipilih sebelumnya.
6. Konfigurasi pengaturan yang tersisa untuk resep Anda seperti biasa. Untuk informasi selengkapnya tentang resep gambar, lihat [Buat versi baru dari resep gambar](#).

2. Anda juga dapat membuka halaman Buat resep dan memilih produk AWS Marketplace gambar untuk digunakan sebagai gambar dasar Anda.

1. Buka konsol EC2 Image Builder [di](https://console.aws.amazon.com/imagebuilder/) <https://console.aws.amazon.com/imagebuilder/>.
2. Dari panel navigasi, pilih Resep gambar di AWS Marketplace bagian. Ini menunjukkan kepada Anda daftar resep gambar yang telah Anda buat.
3. Pilih Buat resep gambar. Ini membuka halaman Buat resep.
4. Masukkan Nama dan Versi resep Anda di bagian Detail resep seperti biasa.
5. Di bagian Gambar dasar, pilih opsi AWS Marketplace gambar. Ini menunjukkan kepada Anda daftar produk AWS Marketplace gambar yang telah Anda berlangganan di tab Langganan. Anda dapat memilih gambar dasar Anda dari daftar.

Anda juga dapat mencari produk gambar lain yang tersedia AWS Marketplace langsung dari AWS Marketplace tab. Pilih Tambahkan produk, atau buka AWS Marketplace tab secara langsung. Untuk informasi selengkapnya tentang cara mengatur filter dan mencari di AWS Marketplace, lihat [Temukan produk AWS Marketplace gambar dari konsol Image Builder](#).

6. Masukkan detail yang tersisa seperti biasa, dan pilih Buat resep.

Note

Jika langganan produk image Anda menyertakan komponen AWSTOE build, Anda dapat memilihnya dari daftar komponen Build. Pilih `Third party managed` dari daftar tipe pemilik komponen untuk melihatnya. Jika langganan produk Anda menyertakan komponen AWSTOE pengujian, ikuti prosedur yang sama untuk daftar komponen Uji.

Integrasi Amazon SNS di Image Builder

Amazon Simple Notification Service (Amazon SNS) adalah layanan terkelola yang menyediakan pengiriman pesan asinkron dari penerbit ke pelanggan (juga dikenal sebagai produsen dan konsumen). Anda dapat menentukan topik SNS dalam konfigurasi infrastruktur Anda. Saat Anda membuat gambar atau menjalankan pipeline, Image Builder dapat mempublikasikan pesan mendetail tentang status gambar Anda ke topik ini. Ketika status gambar mencapai salah satu status berikut, Image Builder menerbitkan pesan:

- AVAILABLE
- FAILED

Untuk contoh pesan SNS dari Image Builder, lihat [Format pesan SNS](#). Jika Anda ingin membuat topik SNS baru, lihat [Memulai Amazon SNS](#) di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

Topik SNS Terenkripsi

Jika topik SNS Anda dienkripsi, Anda harus memberikan izin dalam AWS KMS key kebijakan untuk peran layanan Image Builder untuk melakukan tindakan berikut:

- `kms:Decrypt`
- `kms:GenerateDataKey`

Note

Jika topik SNS Anda dienkripsi, kunci yang mengenkripsi topik ini harus berada di akun tempat layanan Image Builder berjalan. Image Builder tidak dapat mengirim notifikasi ke topik SNS yang dienkripsi dengan kunci dari akun lain.

Contoh penambahan kebijakan kunci KMS

Contoh berikut menunjukkan bagian tambahan yang Anda tambahkan ke kebijakan kunci KMS. Gunakan Amazon Resource Name (ARN) untuk peran terkait layanan IAM yang dibuat Image Builder di bawah akun Anda saat pertama kali membuat image Image Builder. Untuk mempelajari lebih lanjut tentang peran terkait layanan Image Builder, lihat [Menggunakan peran terkait layanan untuk EC2 Image Builder](#)

```
{
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::123456789012:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder"
    },
    "Action": [
      "kms:GenerateDataKey*",
      "kms:Decrypt"
    ],
    "Resource": "*"
  }]
}
```

Anda dapat menggunakan salah satu metode berikut untuk mendapatkan ARN.

AWS Management Console

Untuk mendapatkan ARN untuk peran terkait layanan yang dibuat Image Builder di bawah akun Anda AWS Management Console, ikuti langkah-langkah berikut:

1. Buka konsol IAM di <https://console.aws.amazon.com/iam/>.
2. Pada panel navigasi sebelah kiri, pilih Peran.

3. Cari `ImageBuilder`, dan pilih nama Peran berikut dari hasil: `AWSServiceRoleForImageBuilder`. Ini menampilkan halaman detail peran.
4. Untuk menyalin ARN ke clipboard Anda, pilih ikon di sebelah nama ARN.

AWS CLI

Untuk mendapatkan ARN untuk peran terkait layanan yang dibuat Image Builder di bawah akun Anda AWS CLI, gunakan perintah IAM [get-role](#), sebagai berikut.

```
aws iam get-role --role-name AWSServiceRoleForImageBuilder
```

Output sampel sebagian:

```
{
  "Role": {
    "Path": "/aws-service-role/imagebuilder.amazonaws.com/",
    "RoleName": "AWSServiceRoleForImageBuilder",
    ...
    "Arn": "arn:aws:iam::123456789012:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder",
    ...
  }
}
```

Format pesan SNS

Setelah Image Builder menerbitkan pesan ke topik Amazon SNS Anda, layanan lain yang berlangganan topik dapat memfilter pada format pesan dan menentukan apakah memenuhi kriteria untuk tindakan lebih lanjut. Misalnya, pesan sukses mungkin memulai tugas untuk memperbarui penyimpanan AWS Systems Manager parameter, atau meluncurkan alur kerja pengujian kepatuhan eksternal untuk AMI keluaran.

Contoh berikut menunjukkan payload JSON untuk pesan tipikal yang diterbitkan Image Builder saat build pipeline berjalan hingga selesai, dan membuat image Linux.

```
{
  "versionlessArn": "arn:aws:imagebuilder:us-west-1:123456789012:image/example-linux-
image",
  "semver": 1237940039285380274899124227,
```

```
"arn": "arn:aws:imagebuilder:us-west-1:123456789012:image/example-linux-
image/1.0.0/3",
"name": "example-linux-image",
"version": "1.0.0",
"type": "AMI",
"buildVersion": 3,
"state": {
  "status": "AVAILABLE"
},
"platform": "Linux",
"imageRecipe": {
  "arn": "arn:aws:imagebuilder:us-west-1:123456789012:image-recipe/example-linux-
image/1.0.0",
  "name": "amjule-barebones-linux",
  "version": "1.0.0",
  "components": [
    {
      "componentArn": "arn:aws:imagebuilder:us-west-1:123456789012:component/update-
linux/1.0.2/1"
    }
  ],
  "platform": "Linux",
  "parentImage": "arn:aws:imagebuilder:us-west-1:987654321098:image/amazon-linux-2-
x86/2022.6.14/1",
  "blockDeviceMappings": [
    {
      "deviceName": "/dev/xvda",
      "ebs": {
        "encrypted": false,
        "deleteOnTermination": true,
        "volumeSize": 8,
        "volumeType": "gp2"
      }
    }
  ],
  "dateCreated": "Feb 24, 2021 12:31:54 AM",
  "tags": {
    "internalId": "1a234567-8901-2345-bcd6-ef7890123456",
    "resourceArn": "arn:aws:imagebuilder:us-west-1:123456789012:image-recipe/example-
linux-image/1.0.0"
  },
  "workingDirectory": "/tmp",
  "accountId": "462045008730"
},
```

```
"sourcePipelineArn": "arn:aws:imagebuilder:us-west-1:123456789012:image-pipeline/
example-linux-pipeline",
"infrastructureConfiguration": {
  "arn": "arn:aws:imagebuilder:us-west-1:123456789012:infrastructure-configuration/
example-linux-infra-config-uswest1",
  "name": "example-linux-infra-config-uswest1",
  "instanceProfileName": "example-linux-ib-baseline-admin",
  "tags": {
    "internalId": "234abc56-d789-0123-a4e5-6b789d012c34",
    "resourceArn": "arn:aws:imagebuilder:us-west-1:123456789012:infrastructure-
configuration/example-linux-infra-config-uswest1"
  },
  "logging": {
    "s3Logs": {
      "s3BucketName": "12345-example-linux-testbucket-uswest1"
    }
  },
  "keyPair": "example-linux-key-pair-uswest1",
  "terminateInstanceOnFailure": true,
  "snsTopicArn": "arn:aws:sns:us-west-1:123456789012:example-linux-ibnotices-
uswest1",
  "dateCreated": "Feb 24, 2021 12:31:55 AM",
  "accountId": "123456789012"
},
"imageTestsConfigurationDocument": {
  "imageTestsEnabled": true,
  "timeoutMinutes": 720
},
"distributionConfiguration": {
  "arn": "arn:aws:imagebuilder:us-west-1:123456789012:distribution-configuration/
example-linux-distribution",
  "name": "example-linux-distribution",
  "dateCreated": "Feb 24, 2021 12:31:56 AM",
  "distributions": [
    {
      "region": "us-west-1",
      "amiDistributionConfiguration": {}
    }
  ],
  "tags": {
    "internalId": "345abc67-8910-12d3-4ef5-67a8b90c12de",
    "resourceArn": "arn:aws:imagebuilder:us-west-1:123456789012:distribution-
configuration/example-linux-distribution"
  },
}
```

```

    "accountId": "123456789012"
  },
  "dateCreated": "Jul 28, 2022 1:13:45 AM",
  "outputResources": {
    "amis": [
      {
        "region": "us-west-1",
        "image": "ami-01a23bc4def5a6789",
        "name": "example-linux-image 2022-07-28T01-14-17.416Z",
        "accountId": "123456789012"
      }
    ]
  },
  "buildExecutionId": "ab0cd12e-34fa-5678-b901-2c3456d789e0",
  "testExecutionId": "6a7b8901-cdef-234a-56b7-8cd89ef01234",
  "distributionJobId": "1f234567-8abc-9d0e-1234-fa56b7c890de",
  "integrationJobId": "432109b8-afe7-6dc5-4321-0ba98f7654e3",
  "accountId": "123456789012",
  "osVersion": "Amazon Linux 2",
  "enhancedImageMetadataEnabled": true,
  "buildType": "USER_INITIATED",
  "tags": {
    "internalId": "901e234f-a567-89bc-0123-d4e567f89a01",
    "resourceArn": "arn:aws:imagebuilder:us-west-1:123456789012:image/example-linux-image/1.0.0/3"
  }
}

```

Contoh berikut menunjukkan payload JSON untuk pesan tipikal yang diterbitkan Image Builder untuk kegagalan pembuatan pipeline untuk image Linux.

```

{
  "versionlessArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-image",
  "semver": 1237940039285380274899124231,
  "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-image/1.0.0/7",
  "name": "My Example Image",
  "version": "1.0.0",
  "type": "AMI",
  "buildVersion": 7,
  "state": {
    "status": "FAILED",
    "reason": "Image Failure reason."
  }
}

```



```

},
"platform": "Linux",
"imageRecipe": {
  "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-
image/1.0.0",
  "name": "My Example Image",
  "version": "1.0.0",
  "description": "Testing Image recipe",
  "components": [
    {
      "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/my-
example-image-component/1.0.0/1"
    }
  ],
"platform": "Linux",
"parentImage": "ami-0cd12345db678d90f",
"dateCreated": "Jun 21, 2022 11:36:14 PM",
"tags": {
  "internalId": "1a234567-8901-2345-bcd6-ef7890123456",
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-
example-image/1.0.0"
},
"accountId": "123456789012"
},
"sourcePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/my-
example-image-pipeline",
"infrastructureConfiguration": {
  "arn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/
my-example-infra-config",
  "name": "SNS topic Infra config",
  "description": "An example that will retain instances of failed builds",
  "instanceTypes": [
    "t2.micro"
  ],
"instanceProfileName": "EC2InstanceProfileForImageBuilder",
"tags": {
  "internalId": "234abc56-d789-0123-a4e5-6b789d012c34",
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-
configuration/my-example-infra-config"
},
"terminateInstanceOnFailure": true,
"snsTopicArn": "arn:aws:sns:us-west-2:123456789012:example-pipeline-notification-
topic",
"dateCreated": "Jul 5, 2022 7:31:53 PM",

```

```

    "accountId": "123456789012"
  },
  "imageTestsConfigurationDocument": {
    "imageTestsEnabled": true,
    "timeoutMinutes": 720
  },
  "distributionConfiguration": {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-example-distribution-config",
    "name": "New distribution config",
    "dateCreated": "Dec 3, 2021 9:24:22 PM",
    "distributions": [
      {
        "region": "us-west-2",
        "amiDistributionConfiguration": {},
        "fastLaunchConfigurations": [
          {
            "enabled": true,
            "snapshotConfiguration": {
              "targetResourceCount": 2
            },
            "maxParallelLaunches": 2,
            "launchTemplate": {
              "launchTemplateId": "lt-01234567890"
            },
            "accountId": "123456789012"
          }
        ]
      }
    ]
  },
  "tags": {
    "internalId": "1fec23a-4f56-7f89-01e2-345678abbe90",
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-example-distribution-config"
  },
  "accountId": "123456789012"
},
"dateCreated": "Jul 5, 2022 7:40:15 PM",
"outputResources": {
  "amis": []
},
"accountId": "123456789012",
"enhancedImageMetadataEnabled": true,
"buildType": "SCHEDULED",

```

```
"tags": {
  "internalId": "456c78b9-0e12-3f45-afb6-7e89b0f1a23b",
  "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-
image/1.0.0/7"
}
```

Produk kepatuhan untuk gambar Image Builder

Dengan standar keamanan yang terus berkembang, dapat menjadi tantangan untuk menjaga kepatuhan dan melindungi organisasi Anda dari ancaman dunia maya. Untuk membantu memastikan bahwa gambar kustom Anda sesuai, dan tetap seperti itu melalui pembaruan otomatis saat penayang merilis versi baru, Image Builder terintegrasi dengan produk dan komponen AWS Marketplace kepatuhan. AWSTOE

Image Builder terintegrasi dengan produk kepatuhan berikut:

- Pusat Pengerasan Tolok Ukur Keamanan Internet (CIS)

Anda dapat menggunakan CIS Hardened Images dan komponen pengerasan CIS terkait untuk membangun gambar khusus yang sesuai dengan pedoman CIS Benchmarks Level 1 terbaru. CIS Hardened Images tersedia di AWS Marketplace Untuk mempelajari lebih lanjut tentang cara mengatur dan menggunakan CIS Hardened Images dan komponen pengerasan, lihat [Panduan Mulai Cepat di portal dukungan](#) situs web CIS.

Note

Saat berlangganan CIS Hardened Image, Anda juga mendapatkan akses ke komponen build terkait yang menjalankan skrip untuk menerapkan pedoman CIS Benchmark Level 1 untuk konfigurasi Anda. Untuk informasi selengkapnya, lihat [Komponen pengerasan CIS](#).

- Panduan Implementasi Teknis Keamanan (STIG)

Untuk kepatuhan STIG, penggunaan dapat menggunakan komponen STIG Amazon-managed AWS Task Orchestrator and Executor (AWSTOE) dalam resep Image Builder Anda. Komponen STIG memindai instans build Anda untuk kesalahan konfigurasi dan menjalankan skrip remediasi untuk memperbaiki masalah yang mereka temukan. Kami tidak dapat menjamin kepatuhan STIG untuk gambar yang Anda buat dengan Image Builder. Anda harus bekerja sama dengan tim kepatuhan organisasi Anda untuk memverifikasi bahwa gambar akhir Anda sesuai. Untuk daftar

lengkap komponen AWSTOE STIG yang dapat Anda gunakan dalam resep Image Builder Anda, lihat [Amazon mengelola komponen pengerasan STIG untuk EC2 Image Builder](#).

Memantau peristiwa dan log di EC2 Image Builder

Untuk menjaga keandalan, ketersediaan, dan kinerja pipeline EC2 Image Builder Anda, penting untuk memantau peristiwa dan log. Peristiwa dan log membantu Anda melihat gambaran besar dan menyelami detail saat panggilan API gagal. Image Builder terintegrasi dengan layanan yang dapat mengirim peringatan dan memulai respons otomatis ketika peristiwa cocok dengan kriteria yang telah Anda konfigurasi.

Topik berikut menjelaskan teknik pemantauan yang dapat Anda gunakan melalui layanan yang terintegrasi dengan Image Builder.

Memantau peristiwa dan log

- [Pencatatan panggilan EC2 Image Builder API menggunakan AWS CloudTrail](#)

Pencatatan panggilan EC2 Image Builder API menggunakan AWS CloudTrail

EC2 Image Builder terintegrasi AWS CloudTrail dengan, layanan yang menyediakan catatan tindakan semua panggilan API untuk diambil oleh pengguna, peran, atau AWS layanan melalui Image Builder API. CloudTrail menangkap Image Builder sebagai peristiwa. Panggilan yang diambil mencakup panggilan dari konsol Image Builder dan panggilan kode ke operasi Image Builder API.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket S3, termasuk peristiwa untuk Image Builder. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk Image Builder, alamat IP dari mana permintaan itu dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat [Panduan AWS CloudTrail Pengguna](#).

Informasi Image Builder di CloudTrail

CloudTrail diaktifkan pada Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di Image Builder, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru

di Akun AWS Anda. Untuk informasi selengkapnya, lihat [Melihat peristiwa dengan Riwayat CloudTrail acara](#).

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

- [Ikhtisar untuk membuat jejak](#).
- [CloudTrail layanan dan integrasi yang didukung](#).
- [Mengonfigurasi notifikasi Amazon SNS](#) untuk CloudTrail
- [Menerima file CloudTrail log dari berbagai wilayah](#).
- [Menerima file CloudTrail log dari beberapa akun](#).

CloudTrail mencatat semua tindakan Image Builder yang didokumentasikan dalam Referensi [API EC2 Image Builder](#). Misalnya, panggilan `createImagePipeline`, `updateInfrastructureConfiguration`, dan `startImagePipelineExecution` tindakan menghasilkan entri dalam file CloudTrail log.

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

- Jika permintaan tersebut dibuat dengan kredensial pengguna root atau IAM.
- Jika permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna federasi.
- Apakah permintaan tersebut dibuat oleh layanan AWS lain.

Untuk informasi selengkapnya tentang menentukan siapa yang meminta acara, lihat elemen [CloudTrail UserIdentity](#).

Keamanan di EC2 Image Builder

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

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

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang berjalan Layanan AWS di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga melakukan pengujian dan verifikasi secara berkala terhadap efektivitas keamanan kami sebagai bagian dari [Program Kepatuhan AWS](#). Untuk mempelajari tentang program kepatuhan yang berlaku untuk EC2 Image Builder, [Layanan AWS lihat di Cakupan oleh Layanan AWS 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 hukum dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Image Builder. Topik berikut menunjukkan cara mengonfigurasi Image Builder untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan Layanan AWS yang lain yang membantu Anda memantau dan mengamankan sumber daya Image Builder Anda.

Topik

- [Perlindungan data di EC2 Image Builder](#)
- [Identity and Access Management untuk EC2 Image Builder](#)
- [Validasi kepatuhan untuk EC2 Image Builder](#)
- [Ketahanan dalam EC2 Image Builder](#)
- [Keamanan infrastruktur di Image Builder](#)
- [Manajemen Patch di EC2 Image Builder](#)
- [Praktik terbaik keamanan untuk EC2 Image Builder](#)

Perlindungan data di EC2 Image Builder

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di EC2 Image Builder. Sebagaimana diuraikan dalam model ini, AWS bertanggung jawab untuk memberikan perlindungan terhadap infrastruktur global yang menjalankan semua AWS Cloud. Anda harus bertanggung jawab untuk memelihara kendali terhadap konten yang di-hosting pada infrastruktur ini. Anda juga bertanggung jawab atas tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat [FAQ Privasi Data](#). Untuk informasi tentang perlindungan data di Eropa, silakan lihat postingan blog [Model Tanggung Jawab Bersama AWS dan GDPR](#) di Blog Keamanan AWS.

Untuk tujuan perlindungan data, sebaiknya Anda melindungi kredensial Akun AWS dan menyiapkan AWS IAM Identity Center atau AWS Identity and Access Management (IAM) untuk pengguna individu. Dengan cara seperti itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugas mereka. Kami juga merekomendasikan agar Anda mengamankan data Anda dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk melakukan komunikasi dengan sumber daya AWS. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan log aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi enkripsi AWS, bersama dengan semua kontrol keamanan default dalam Layanan AWS.
- Gunakan layanan keamanan terkelola lanjutan seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi selengkapnya tentang titik akhir FIPS yang tersedia, silakan lihat [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Sebaiknya Anda tidak memasukkan informasi rahasia atau sensitif, seperti alamat email pelanggan, ke dalam tanda atau bidang teks bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Image Builder atau lainnya Layanan AWS menggunakan konsol, APIAWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang teks bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau diagnostik. Saat Anda memberikan URL ke server eksternal, sebaiknya Anda tidak menyertakan informasi kredensial di URL untuk memvalidasi permintaan Anda ke server tersebut.

Enkripsi dan manajemen kunci di EC2 Image Builder

Image Builder mengenkripsi data saat transit dan diam secara default dengan kunci KMS milik layanan, kecuali untuk yang berikut ini:

- Komponen khusus - Image Builder mengenkripsi komponen kustom dengan kunci KMS default Anda, atau kunci KMS milik layanan.
- Alur kerja gambar - Image Builder dapat mengenkripsi alur kerja gambar Anda dengan kunci yang dikelola pelanggan jika Anda menentukan kunci selama pembuatan alur kerja. Image Builder menangani enkripsi dan dekripsi dengan kunci Anda untuk menjalankan alur kerja yang telah Anda konfigurasi untuk gambar Anda.

Anda dapat mengelola kunci Anda sendiri melalui AWS KMS. Namun, Anda tidak memiliki izin untuk mengelola kunci KMS Image Builder yang dimiliki oleh Image Builder. Untuk informasi selengkapnya tentang mengelola kunci KMS AWS Key Management Service, lihat [Memulai](#) di Panduan AWS Key Management Service Pengembang.

Konteks enkripsi

Untuk memberikan pemeriksaan integritas dan keaslian tambahan pada data terenkripsi Anda, Anda memiliki opsi untuk menyertakan [konteks enkripsi](#) saat mengenkripsi data. Ketika sumber daya dienkripsi dengan konteks enkripsi, secara AWS KMS kriptografis mengikat konteks ke ciphertext. Sumber daya hanya dapat didekripsi jika pemohon memberikan kecocokan yang tepat dan peka huruf besar/kecil untuk konteksnya.

Contoh kebijakan di bagian ini menggunakan konteks enkripsi yang menyerupai Amazon Resource Name (ARN) sumber daya alur kerja Image Builder.

Enkripsi alur kerja gambar dengan kunci terkelola pelanggan

Untuk menambahkan lapisan perlindungan, Anda dapat mengenkripsi sumber daya alur kerja Image Builder dengan kunci terkelola pelanggan Anda sendiri. Jika Anda menggunakan kunci terkelola pelanggan untuk mengenkripsi alur kerja Image Builder yang Anda buat, Anda harus memberikan akses dalam kebijakan kunci agar Image Builder menggunakan kunci Anda saat mengenkripsi dan mendekripsi sumber daya alur kerja. Anda dapat mencabut akses kapan saja. Namun, Image Builder tidak akan memiliki akses ke alur kerja apa pun yang sudah dienkripsi jika Anda mencabut akses ke kunci.

Proses untuk memberikan akses Image Builder untuk menggunakan kunci terkelola pelanggan Anda memiliki dua langkah, sebagai berikut:

Langkah 1: Tambahkan izin kebijakan utama untuk alur kerja Image Builder

Untuk mengaktifkan Image Builder mengenkripsi dan mendekripsi sumber daya alur kerja saat membuat atau menggunakan alur kerja tersebut, Anda harus menentukan izin dalam kebijakan kunci KMS.

Contoh kebijakan kunci ini memberikan akses ke pipeline Image Builder untuk mengenkripsi sumber daya alur kerja selama proses pembuatan, dan mendekripsi sumber daya alur kerja untuk menggunakannya. Kebijakan ini juga memberikan akses kepada administrator kunci. Konteks enkripsi dan spesifikasi sumber daya menggunakan wildcard untuk mencakup semua Wilayah tempat Anda memiliki sumber daya alur kerja.

Sebagai prasyarat untuk menggunakan alur kerja gambar, Anda membuat peran eksekusi alur kerja IAM yang memberikan izin bagi Image Builder untuk menjalankan tindakan alur kerja. Prinsipal untuk pernyataan pertama yang ditampilkan dalam contoh kebijakan kunci di sini harus menentukan peran eksekusi alur kerja IAM Anda.

Untuk informasi selengkapnya tentang kunci yang dikelola pelanggan, lihat [Mengelola akses ke kunci yang dikelola pelanggan](#) di Panduan AWS Key Management Service Pengembang.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow access to build images with encrypted workflow",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/YourImageBuilderExecutionRole"
      },
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:imagebuilder:arn":
            "arn:aws:imagebuilder:*:111122223333:workflow/*"
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    "Sid": "Allow access for key administrators",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:root"
    },
    "Action": [
      "kms:*"
    ],
    "Resource": "arn:aws:kms:*:111122223333:key/"
  }
]
}

```

Langkah 2: Berikan akses kunci ke peran eksekusi alur kerja Anda

Peran IAM yang Image Builder asumsikan untuk menjalankan alur kerja Anda memerlukan izin untuk menggunakan kunci terkelola pelanggan Anda. Tanpa akses ke kunci Anda, Image Builder tidak akan dapat mengenkripsi atau mendekripsi sumber daya alur kerja Anda dengannya.

Edit kebijakan untuk peran eksekusi alur kerja Anda untuk menambahkan pernyataan kebijakan berikut.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow access to the workflow key",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/key_ID",
      "Condition": {
        "StringLike": {
          "kms:EncryptionContext:aws:imagebuilder:arn":
            "arn:aws:imagebuilder:*:111122223333:workflow/*"
        }
      }
    }
  ]
}

```

```
]
}
```

AWS CloudTrail cara untuk alur kerja gambar

Contoh berikut menunjukkan AWS CloudTrail entri tipikal untuk mengenkripsi dan mendekripsi alur kerja gambar yang disimpan dengan kunci yang dikelola pelanggan.

Contoh: GenerateDataKey

Contoh ini menunjukkan seperti apa CloudTrail peristiwa saat Image Builder memanggil aksi AWS KMS GenerateDataKey API dari tindakan Image Builder CreateWorkflow API. Image Builder harus mengenkripsi alur kerja baru sebelum membuat sumber daya alur kerja.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "PRINCIPALID1234567890:workflow-role-name",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/workflow-role-name",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "PRINCIPALID1234567890",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2023-11-21T20:29:31Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "imagebuilder.amazonaws.com"
  },
  "eventTime": "2023-11-21T20:31:03Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "imagebuilder.amazonaws.com",
```

```

"userAgent": "imagebuilder.amazonaws.com",
"requestParameters": {
  "encryptionContext": {
    "aws:imagebuilder:arn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/build/sample-encrypted-workflow/1.0.0/*",
    "aws-crypto-public-key": "key value"
  },
  "keyId": "arn:aws:kms:us-west-2:111122223333:alias/ExampleKMSKey",
  "numberOfBytes": 32
},
"responseElements": null,
"requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEEaaaaa",
"eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-EXAMPLEEzzzzz"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}

```

Contoh: Dekripsi

Contoh ini menunjukkan seperti apa CloudTrail peristiwa saat Image Builder memanggil aksi AWS KMS Decrypt API dari tindakan Image Builder GetWorkflow API. Pipeline Image Builder perlu mendekripsi sumber daya alur kerja sebelum mereka dapat menggunakannya.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "PRINCIPALID1234567890:workflow-role-name",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/workflow-role-name",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {

```

```

    "type": "Role",
    "principalId": "PRINCIPALID1234567890",
    "arn": "arn:aws:iam::111122223333:role/Admin",
    "accountId": "111122223333",
    "userName": "Admin"
  },
  "webIdFederationData": {},
  "attributes": {
    "creationDate": "2023-11-21T20:29:31Z",
    "mfaAuthenticated": "false"
  }
},
"invokedBy": "imagebuilder.amazonaws.com"
},
"eventTime": "2023-11-21T20:34:25Z",
"eventSource": "kms.amazonaws.com",
"eventName": "Decrypt",
"awsRegion": "us-west-2",
"sourceIPAddress": "imagebuilder.amazonaws.com",
"userAgent": "imagebuilder.amazonaws.com",
"requestParameters": {
  "keyId": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLEzzzzz",
  "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
  "encryptionContext": {
    "aws:imagebuilder:arn": "arn:aws:imagebuilder:us-west-2:111122223333:workflow/build/
sample-encrypted-workflow/1.0.0/*",
    "aws-crypto-public-key": "ABC123def4567890abc12345678/90dE/F123abcDEF+4567890abc123D
+ef1=="
  }
},
"responseElements": null,
"requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbb",
"eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-EXAMPLEzzzzz"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,

```

```
"recipientAccountId": "111122223333",  
"eventCategory": "Management"  
}
```

Penyimpanan data di EC2 Image Builder

Image Builder tidak menyimpan log Anda di layanan. Semua log disimpan di instans Amazon EC2 Anda yang digunakan untuk membuat gambar, atau di log otomatisasi Systems Manager Anda.

Privasi Lalu Lintas Antar Jaringan di EC2 Image Builder

Koneksi diamankan antara Image Builder dan lokasi lokal, antara AZ dalam AWS Wilayah, dan antar AWS Wilayah melalui HTTPS. Tidak ada koneksi langsung antar akun.

Identity and Access Management untuk EC2 Image Builder

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Bagaimana EC2 Image Builder bekerja dengan IAM](#)
- [Kebijakan berbasis identitas EC2 Image Builder](#)
- [Kebijakan berbasis sumber daya EC2 Image Builder](#)
- [Mengggunakan kebijakan terkelola untuk EC2 Image Builder](#)
- [Mengggunakan peran terkait layanan untuk EC2 Image Builder](#)
- [Memecahkan masalah identitas dan akses EC2 Image Builder](#)

Audiens

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

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

izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Image Builder, lihat [Memecahkan masalah identitas dan akses EC2 Image Builder](#).

Administrator layanan - Jika Anda bertanggung jawab atas sumber daya Image Builder di perusahaan Anda, Anda mungkin memiliki akses penuh ke Image Builder. Tugas Anda adalah menentukan fitur dan sumber daya Image Builder 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 dasar IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM dengan Image Builder, lihat [Bagaimana EC2 Image Builder bekerja dengan IAM](#).

Administrator IAM - Jika Anda administrator IAM, Anda mungkin ingin mempelajari detail tentang cara menulis kebijakan untuk mengelola akses ke Image Builder. Untuk melihat contoh kebijakan berbasis identitas Image Builder yang dapat Anda gunakan di IAM, lihat [Kebijakan berbasis identitas Image Builder](#)

Mengautentikasi dengan identitas

Untuk informasi terperinci tentang cara menyediakan otentikasi bagi orang dan proses di dalam AndaAkun AWS, lihat [Identitas](#) di Panduan Pengguna IAM.

Bagaimana EC2 Image Builder bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Image Builder, pelajari fitur IAM yang tersedia untuk digunakan dengan Image Builder.

Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Image Builder dan AWS layanan lainnya dengan sebagian besar fitur IAM, lihat [AWSlayanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Kebijakan berbasis identitas untuk Image Builder

Mendukung kebijakan berbasis identitas	Ya
--	----

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, misalnya pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol apa yang pengguna tindakan dan peran dapat kerjakan, pada sumber daya mana, dan

dalam keadaan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, silakan lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

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

Contoh kebijakan berbasis identitas untuk Image Builder

Untuk melihat contoh kebijakan berbasis identitas Image Builder, lihat. [Kebijakan berbasis identitas Image Builder](#)

Kebijakan berbasis sumber daya dalam Image Builder

Mendukung kebijakan berbasis sumber daya	Tidak
--	-------

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan-kebijakan berbasis sumber daya adalah kebijakan terpercaya 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, kebijakan tersebut menentukan tindakan apa yang dapat dilakukan oleh pengguna utama yang ditentukan di sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan pengguna utama](#) dalam kebijakan berbasis sumber daya. Pengguna utama dapat mencakup akun, pengguna, peran, pengguna gabungan, atau Layanan AWS.

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

diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, silakan lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

Tindakan kebijakan untuk Image Builder

Mendukung tindakan kebijakan

Ya

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

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

Menyertakan tindakan dalam suatu kebijakan untuk memberikan izin guna melakukan operasi yang terkait.

Untuk melihat daftar tindakan Image Builder, lihat [Tindakan yang ditentukan oleh EC2 Image Builder](#) di Referensi Otorisasi Layanan.

Tindakan kebijakan di Image Builder menggunakan awalan berikut sebelum tindakan:

```
imagebuilder
```

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

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

Untuk melihat contoh kebijakan berbasis identitas Image Builder, lihat. [Kebijakan berbasis identitas Image Builder](#)

Sumber daya kebijakan untuk Image Builder

Mendukung sumber daya kebijakan	Ya
---------------------------------	----

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen kebijakan JSON `Resource` menentukan objek atau objek-objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan entah elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan-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 bagi semua sumber daya.

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

Untuk melihat daftar jenis sumber daya Image Builder dan ARNnya, lihat Sumber [daya yang ditentukan oleh EC2 Image Builder](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditentukan oleh EC2 Image Builder](#).

Untuk melihat contoh kebijakan berbasis identitas Image Builder, lihat. [Kebijakan berbasis identitas Image Builder](#)

Kunci kondisi kebijakan untuk Image Builder

Mendukung kunci-kunci persyaratan kebijakan spesifik layanan	Ya
--	----

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

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

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

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

AWS mendukung kunci-kunci syarat global dan kunci-kunci syarat spesifik layanan. Untuk melihat semua kunci persyaratan global AWS, silakan lihat [kunci konteks syarat global AWS](#) di Panduan Pengguna IAM.

Untuk melihat daftar kunci kondisi Image Builder, lihat [Kunci kondisi untuk EC2 Image Builder](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang ditentukan oleh EC2 Image Builder](#).

Untuk melihat contoh kebijakan berbasis identitas Image Builder, lihat. [Kebijakan berbasis identitas Image Builder](#)

ACL di Image Builder

Mendukung ACL

Tidak

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis

sumber daya, meskipun kebijakan-kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Image Builder

Mendukung ABAC (tag dalam kebijakan)	Parsial
--------------------------------------	---------

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Di AWS, atribut-atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak sumber daya AWS. Pemberian tag ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi-operasi ketika 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 dimana pengelolaan kebijakan menjadi rumit.

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

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

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

Menggunakan kredensial sementara dengan Image Builder

Mendukung kredensial temporer	Ya
-------------------------------	----

Beberapa Layanan AWS tidak berfungsi saat Anda masuk dengan menggunakan kredensial temporer. Sebagai informasi tambahan, termasuk tentang Layanan AWS mana saja yang berfungsi dengan kredensial temporer, silakan lihat [Layanan AWS yang berfungsi dengan IAM](#) di Panduan Pengguna IAM.

Anda menggunakan kredensial temporer jika Anda masuk ke AWS Management Console dengan menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Sebagai contoh, ketika Anda mengakses AWS dengan menggunakan tautan masuk tunggal (SSO) milik perusahaan Anda, proses itu secara otomatis akan membuat kredensial temporer. Anda juga akan secara otomatis membuat kredensial temporer ketika Anda masuk ke konsol sebagai seorang pengguna dan kemudian beralih peran. Untuk informasi selengkapnya tentang peralihan peran, silakan lihat [Peralihan peran \(konsol\)](#) di Panduan Pengguna IAM.

Anda dapat secara manual membuat kredensial temporer menggunakan AWS CLI atau API AWS. Anda kemudian dapat menggunakan kredensial temporer tersebut untuk mengakses AWS. AWS menyarankan agar Anda secara dinamis membuat kredensial temporer alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, silakan lihat [Kredensial keamanan temporer di IAM](#).

Izin utama lintas layanan untuk Image Builder

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

Saat Anda menggunakan pengguna IAM atau peran IAM untuk mengerjakan tindakan di AWS, Anda akan dianggap sebagai pengguna utama. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian dilanjutkan oleh tindakan lain pada layanan yang berbeda. FAS menggunakan izin dari pengguna utama untuk memanggil Layanan AWS, yang dikombinasikan dengan Layanan AWS yang diminta untuk membuat pengajuan ke layanan hilir. Permintaan FAS hanya diajukan ketika sebuah layanan menerima pengajuan yang memerlukan interaksi dengan Layanan AWS lain atau sumber daya lain untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, silakan lihat [Meneruskan sesi akses](#).

Peran layanan untuk Image Builder

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

Peran layanan adalah sebuah [peran IAM](#) yang diambil oleh sebuah 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.

⚠ Warning

Mengubah izin untuk peran layanan dapat merusak fungsionalitas Image Builder. Edit peran layanan hanya jika Image Builder memberikan panduan untuk melakukannya.

Peran terkait layanan untuk Image Builder

Mendukung peran yang tertaut dengan layanan Tidak

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

Untuk detail tentang peran terkait layanan Image Builder, lihat [Menggunakan peran terkait layanan untuk EC2 Image Builder](#)

Kebijakan berbasis identitas Image Builder

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, dan juga kondisi di mana tindakan diizinkan atau ditolak. Image Builder mendukung tindakan, sumber daya, dan kunci kondisi tertentu. Untuk informasi tentang semua elemen yang Anda gunakan dalam kebijakan JSON, lihat [Tindakan, Sumber Daya, dan Kunci Kondisi untuk Image Builder Amazon EC2](#) di Panduan Pengguna IAM.

Tindakan

Tindakan kebijakan di Image Builder menggunakan awalan berikut sebelum tindakan: `imagebuilder:`. Pernyataan kebijakan harus memuat elemen `Action` atau `NotAction`. Image Builder mendefinisikan serangkaian tindakannya sendiri yang menjelaskan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma seperti berikut:

```
"Action": [  
    "imagebuilder:action1",
```

```
"imagebuilder:action2"
```

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

```
"Action": "imagebuilder:List*"
```

Untuk melihat daftar tindakan Image Builder, lihat [Tindakan, Sumber Daya, dan Kunci Kondisi Layanan AWS](#) di Panduan Pengguna IAM.

Mengelola kebijakan menggunakan akses

Untuk informasi terperinci tentang cara mengelola akses AWS dengan membuat kebijakan dan melampirkannya ke identitas atau AWS sumber daya IAM, lihat [Kebijakan dan Izin di Panduan Pengguna IAM](#).

Peran IAM yang Anda kaitkan dengan profil instans harus memiliki izin untuk menjalankan komponen build dan test yang disertakan dalam image Anda. Kebijakan peran IAM berikut harus dilampirkan ke peran IAM yang terkait dengan profil instance:

- EC2InstanceProfileForImageBuilder
- EC2InstanceProfileForImageBuilderECRContainerBuilds
- AmazonSSMManagedInstanceCore

Sumber daya

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen kebijakan JSON `Resource` menentukan objek atau objek-objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan entah elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan-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 bagi semua sumber daya.


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

Sumber daya instans Image Builder memiliki Amazon Resource Name (ARN) berikut.

```
arn:aws:imagebuilder:region:account-id:resource:resource-id
```

Untuk informasi lebih lanjut tentang format ARN, lihat [Amazon Resource Name \(ARN\) dan Namespace Layanan AWS](#).

Misalnya, untuk menentukan `i-1234567890abcdef0` instance dalam pernyataan Anda, gunakan ARN berikut.

```
"Resource": "arn:aws:imagebuilder:us-east-1:123456789012:instance/i-1234567890abcdef0"
```

Untuk menentukan semua instance milik akun tertentu, gunakan wildcard (*).

```
"Resource": "arn:aws:imagebuilder:us-east-1:123456789012:instance/*"
```

Beberapa tindakan Image Builder, seperti untuk membuat sumber daya, tidak dapat dilakukan pada sumber daya tertentu. Dalam kasus tersebut, Anda harus menggunakan wildcard (*).

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

Banyak tindakan EC2 Image Builder API melibatkan banyak sumber daya. Untuk menentukan beberapa sumber daya dalam satu pernyataan tunggal, pisahkan ARN dengan koma.

```
"Resource": [
  "resource1",
  "resource2"
```

Kunci syarat

Image Builder menyediakan kunci kondisi khusus layanan dan dukungan menggunakan beberapa kunci kondisi global. Untuk melihat semua kunci syarat global AWS, lihat [Kunci Konteks Syarat Global AWS](#) dalam Panduan Pengguna IAM. Kunci kondisi khusus layanan berikut disediakan.

imagebuilder: CreatedResourceTagKeys

Bekerja dengan [operator string](#).

Gunakan kunci ini untuk memfilter akses dengan adanya kunci tag dalam permintaan. Ini memungkinkan Anda mengelola sumber daya yang dibuat Image Builder.

Ketersediaan — Kunci ini hanya tersedia untuk UpdateInfrastructureConfiguration API CreateInfrastructureConfiguration dan.

imagebuilder:/CreatedResourceTag<key>

Bekerja dengan [operator string](#).

Gunakan kunci ini untuk memfilter akses dengan pasangan nilai kunci tag yang dilampirkan ke sumber daya yang dibuat Image Builder. Ini memungkinkan Anda mengelola sumber daya Image Builder melalui tag yang ditentukan.

Ketersediaan — Kunci ini hanya tersedia untuk UpdateInfrastructureConfiguration API CreateInfrastructureConfiguration dan.

ImageBuilder: EC2 MetadataHttpTokens

Bekerja dengan [operator string](#).

Gunakan kunci ini untuk memfilter akses dengan Persyaratan Token HTTP Metadata Instans EC2 yang ditentukan dalam permintaan.

Nilai untuk kunci ini dapat berupa `optional` atau `required`.

Ketersediaan — Kunci ini hanya tersedia untuk UpdateInfrastructureConfiguration API CreateInfrastructureConfiguration dan.

imagebuilder: StatusTopicArn

Bekerja dengan [operator string](#).

Gunakan kunci ini untuk memfilter akses oleh ARN Topik SNS dalam permintaan pemberitahuan status terminal mana yang akan dipublikasikan.

Ketersediaan — Kunci ini hanya tersedia untuk UpdateInfrastructureConfiguration API CreateInfrastructureConfiguration dan.

Contoh

Untuk melihat contoh kebijakan berbasis identitas Image Builder, lihat. [Kebijakan berbasis identitas EC2 Image Builder](#)

Kebijakan berbasis sumber daya Image Builder

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada sumber daya Image Builder dan dalam kondisi apa. Image Builder mendukung kebijakan izin berbasis sumber daya untuk komponen, gambar, dan resep gambar. Kebijakan berbasis sumber daya mengizinkan Anda memberikan izin penggunaan ke akun lain berdasarkan penggunaan sumber daya. Anda juga dapat menggunakan kebijakan berbasis sumber daya untuk mengizinkan AWS layanan mengakses komponen, gambar, dan resep gambar Anda.

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau entitas IAM di akun lain sebagai [prinsipal di kebijakan berbasis sumber daya](#). Menambahkan pengguna utama akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berbeda Akun AWS, Anda juga harus memberikan izin entitas utama untuk mengakses sumber daya. Berikan izin dengan melampirkan kebijakan berbasis identitas ke entitas tersebut. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi lebih lanjut, lihat [Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

Untuk informasi tentang cara melampirkan kebijakan berbasis sumber daya ke komponen, gambar, atau resep gambar, lihat. [Bagikan sumber daya EC2 Image Builder](#)

Note

Saat Anda memperbarui kebijakan sumber daya menggunakan Image Builder, pembaruan akan muncul di konsol RAM.

Otorisasi berdasarkan tag Image Builder

Anda dapat melampirkan tag ke sumber daya Image Builder atau meneruskan tag dalam permintaan ke Image Builder. Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di [elemen syarat](#) dari sebuah kebijakan dengan menggunakan kunci-kunci persyaratan `imagebuilder:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Untuk informasi selengkapnya tentang menandai resource Image Builder, lihat [Menandai sumber daya \(AWS CLI\)](#).

Peran IAM Image Builder

[IAM role](#) adalah entitas dalam Akun AWS Anda yang memiliki izin khusus.

Menggunakan kredensial sementara dengan Image Builder

Anda dapat menggunakan kredensial sementara untuk masuk dengan gabungan, menjalankan IAM role, atau menjalankan peran lintas akun. Anda memperoleh kredensi keamanan sementara dengan memanggil operasi AWS STS API seperti [AssumeRole](#) atau [GetFederationToken](#)

Peran tertaut layanan

[Peran terkait layanan](#) memungkinkan Layanan AWS untuk mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran tertaut layanan muncul di akun IAM Anda dan dimiliki oleh layanan tersebut. Pengguna dengan akses administratif dapat melihat tetapi tidak mengedit izin untuk peran terkait layanan.

Image Builder mendukung peran terkait layanan. Untuk informasi tentang membuat atau mengelola peran terkait layanan Image Builder, lihat [Menggunakan peran terkait layanan untuk EC2 Image Builder](#)

Peran layanan

Fitur ini memungkinkan layanan untuk menerima [peran layanan](#) atas nama Anda. Peran ini mengizinkan layanan untuk mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran layanan muncul di akun IAM Anda dan dimiliki oleh akun tersebut. Ini berarti bahwa pengguna dengan akses administratif dapat mengubah izin untuk peran ini. Namun, melakukan hal itu dapat merusak fungsionalitas layanan.

Kebijakan berbasis identitas EC2 Image Builder

Topik

- [Praktik terbaik kebijakan berbasis identitas](#)
- [Menggunakan konsol Image Builder](#)

Praktik terbaik kebijakan berbasis identitas

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Image Builder di akun Anda. Tindakan ini mengenakan biaya kepada Anda Akun AWS. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan terkelola AWS dan beralih ke izin dengan hak akses paling rendah – Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan terkelola AWS yang memberikan izin untuk banyak kasus penggunaan umum. Kebijakan terdapat di Akun AWS Anda. Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan AWS yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, silakan lihat [kebijakan-kebijakan terkelola AWS](#) atau [kebijakan-kebijakan terkelola AWS untuk fungsi tugas](#) 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 mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan pengguna IAM untuk mengajukan izin, silakan lihat [Kebijakan dan izin di IAM](#) di Panduan Pengguna IAM.
- Gunakan syarat dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu syarat ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis syarat kebijakan untuk menentukan bahwa semua pengajuan harus dikirim menggunakan SSL. Anda juga dapat menggunakan syarat untuk memberi akses ke tindakan layanan jika digunakan melalui Layanan AWS yang spesifik, seperti AWS CloudFormation. Untuk informasi selengkapnya, silakan lihat [Elemen kebijakan JSON IAM: Syarat](#) di Panduan Pengguna IAM.
- Gunakan Analizer Akses IAM untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – Analizer Akses IAM memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. Analizer Akses IAM menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, silakan lihat [validasi kebijakan Analizer Akses IAM](#) di Panduan Pengguna IAM.
- Memerlukan autentikasi multi-faktor (MFA) – Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Akun AWS Anda, aktifkan MFA untuk keamanan tambahan.

Untuk meminta MFA ketika operasi API dipanggil, tambahkan syarat MFA pada kebijakan Anda. Untuk informasi selengkapnya, silakan lihat [Mengonfigurasi akses API yang diproteksi MFA](#) di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, silakan lihat [Praktik terbaik keamanan di IAM](#) di Panduan Pengguna IAM.

Menggunakan konsol Image Builder

Untuk mengakses konsol EC2 Image Builder, Anda harus memiliki set izin minimum. Izin ini memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Image Builder di Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tersebut tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna IAM atau peran) dengan kebijakan tersebut.

Untuk memastikan bahwa entitas IAM Anda dapat menggunakan konsol Image Builder, Anda harus melampirkan salah satu kebijakan AWS terkelola berikut:

- [AWSImageBuilderReadOnlyAccesskebijakan](#)
- [AWSImageBuilderFullAccesskebijakan](#)

Untuk informasi selengkapnya tentang kebijakan terkelola Image Builder, lihat [Menggunakan kebijakan terkelola untuk EC2 Image Builder](#).

Important

`AWSImageBuilderFullAccessKebijakan` ini diperlukan untuk membuat peran terkait layanan Image Builder. Ketika Anda melampirkan kebijakan ini ke entitas IAM, Anda juga harus melampirkan kebijakan kustom berikut dan menyertakan sumber daya yang ingin Anda gunakan yang tidak ada `imagebuilder` dalam nama sumber daya:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "sns:Publish"
    ],
    "Resource": "sns topic arn"
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:GetInstanceProfile"
    ],
    "Resource": "instance profile role arn"
  },
  {
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "instance profile role arn",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": "ec2.amazonaws.com"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:ListBucket"
    ],
    "Resource": "bucket arn"
  }
]
}
```

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang sedang Anda coba lakukan.

Kebijakan berbasis sumber daya EC2 Image Builder

Untuk informasi tentang cara membuat komponen, lihat [Mengelola komponen dengan Image Builder](#).

Membatasi akses komponen Image Builder ke alamat IP tertentu

Contoh berikut memberikan izin kepada setiap pengguna untuk melakukan operasi Image Builder pada komponen. Namun, permintaan harus berasal dari rentang alamat IP yang ditentukan dalam syarat.

Kondisi dalam pernyataan ini mengidentifikasi rentang 54.240.143.* alamat IP Internet Protocol versi 4 (IPv4) yang diizinkan, dengan satu pengecualian: 54.240.143.188.

Blok Condition menggunakan syarat IpAddress dan NotIpAddress dan kunci syarat `aws:SourceIp`, yang merupakan kunci syarat lingkup AWS. Untuk informasi selengkapnya tentang kunci kondisi ini, lihat [Menentukan Ketentuan dalam Kebijakan](#). Nilai IPv4 `aws:sourceIp` menggunakan notasi CIDR standar. Untuk informasi lebih lanjut, lihat [Operator Syarat Alamat IP](#) dalam Panduan Pengguna IAM.

```
{
  "Version": "2012-10-17",
  "Id": "IBPolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "imagebuilder.GetComponent:*",
      "Resource": "arn:aws:imagebuilder::examplecomponent/*",
      "Condition": {
        "IpAddress": {"aws:SourceIp": "54.240.143.0/24"},
        "NotIpAddress": {"aws:SourceIp": "54.240.143.188/32"}
      }
    }
  ]
}
```

Menggunakan kebijakan terkelola untuk EC2 Image Builder

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

Perlu diingat bahwa kebijakan terkelola AWS mungkin tidak memberikan izin hak akses paling rendah untuk kasus penggunaan khusus Anda karena tersedia untuk digunakan semua pelanggan AWS.

Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan [kebijakan yang dikelola pelanggan](#) yang khusus untuk kasus penggunaan Anda.

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

Untuk informasi selengkapnya, silakan lihat [kebijakan terkelola AWS](#) di Panduan Pengguna IAM.

AWSImageBuilderFullAccesskebijakan

AWSImageBuilderFullAccessKebijakan ini memberikan akses penuh ke resource Image Builder untuk peran yang dilampirkan, memungkinkan peran untuk mencantumkan, mendeskripsikan, membuat, memperbarui, dan menghapus resource Image Builder. Kebijakan ini juga memberikan izin yang ditargetkan untuk terkait Layanan AWS yang diperlukan, misalnya, untuk memverifikasi sumber daya, atau untuk menampilkan sumber daya saat ini untuk akun di. AWS Management Console

Detail izin

Kebijakan ini mencakup izin berikut:

- Image Builder — Akses administratif diberikan, sehingga peran dapat mencantumkan, mendeskripsikan, membuat, memperbarui, dan menghapus sumber daya Image Builder.
- Amazon EC2 — Akses diberikan untuk Amazon EC2. Jelaskan tindakan yang diperlukan untuk memverifikasi keberadaan sumber daya atau mendapatkan daftar sumber daya milik akun.
- IAM — Akses diberikan untuk mendapatkan dan menggunakan profil instance yang namanya berisi “imagebuilder”, untuk memverifikasi keberadaan peran terkait layanan Image Builder melalui aksi `iam:GetRole` API, dan untuk membuat peran terkait layanan Image Builder.
- License Manager — Akses diberikan untuk membuat daftar konfigurasi lisensi atau lisensi untuk sumber daya.
- Amazon S3 - Akses diberikan untuk daftar bucket milik akun, dan juga bucket Image Builder dengan “imagebuilder” dalam nama mereka.
- Amazon SNS - Izin tulis diberikan kepada Amazon SNS untuk memverifikasi kepemilikan topik untuk topik yang berisi “imagebuilder”.

Contoh kebijakan

Berikut ini adalah contoh `AWSImageBuilderFullAccess` kebijakan tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "imagebuilder:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:ListTopics"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:Publish"
      ],
      "Resource": "arn:aws:sns:*:*:*imagebuilder*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "license-manager:ListLicenseConfigurations",
        "license-manager:ListLicenseSpecificationsForResource"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole"
      ],
      "Resource": "arn:aws:iam::*:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder"
    },
  ],
}
```

```
{
  "Effect": "Allow",
  "Action": [
    "iam:GetInstanceProfile"
  ],
  "Resource": "arn:aws:iam::*:instance-profile/*imagebuilder*"
},
{
  "Effect": "Allow",
  "Action": [
    "iam:ListInstanceProfiles",
    "iam:ListRoles"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": "iam:PassRole",
  "Resource": [
    "arn:aws:iam::*:instance-profile/*imagebuilder*",
    "arn:aws:iam::*:role/*imagebuilder*"
  ],
  "Condition": {
    "StringEquals": {
      "iam:PassedToService": "ec2.amazonaws.com"
    }
  }
},
{
  "Effect": "Allow",
  "Action": [
    "s3:ListAllMyBuckets",
    "s3:GetBucketLocation"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "s3:ListBucket"
  ],
  "Resource": "arn:aws:s3:::*imagebuilder*"
},
{
```

```

        "Action": "iam:CreateServiceLinkedRole",
        "Effect": "Allow",
        "Resource": "arn:aws:iam::*:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder",
        "Condition": {
            "StringLike": {
                "iam:AWSServiceName": "imagebuilder.amazonaws.com"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action": [
            "ec2:DescribeImages",
            "ec2:DescribeSnapshots",
            "ec2:DescribeVpcs",
            "ec2:DescribeRegions",
            "ec2:DescribeVolumes",
            "ec2:DescribeSubnets",
            "ec2:DescribeKeyPairs",
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeInstanceTypeOfferings",
            "ec2:DescribeLaunchTemplates"
        ],
        "Resource": "*"
    }
]
}

```

AWSImageBuilderReadOnlyAccesskebijakan

AWSImageBuilderReadOnlyAccessKebijakan ini menyediakan akses hanya-baca ke semua resource Image Builder. Izin diberikan untuk memverifikasi bahwa peran terkait layanan Image Builder ada melalui tindakan API. `iam:GetRole`

Detail izin

Kebijakan ini mencakup izin berikut:

- Image Builder - Akses diberikan untuk akses hanya-baca ke sumber daya Image Builder.
- IAM — Akses diberikan untuk memverifikasi keberadaan peran terkait layanan Image Builder melalui tindakan API. `iam:GetRole`

Contoh kebijakan

Berikut ini adalah contoh `AWSImageBuilderReadOnlyAccess` kebijakan tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "imagebuilder:Get*",
        "imagebuilder:List*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole"
      ],
      "Resource": "arn:aws:iam::*:role/aws-service-role/
imagebuilder.amazonaws.com/AWSServiceRoleForImageBuilder"
    }
  ]
}
```

AWSServiceRoleForImageBuilderkebijakan

`AWSServiceRoleForImageBuilderKebijakan` ini memungkinkan Image Builder menelepon Layanan AWS atas nama Anda.

Detail izin

Kebijakan ini dilampirkan ke peran terkait layanan Image Builder saat peran dibuat melalui Systems Manager. Untuk meninjau izin tertentu yang diberikan, lihat [contoh kebijakan](#) di bagian ini. Untuk informasi selengkapnya tentang peran terkait layanan Image Builder, lihat. [Menggunakan peran terkait layanan untuk EC2 Image Builder](#)

Kebijakan ini mencakup izin berikut:

- CloudWatch Log — Akses diberikan untuk membuat dan mengunggah CloudWatch Log ke grup log mana pun yang namanya dimulai dengan `/aws/imagebuilder/`.

- Amazon EC2 — Akses diberikan kepada Image Builder untuk membuat gambar dan meluncurkan instans EC2 di akun Anda, menggunakan snapshot, volume, antarmuka jaringan, subnet, grup keamanan, konfigurasi lisensi, dan pasangan kunci terkait sesuai kebutuhan, selama gambar, instance, dan volume yang sedang dibuat atau digunakan ditandai dengan atau. `CreatedBy: EC2 Image Builder` `CreatedBy: EC2 Fast Launch`

Image Builder dapat memperoleh informasi tentang gambar Amazon EC2, atribut instans, status instans, jenis instans yang tersedia untuk akun Anda, templat peluncuran, subnet, host, dan tag di sumber daya Amazon EC2 Anda.

Image Builder dapat memperbarui pengaturan gambar untuk mengaktifkan atau menonaktifkan peluncuran instance Windows yang lebih cepat di akun Anda, di mana gambar ditandai.

`CreatedBy: EC2 Image Builder`

Selain itu, Image Builder dapat memulai, menghentikan, dan menghentikan instance yang berjalan di akun Anda, membagikan snapshot Amazon EBS, membuat dan memperbarui gambar dan meluncurkan templat, membatalkan pendaftaran gambar yang ada, menambahkan tag, dan mereplikasi gambar di seluruh akun yang telah Anda berikan izin melalui kebijakan. `Ec2ImageBuilderCrossAccountDistributionAccess` Penandaan Image Builder diperlukan untuk semua tindakan ini, seperti yang dijelaskan sebelumnya.

- Amazon ECR — Akses diberikan kepada Image Builder untuk membuat repositori jika diperlukan untuk pemindaian kerentanan gambar kontainer, dan menandai sumber daya yang dibuatnya untuk membatasi ruang lingkup operasinya. Akses juga diberikan kepada Image Builder untuk menghapus gambar kontainer yang dibuat untuk pemindaian setelah mengambil snapshot dari kerentanan.
- EventBridge— Akses diberikan kepada Image Builder untuk membuat dan mengelola EventBridge aturan.
- IAM - Akses diberikan kepada Image Builder untuk meneruskan peran apa pun di akun Anda ke Amazon EC2, dan ke Impor/Ekspor VM.
- Amazon Inspector — Akses diberikan kepada Image Builder untuk menentukan kapan Amazon Inspector menyelesaikan pemindaian instans build, dan mengumpulkan temuan untuk gambar yang dikonfigurasi untuk mengizinkannya.
- AWS KMS— Akses diberikan kepada Amazon EBS untuk mengenkripsi, mendekripsi, atau mengenkripsi ulang volume Amazon EBS. Ini penting untuk memastikan bahwa volume terenkripsi berfungsi saat Image Builder membuat gambar.

- License Manager — Akses diberikan kepada Image Builder untuk memperbarui spesifikasi License Manager melalui `license-manager:UpdateLicenseSpecificationsForResource`.
- Amazon SNS - Izin tulis diberikan untuk topik Amazon SNS apa pun di akun Anda.
- Systems Manager — Akses diberikan kepada Image Builder untuk mencantumkan perintah Systems Manager dan pemanggilannya, entri inventaris, menjelaskan informasi instance dan status eksekusi otomatisasi, dan mendapatkan detail pemanggilan perintah. Image Builder juga dapat mengirim sinyal otomatisasi, dan menghentikan eksekusi otomatisasi untuk sumber daya apa pun di akun Anda.

Image Builder dapat mengeluarkan pemanggilan perintah run ke instance apa pun yang diberi tag `"CreatedBy": "EC2 Image Builder"` untuk file skrip berikut: `AWS-RunPowerShellScript`, `AWS-RunShellScript` atau `AWSEC2-RunSysprep` Image Builder dapat memulai eksekusi otomatisasi Systems Manager di akun Anda untuk dokumen otomatisasi tempat nama dimulai `ImageBuilder`.

Image Builder juga dapat membuat atau menghapus asosiasi State Manager untuk instans apa pun di akun Anda, selama dokumen asosiasi tersebut `AWS-GatherSoftwareInventory`, dan untuk membuat peran terkait layanan Systems Manager di akun Anda.

- AWS STS— Akses diberikan kepada Image Builder untuk mengambil peran yang dinamai `EC2ImageBuilderDistributionCrossAccountRole` dari akun Anda ke akun mana pun yang diizinkan oleh kebijakan Trust tentang peran tersebut. Ini digunakan untuk distribusi gambar lintas akun.

Contoh kebijakan

Berikut ini adalah contoh `AWSServiceRoleForImageBuilder` kebijakan tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "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:*:*:key-pair/*",
        "arn:aws:ec2:*:*:launch-template/*",
        "arn:aws:license-manager:*:*:license-configuration:*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:RunInstances"
    ],
    "Resource": [
        "arn:aws:ec2:*:*:volume/*",
        "arn:aws:ec2:*:*:instance/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/CreatedBy": [
                "EC2 Image Builder",
                "EC2 Fast Launch"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": "iam:PassRole",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [
                "ec2.amazonaws.com",
                "ec2.amazonaws.com.cn",
                "vmie.amazonaws.com"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:StopInstances",
        "ec2:StartInstances",
        "ec2:TerminateInstances"
    ]
}

```



```

    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/CreatedBy": "EC2 Image Builder"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:CopyImage",
        "ec2:CreateImage",
        "ec2:CreateLaunchTemplate",
        "ec2:DeregisterImage",
        "ec2:DescribeImages",
        "ec2:DescribeInstanceAttribute",
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:DescribeInstanceTypeOfferings",
        "ec2:DescribeInstanceTypes",
        "ec2:DescribeSubnets",
        "ec2:DescribeTags",
        "ec2:ModifyImageAttribute",
        "ec2:DescribeImportImageTasks",
        "ec2:DescribeExportImageTasks",
        "ec2:DescribeSnapshots",
        "ec2:DescribeHosts"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:ModifySnapshotAttribute"
    ],
    "Resource": "arn:aws:ec2:*::snapshot/*",
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/CreatedBy": "EC2 Image Builder"
        }
    }
},
{

```

```

    "Effect": "Allow",
    "Action": [
      "ec2:CreateTags"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction": [
          "RunInstances",
          "CreateImage"
        ],
        "aws:RequestTag/CreatedBy": [
          "EC2 Image Builder",
          "EC2 Fast Launch"
        ]
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CreateTags"
    ],
    "Resource": [
      "arn:aws:ec2:*::image/*",
      "arn:aws:ec2:*::export-image-task/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CreateTags"
    ],
    "Resource": [
      "arn:aws:ec2:*::snapshot/*",
      "arn:aws:ec2:*::launch-template/*"
    ],
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/CreatedBy": [
          "EC2 Image Builder",
          "EC2 Fast Launch"
        ]
      }
    }
  }
}

```

```

    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "license-manager:UpdateLicenseSpecificationsForResource"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "sns:Publish"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ssm:ListCommands",
      "ssm:ListCommandInvocations",
      "ssm:AddTagsToResource",
      "ssm:DescribeInstanceInformation",
      "ssm:GetAutomationExecution",
      "ssm:StopAutomationExecution",
      "ssm:ListInventoryEntries",
      "ssm:SendAutomationSignal",
      "ssm:DescribeInstanceAssociationsStatus",
      "ssm:DescribeAssociationExecutions",
      "ssm:GetCommandInvocation"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "ssm:SendCommand",
    "Resource": [
      "arn:aws:ssm:*:*:document/AWS-RunPowerShellScript",
      "arn:aws:ssm:*:*:document/AWS-RunShellScript",
      "arn:aws:ssm:*:*:document/AWSEC2-RunSysprep",
      "arn:aws:s3::*:*"
    ]
  },
  {

```

```

    "Effect": "Allow",
    "Action": [
      "ssm:SendCommand"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:instance/*"
    ],
    "Condition": {
      "StringEquals": {
        "ssm:resourceTag/CreatedBy": [
          "EC2 Image Builder"
        ]
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "ssm:StartAutomationExecution",
    "Resource": "arn:aws:ssm:*:*:automation-definition/ImageBuilder*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ssm:CreateAssociation",
      "ssm>DeleteAssociation"
    ],
    "Resource": [
      "arn:aws:ssm:*:*:document/AWS-GatherSoftwareInventory",
      "arn:aws:ssm:*:*:association/*",
      "arn:aws:ec2:*:*:instance/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncryptFrom",
      "kms:ReEncryptTo",
      "kms:GenerateDataKeyWithoutPlaintext"
    ],
    "Resource": "*",
    "Condition": {
      "ForAllValues:StringEquals": {

```

```
        "kms:EncryptionContextKeys": [
            "aws:ebs:id"
        ]
    },
    "StringLike": {
        "kms:ViaService": [
            "ec2.*.amazonaws.com"
        ]
    }
},
{
    "Effect": "Allow",
    "Action": [
        "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
        "StringLike": {
            "kms:ViaService": [
                "ec2.*.amazonaws.com"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": "kms:CreateGrant",
    "Resource": "*",
    "Condition": {
        "Bool": {
            "kms:GrantIsForAWSResource": true
        },
        "StringLike": {
            "kms:ViaService": [
                "ec2.*.amazonaws.com"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
```

```

    "Resource": "arn:aws:iam::*:role/
EC2ImageBuilderDistributionCrossAccountRole"
  },
  {
    "Effect": "Allow",
    "Action": [
      "logs:CreateLogStream",
      "logs:CreateLogGroup",
      "logs:PutLogEvents"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/imagebuilder/*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CreateLaunchTemplateVersion",
      "ec2:DescribeLaunchTemplates",
      "ec2:ModifyLaunchTemplate",
      "ec2:DescribeLaunchTemplateVersions"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:ExportImage"
    ],
    "Resource": "arn:aws:ec2::*:image/*",
    "Condition": {
      "StringEquals": {
        "ec2:ResourceTag/CreatedBy": "EC2 Image Builder"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:ExportImage"
    ],
    "Resource": "arn:aws:ec2::*:export-image-task/*"
  },
  {
    "Effect": "Allow",
    "Action": [

```

```

        "ec2:CancelExportTask"
    ],
    "Resource": "arn:aws:ec2:*:*:export-image-task/*",
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/CreatedBy": "EC2 Image Builder"
        }
    }
},
{
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:AWSserviceName": [
                "ssm.amazonaws.com",
                "ec2fastlaunch.amazonaws.com"
            ]
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:EnableFastLaunch"
    ],
    "Resource": [
        "arn:aws:ec2:*:*:image/*",
        "arn:aws:ec2:*:*:launch-template/*"
    ],
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/CreatedBy": "EC2 Image Builder"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "inspector2:ListCoverage",
        "inspector2:ListFindings"
    ],
    "Resource": "*"
}

```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:CreateRepository"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/CreatedBy": "EC2 Image Builder"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:TagResource"
      ],
      "Resource": "arn:aws:ecr:*:*:repository/image-builder-*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/CreatedBy": "EC2 Image Builder"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchDeleteImage"
      ],
      "Resource": "arn:aws:ecr:*:*:repository/image-builder-*",
      "Condition": {
        "StringEquals": {
          "ecr:ResourceTag/CreatedBy": "EC2 Image Builder"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:DeleteRule",
        "events:DescribeRule",
        "events:PutRule",
        "events:PutTargets",
```



```

        "events:RemoveTargets"
    ],
    "Resource": [
        "arn:aws:events:*:*:rule/ImageBuilder-*"
    ]
}
]
}

```

Ec2ImageBuilderCrossAccountDistributionAccesskebijakan

Ec2ImageBuilderCrossAccountDistributionAccessKebijakan ini memberikan izin bagi Image Builder untuk mendistribusikan gambar di seluruh akun di Wilayah target. Selain itu, Image Builder dapat mendeskripsikan, menyalin, dan menerapkan tag ke gambar Amazon EC2 apa pun di akun. Kebijakan ini juga memberikan kemampuan untuk memodifikasi izin AMI melalui tindakan `ec2:ModifyImageAttribute` API.

Detail izin

Kebijakan ini mencakup izin berikut:

- Amazon EC2 — Akses diberikan kepada Amazon EC2 untuk mendeskripsikan, menyalin, dan memodifikasi atribut untuk gambar, dan untuk membuat tag untuk gambar Amazon EC2 apa pun di akun.

Contoh kebijakan

Berikut ini adalah contoh `Ec2ImageBuilderCrossAccountDistributionAccess` kebijakan tersebut.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ec2:CreateTags",
      "Resource": "arn:aws:ec2:*:*:image/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeImages",
        "ec2:CopyImage",

```

```

        "ec2:ModifyImageAttribute"
      ],
      "Resource": "*"
    }
  ]
}

```

EC2ImageBuilderLifecycleExecutionPolicykebijakan

EC2ImageBuilderLifecycleExecutionPolicyKebijakan ini memberikan izin bagi Image Builder untuk melakukan tindakan seperti menghentikan, menonaktifkan, atau menghapus sumber daya gambar Image Builder dan sumber daya dasarnya (AMI, snapshot) guna mendukung aturan otomatis untuk tugas pengelolaan siklus hidup gambar.

Detail izin

Kebijakan ini mencakup izin berikut:

- Amazon EC2 — Akses diberikan kepada Amazon EC2 untuk melakukan tindakan berikut untuk Amazon Machine Images (AMI) di akun yang ditandai. `CreatedBy: EC2 Image Builder`
 - Aktifkan dan nonaktifkan AMI.
 - Aktifkan dan nonaktifkan penghentian gambar.
 - Jelaskan dan deregister AMI.
 - Menjelaskan dan memodifikasi atribut gambar AMI.
 - Hapus snapshot volume yang terkait dengan AMI.
 - Ambil tag untuk sumber daya.
 - Menambahkan atau menghapus tag dari AMI untuk penghentian.
- Amazon ECR — Akses diberikan kepada Amazon ECR untuk melakukan tindakan batch berikut pada repositori ECR dengan tag. `LifecycleExecutionAccess: EC2 Image Builder`
Tindakan Batch mendukung aturan siklus hidup gambar kontainer otomatis.
 - `ecr:BatchGetImage`
 - `ecr:BatchDeleteImage`

Akses diberikan pada tingkat repositori untuk repositori ECR yang ditandai dengan.

`LifecycleExecutionAccess: EC2 Image Builder`

- AWSGrup sumber daya - Akses diberikan kepada Image Builder untuk mendapatkan sumber daya berdasarkan tag.

- EC2 Image Builder — Akses diberikan kepada Image Builder untuk menghapus sumber daya gambar Image Builder.

Contoh kebijakan

Berikut ini adalah contoh EC2ImageBuilderLifecycleExecutionPolicy kebijakan tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Ec2ImagePermission",
      "Effect": "Allow",
      "Action": [
        "ec2:EnableImage",
        "ec2:DeregisterImage",
        "ec2:EnableImageDeprecation",
        "ec2:DescribeImageAttribute",
        "ec2:DisableImage",
        "ec2:DisableImageDeprecation"
      ],
      "Resource": "arn:aws:ec2:*::image/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/CreatedBy": "EC2 Image Builder"
        }
      }
    },
    {
      "Sid": "EC2DeleteSnapshotPermission",
      "Effect": "Allow",
      "Action": "ec2:DeleteSnapshot",
      "Resource": "arn:aws:ec2:*::snapshot/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/CreatedBy": "EC2 Image Builder"
        }
      }
    },
    {
      "Sid": "EC2TagsPermission",
      "Effect": "Allow",
      "Action": [
```

```

        "ec2:DeleteTags",
        "ec2:CreateTags"
    ],
    "Resource": [
        "arn:aws:ec2:*::snapshot/*",
        "arn:aws:ec2:*::image/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/DeprecatedBy": "EC2 Image Builder",
            "aws:ResourceTag/CreatedBy": "EC2 Image Builder"
        },
        "ForAllValues:StringEquals": {
            "aws:TagKeys": "DeprecatedBy"
        }
    }
},
{
    "Sid": "ECRImagePermission",
    "Effect": "Allow",
    "Action": [
        "ecr:BatchGetImage",
        "ecr:BatchDeleteImage"
    ],
    "Resource": "arn:aws:ecr:*::repository/*",
    "Condition": {
        "StringEquals": {
            "ecr:ResourceTag/LifecycleExecutionAccess": "EC2 Image Builder"
        }
    }
},
{
    "Sid": "ImageBuilderEC2TagServicePermission",
    "Effect": "Allow",
    "Action": [
        "ec2:DescribeImages",
        "tag:GetResources",
        "imagebuilder:DeleteImage"
    ],
    "Resource": "*"
}
]
}

```

EC2InstanceProfileForImageBuilderkebijakan

EC2InstanceProfileForImageBuilderKebijakan ini memberikan izin minimum yang diperlukan untuk instans EC2 agar dapat bekerja dengan Image Builder. Ini tidak termasuk izin yang diperlukan untuk menggunakan Agen Systems Manager.

Detail izin

Kebijakan ini mencakup izin berikut:

- CloudWatch Log — Akses diberikan untuk membuat dan mengunggah CloudWatch Log ke grup log mana pun yang namanya dimulai dengan `/aws/imagebuilder/`.
- Image Builder - Akses diberikan untuk mendapatkan komponen Image Builder apa pun.
- AWS KMS— Akses diberikan untuk mendekripsi komponen Image Builder, jika dienkripsi melalui AWS KMS
- Amazon S3 — Akses diberikan untuk mendapatkan objek yang disimpan dalam bucket Amazon S3 yang namanya dimulai dengan `.ec2imagebuilder-`

Contoh kebijakan

Berikut ini adalah contoh EC2InstanceProfileForImageBuilder kebijakan tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "imagebuilder:GetComponent"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
```

```

        "kms:EncryptionContextKeys": "aws:imagebuilder:arn",
        "aws:CalledVia": [
            "imagebuilder.amazonaws.com"
        ]
    }
},
{
    "Effect": "Allow",
    "Action": [
        "s3:GetObject"
    ],
    "Resource": "arn:aws:s3:::ec2imagebuilder*"
},
{
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogStream",
        "logs:CreateLogGroup",
        "logs:PutLogEvents"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/imagebuilder/*"
}
]
}

```

EC2InstanceProfileForImageBuilderECRContainerBuildsKebijakan

EC2InstanceProfileForImageBuilderECRContainerBuildsKebijakan ini memberikan izin minimum yang diperlukan untuk instans EC2 saat bekerja dengan Image Builder untuk membuat image Docker, lalu mendaftarkan dan menyimpan gambar di repositori container Amazon ECR. Ini tidak termasuk izin yang diperlukan untuk menggunakan Agen Systems Manager.

Detail izin

Kebijakan ini mencakup izin berikut:

- CloudWatch Log — Akses diberikan untuk membuat dan mengunggah CloudWatch Log ke grup log mana pun yang namanya dimulai dengan `/aws/imagebuilder/`.
- Amazon ECR — Akses diberikan kepada Amazon ECR untuk mendapatkan, mendaftar, dan menyimpan gambar kontainer, dan untuk mendapatkan token otorisasi.
- Image Builder - Akses diberikan untuk mendapatkan komponen Image Builder atau resep wadah.

- AWS KMS— Akses diberikan untuk mendekripsi komponen Image Builder atau resep wadah, jika dienkripsi melalui AWS KMS
- Amazon S3 — Akses diberikan untuk mendapatkan objek yang disimpan dalam bucket Amazon S3 yang namanya dimulai dengan `ec2imagebuilder-`

Contoh kebijakan

Berikut ini adalah contoh `EC2InstanceProfileForImageBuilderECRContainerBuilds` kebijakan tersebut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "imagebuilder:GetComponent",
        "imagebuilder:GetContainerRecipe",
        "ecr:GetAuthorizationToken",
        "ecr:BatchGetImage",
        "ecr:InitiateLayerUpload",
        "ecr:UploadLayerPart",
        "ecr:CompleteLayerUpload",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:PutImage"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "kms:EncryptionContextKeys": "aws:imagebuilder:arn",
          "aws:CalledVia": [
            "imagebuilder.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::ec2imagebuilder*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:CreateLogGroup",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/imagebuilder/*"
    }
  ]
}

```

Image Builder memperbarui kebijakan AWS terkelola

Bagian ini memberikan informasi tentang pembaruan kebijakan AWS terkelola untuk Image Builder sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman [riwayat dokumen](#) Image Builder.

Perubahan	Deskripsi	Tanggal
EC2ImageBuilderLifecycleExecutionPolicy – Kebijakan baru	Image Builder menambahkan EC2ImageBuilderLifecycleExecutionPolicy kebijakan baru yang berisi izin untuk manajemen siklus hidup gambar.	17 November 2023
AWSServiceRoleForImageBuilder – Perbaruan ke kebijakan yang sudah ada	Image Builder membuat perubahan berikut pada peran layanan untuk menyediakan dukungan macOS.	Agustus 28, 2023

Perubahan	Deskripsi	Tanggal
	<ul style="list-style-type: none">• Menambahkan ec2: DescribeHosts aktifkan Image Builder untuk melakukan polling HostID untuk menentukan kapan dalam keadaan valid untuk meluncurkan instance.• Menambahkan ssm:GetCommandInvocation, tindakan API untuk meningkatkan metode yang digunakan Image Builder untuk mendapatkan detail pemanggilan perintah.	

Perubahan	Deskripsi	Tanggal
<p>AWSServiceRoleForImageBuilder – Perbaruan ke kebijakan yang sudah ada</p>	<p>Image Builder membuat perubahan berikut pada peran layanan untuk memungkinkan alur kerja Image Builder mengumpulkan temuan kerentanan untuk build image container AMI dan ECR. Izin baru mendukung fitur deteksi dan pelaporan CVE.</p> <ul style="list-style-type: none">• Menambahkan inspector2: ListCoverage dan inspector 2: untuk memungkinkan ListFindings Image Builder menentukan kapan Amazon Inspector menyelesaikan pemindaian instance pengujian, dan mengumpulkan temuan untuk gambar yang dikonfigurasi untuk mengizinkannya.• Ditambahkan ecr:CreateRepository, dengan persyaratan untuk Image Builder untuk menandai repositori dengan CreatedBy: EC2 Image Builder (). tag-on-create Juga menambahkan ecr:TagResource (diperlukan untuk tag-on-create) dengan batasan CreatedBy tag yang sama, dan batasan	30 Maret 2023

Perubahan	Deskripsi	Tanggal
	<p>tambahan yang memerlukan nama repositori untuk memulai. <code>image-builder-*</code> Batasan nama mencegah eskalasi hak istimewa dan mencegah perubahan pada repositori yang tidak dibuat oleh Image Builder.</p> <ul style="list-style-type: none">• Ditambahkan <code>ecr:BatchDeleteImage</code> untuk repositori ECR ditandai dengan. <code>CreatedBy: EC2 Image Builder</code> Izin ini memerlukan nama repositori untuk memulai. <code>image-builder-*</code>• Menambahkan izin acara untuk Image Builder untuk membuat dan mengelola aturan EventBridge terkelola Amazon yang disertakan <code>ImageBuilder-*</code> dalam nama.	

Perubahan	Deskripsi	Tanggal
<p>AWSServiceRoleForImageBuilder – Perbaruan ke kebijakan yang sudah ada</p>	<p>Image Builder membuat perubahan berikut pada peran layanan:</p> <ul style="list-style-type: none">• Menambahkan lisensi License Manager sebagai sumber daya untuk RunInstance panggilan ec2: untuk memungkinkan pelanggan menggunakan AMI gambar dasar yang terkait dengan konfigurasi lisensi.	22 Maret 2022
<p>AWSServiceRoleForImageBuilder – Perbaruan ke kebijakan yang sudah ada</p>	<p>Image Builder membuat perubahan berikut pada peran layanan:</p> <ul style="list-style-type: none">• Menambahkan izin untuk tindakan EnableFastLaunch API EC2, untuk mengaktifkan dan menonaktifkan peluncuran yang lebih cepat untuk instance Windows.• Memperketat cakupan lebih untuk ec2: CreateTags tindakan dan kondisi tag sumber daya.	Februari 21, 2022

Perubahan	Deskripsi	Tanggal
<p>AWSServiceRoleForImageBuilder – Perbaruan ke kebijakan yang sudah ada</p>	<p>Image Builder membuat perubahan berikut pada peran layanan:</p> <ul style="list-style-type: none">• Menambahkan izin untuk memanggil layanan VMIE untuk mengimpor VM dan membuat AMI dasar darinya.• Cakupan yang diperketat untuk ec2: CreateTags tindakan dan kondisi tag sumber daya.	<p>November 20, 2021</p>
<p>AWSServiceRoleForImageBuilder – Perbaruan ke kebijakan yang sudah ada</p>	<p>Image Builder menambahkan izin baru untuk memperbaiki masalah di mana lebih dari satu asosiasi inventaris menyebabkan pembuatan gambar macet.</p>	<p>Agustus 11, 2021</p>

Perubahan	Deskripsi	Tanggal
AWSImageBuilderFullAccess – Perbaruan ke kebijakan yang sudah ada	Image Builder membuat perubahan berikut pada peran akses penuh: <ul style="list-style-type: none"> • Menambahkan izin untuk mengizinkan <code>ec2:DescribeInstanceTypeOfferings</code>. • Menambahkan izin untuk memanggil <code>ec2:DescribeInstanceTypeOfferings</code> untuk mengaktifkan konsol Image Builder untuk secara akurat mencerminkan jenis instance yang tersedia di akun. 	13 April 2021
Image Builder mulai melacak perubahan	Image Builder mulai melacak perubahan untuk kebijakan AWS terkelolanya.	April 02, 2021

Menggunakan peran terkait layanan untuk EC2 Image Builder

EC2 Image Builder AWS Identity and Access Management menggunakan peran terkait layanan (IAM). Peran terkait layanan adalah jenis unik peran IAM yang ditautkan langsung ke Image Builder. Peran terkait layanan telah ditentukan sebelumnya oleh Image Builder dan menyertakan semua izin yang diperlukan layanan untuk memanggil orang lain Layanan AWS atas nama Anda.

Peran terkait layanan membuat pengaturan Image Builder lebih efisien, karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Image Builder mendefinisikan izin peran terkait layanan, dan kecuali ditentukan lain, hanya Image Builder yang dapat mengambil perannya. Izin

yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin. Kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat [Layanan AWS yang berfungsi dengan IAM](#) dan cari layanan yang memiliki Ya di kolom Peran Tertaut Layanan. Pilih Ya dengan sebuah tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Izin peran terkait layanan untuk Image Builder

Image Builder menggunakan peran `AWSServiceRoleForImageBuilder` terkait layanan untuk memungkinkan EC2 Image Builder mengakses AWS sumber daya atas nama Anda. Peran terkait layanan mempercayai layanan `imagebuilder.amazonaws.com` untuk mengambil peran tersebut.

Anda tidak perlu membuat peran tertaut layanan secara manual. Saat Anda membuat image Image Builder pertama Anda di AWS Management Console, Management Console AWS CLI, atau AWS API, Image Builder akan membuat peran terkait layanan untuk Anda.

Tindakan berikut membuat gambar baru:

- Jalankan wizard pipeline di konsol Image Builder untuk membuat gambar kustom.
- Gunakan salah satu tindakan API berikut, atau AWS CLI perintahnya yang sesuai:
 - Tindakan [CreateImage](#) API (`create-image` dalam AWS CLI).
 - Tindakan [ImportVmImage](#) API (`import-vm-image` dalam AWS CLI).
 - Tindakan [StartImagePipelineExecution](#) API (`start-image-pipeline-execution` dalam AWS CLI).

Important

Jika peran terkait layanan dihapus dari akun Anda, Anda dapat menggunakan proses yang sama untuk membuatnya lagi. Saat Anda membuat sumber daya EC2 Image Builder pertama Anda, Image Builder akan membuat peran terkait layanan untuk Anda lagi.

Untuk melihat izin untuk `AWSServiceRoleForImageBuilder`, lihat [AWSServiceRoleForImageBuilderkebijakan](#) halaman. Untuk mempelajari lebih lanjut tentang mengonfigurasi izin untuk peran terkait layanan, lihat Izin Peran Tertaut [Layanan di Panduan Pengguna IAM](#).

Menghapus peran terkait layanan Image Builder dari akun Anda

Anda dapat menggunakan konsol IAM, theAWS CLI, atau AWS API untuk menghapus peran terkait layanan untuk Image Builder secara manual dari akun Anda. Namun, sebelum Anda melakukan ini, Anda harus memastikan bahwa tidak ada sumber daya Image Builder diaktifkan yang merujuknya.

Note

Jika layanan Image Builder menggunakan peran saat Anda mencoba menghapus sumber daya, penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Bersihkan sumber daya Image Builder yang digunakan oleh **AWSServiceRoleForImageBuilder** peran

1. Verifikasi bahwa tidak ada build pipeline yang berjalan sebelum Anda mulai. Untuk membatalkan build yang sedang berjalan, gunakan `cancel-image-creation` perintah dari fileAWS CLI.

```
aws imagebuilder cancel-image-creation --image-build-version-arn arn:aws:imagebuilder:us-east-1:123456789012:image-pipeline/sample-pipeline
```

2. Ubah semua jadwal pipeline untuk menggunakan proses pembuatan manual, atau hapus jika Anda tidak akan menggunakannya lagi. Untuk informasi selengkapnya tentang menghapus sumber daya, lihat [Hapus sumber EC2 Image Builder](#).

Hapus peran terkait layanan menggunakan IAM

Anda dapat menggunakan konsol IAM, theAWS CLI, atau AWS API untuk menghapus **AWSServiceRoleForImageBuilder** peran dari akun Anda. Untuk informasi selengkapnya, silakan lihat [Menghapus Peran Tertaut Layanan](#) di Panduan Pengguna IAM.

Wilayah yang Didukung untuk peran terkait layanan EC2 Image Builder

Image Builder mendukung penggunaan peran terkait layanan di semua AWS Wilayah tempat layanan tersedia. Untuk daftar AWS Wilayah yang didukung, lihat [AWSWilayah dan Titik Akhir](#).

Memecahkan masalah identitas dan akses EC2 Image Builder

Topik

- [Saya tidak berwenang untuk melakukan tindakan di Image Builder](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Image Builder saya](#)

Saya tidak berwenang untuk melakukan tindakan di Image Builder

Jika Anda menerima pesan galat bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh galat berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya `my-example-widget` rekaan, tetapi tidak memiliki izin `imagebuilder:GetWidget` rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
imagebuilder:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya `my-example-widget` dengan menggunakan tindakan `imagebuilder:GetWidget`.

Jika Anda membutuhkan bantuan, hubungi administrator AWS Anda. Administrator Anda adalah orang yang memberikan kredensial masuk Anda.

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 ke Image Builder.

Sebagian Layanan AWS mengizinkan Anda untuk memberikan peran yang sudah ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran tertaut-layanan. Untuk melakukan tindakan tersebut, Anda harus memiliki izin untuk memberikan peran pada layanan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Image Builder. 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 membutuhkan bantuan, hubungi administrator AWS Anda. Administrator Anda adalah orang yang memberikan kredensial masuk Anda.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Image Builder saya

Anda dapat membuat peran yang dapat digunakan para pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa 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 akses kepada orang ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mengetahui apakah Image Builder mendukung fitur ini, lihat [Bagaimana EC2 Image Builder bekerja dengan IAM](#).
- Untuk mempelajari cara memberikan akses ke sumber daya di seluruh Akun AWS yang Anda miliki, silakan lihat [Menyediakan akses ke pengguna IAM di Akun AWS lainnya yang Anda miliki](#) di Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses ke sumber daya Anda ke pihak ketiga Akun AWS, silakan lihat [Menyediakan akses ke akun Akun AWS yang dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, silakan lihat [Menyediakan akses ke pengguna terautentikasi eksternal \(gabungan identitas\)](#) di Panduan Pengguna IAM .
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Perbedaan IAM role dan kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

Validasi kepatuhan untuk EC2 Image Builder

EC2 Image Builder tidak dalam lingkup program kepatuhan AWS apa pun.

Untuk daftar cakupan program kepatuhan tertentu, lihat [AWS Layanan dalam Lingkup menurut Program Kepatuhan Layanan AWS](#) . Layanan AWS Untuk informasi umum, silakan lihat [Program Kepatuhan AWS](#) .

Anda bisa mengunduh laporan audit pihak ke tiga menggunakan AWS Artifact. Untuk informasi lebih lanjut, lihat [Mengunduh Laporan di AWS Artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan Image Builder ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta undang-undang dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Quick Start Keamanan dan Kepatuhan](#) – Panduan deployment ini membahas pertimbangan arsitektur dan menyediakan langkah-langkah untuk melakukan deployment terhadap lingkungan dasar di AWS yang menjadi fokus keamanan dan kepatuhan.
- [Sumber Daya Kepatuhan AWS](#) – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [Mengevaluasi Sumber Daya dengan Aturan](#) di Panduan Developer AWS Config – Layanan AWS Config menilai seberapa baik konfigurasi sumber daya Anda dalam mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub](#) – Layanan AWS ini memberikan pandangan komprehensif tentang status keamanan Anda dalam AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

Anda dapat menggabungkan produk kepatuhan dari AWS Marketplace atau komponen dari AWS Task Orchestrator and Executor (AWSTOE) ke dalam gambar Image Builder untuk membantu memastikan bahwa gambar Anda sesuai. Untuk informasi selengkapnya, lihat [Produk kepatuhan untuk gambar Image Builder](#).

Ketahanan dalam EC2 Image Builder

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan jaringan berlatensi rendah, throughput yang tinggi, dan sangat redundan. Dengan Availability Zone, Anda dapat mendesain dan mengoperasikan aplikasi dan basis data yang secara otomatis mengalami kegagalan di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Layanan EC2 Image Builder memungkinkan Anda mendistribusikan gambar yang dibangun di satu Wilayah dengan Wilayah lain, memberi mereka ketahanan Multi-wilayah untuk AMI. Tidak ada mekanisme untuk “mencadangkan” pipa gambar, resep, atau komponen. Anda dapat menyimpan resep dan dokumen komponen di luar layanan Image Builder, seperti di bucket Amazon S3.

EC2 Image Builder tidak dapat dikonfigurasi untuk Ketersediaan Tinggi (HA). Anda dapat mendistribusikan gambar ke beberapa Wilayah untuk membuat gambar lebih tersedia.

Untuk informasi selengkapnya tentang Wilayah AWS dan Zona Ketersediaan, silakan lihat [Infrastruktur Global AWS](#).

Keamanan infrastruktur di Image Builder

Jaringan AWS global menyediakan kemampuan keamanan dan mengontrol akses jaringan untuk layanan seperti EC2 Image Builder. Untuk informasi selengkapnya tentang keamanan infrastruktur yang AWS menyediakan layanannya, lihat bagian [Keamanan Infrastruktur](#) di whitepaper Pengantar AWS Keamanan.

Untuk mengirim permintaan melalui jaringan AWS global untuk tindakan Image Builder API, perangkat lunak klien Anda harus mematuhi pedoman keamanan berikut:

- Untuk mengirim permintaan tindakan Image Builder API, perangkat lunak klien harus menggunakan versi Transport Layer Security (TLS) yang didukung.

Note

AWS menghapus dukungan untuk TLS versi 1.0 dan 1.1 secara bertahap. Kami sangat menyarankan agar Anda memperbarui perangkat lunak klien Anda untuk menggunakan TLS versi 1.2 atau yang lebih baru sehingga Anda masih dapat terhubung. Untuk informasi lebih lanjut, lihat [posting Blog AWS Keamanan](#) ini.

- Perangkat lunak klien harus mendukung cipher suite dengan perfect forward secrecy (PFS), seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Sebagian besar sistem saat ini, seperti Java 7 dan yang lebih baru, mendukung mode ini.
- Anda harus menandatangani permintaan API Anda dengan ID kunci akses dan kunci akses rahasia yang terkait dengan prinsipal AWS Identity and Access Management (IAM). Atau Anda dapat menggunakan [AWS Security Token Service](#) (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk permintaan Anda.

Selain itu, instans EC2 yang digunakan Image Builder untuk membuat dan menguji gambar harus memiliki akses ke AWS Systems Manager

Manajemen Patch di EC2 Image Builder

EC2 Image Builder menyediakan Amazon Linux 2 terbaru, Amazon Linux 2023, Red Hat Enterprise Linux (RHEL), CentOS, Ubuntu, SUSE Linux Enterprise Server, dan Windows 2012 R2 dan AMI yang lebih baru sebagai sumber gambar terkelola. Anda mempertahankan tanggung jawab penambalan sistem Amazon EC2, sesuai model tanggung jawab [bersama](#). Jika instans EC2 dalam beban kerja aplikasi Anda dapat dengan mudah diganti, maka mungkin lebih efisien untuk memperbarui AMI dasar dan menerapkan ulang semua node komputasi berdasarkan gambar ini.

Berikut ini adalah dua cara Anda dapat memperbarui Image Builder AMI Anda.

- AWS-menyediakan komponen patching - EC2 Image Builder menyediakan dua komponen build, `update-linux` dan `update-windows`, yang menginstal semua pembaruan sistem operasi yang tertunda. Komponen-komponen ini menggunakan modul `UpdateOS` tindakan. Untuk informasi selengkapnya, lihat [UpdateOS](#). Komponen dapat ditambahkan ke pipeline build image Anda dengan memilihnya dari daftar komponen AWS yang disediakan.
- Komponen build kustom dengan operasi patching — Untuk menginstal atau memperbarui tambalan secara selektif pada sistem operasi AMI yang didukung, Anda dapat membuat komponen Image Builder untuk menginstal tambalan yang diperlukan. Komponen khusus dapat menginstal tambalan menggunakan skrip shell (Bash atau PowerShell), atau dapat menggunakan modul `UpdateOS` tindakan untuk menentukan tambalan untuk instalasi atau pengecualian. Untuk informasi selengkapnya, lihat [Modul aksi didukung oleh manajer AWSTOE komponen](#).

Komponen yang menggunakan modul `UpdateOS` tindakan (Linux dan Windows)

```
schemaVersion: 1.0
phases:
  - name: build
  steps:
    - name: UpdateOS
      action: UpdateOS
```

Komponen yang menggunakan Bash untuk menginstal pembaruan yum

```
schemaVersion: 1.0
phases:
```

```
- name: build
steps:
  - name: InstallYumUpdates
    action: ExecuteBash
    inputs:
      commands:
        - sudo yum update -y
```

Praktik terbaik keamanan untuk EC2 Image Builder

EC2 Image Builder menyediakan sejumlah fitur keamanan untuk dipertimbangkan saat Anda mengembangkan dan menerapkan kebijakan keamanan Anda sendiri. Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik ini mungkin tidak sesuai atau cukup untuk lingkungan Anda, anggap praktik terbaik tersebut sebagai pertimbangan yang membantu dan bukan sebagai rekomendasi.

- Jangan gunakan grup keamanan yang terlalu permisif dalam resep Image Builder.
- Jangan berbagi gambar dengan akun yang tidak Anda percayai.
- Jangan membuat gambar publik yang memiliki data pribadi atau sensitif.
- Terapkan semua patch keamanan Windows atau Linux yang tersedia selama pembuatan gambar.

Kami sangat menyarankan Anda menguji gambar Anda untuk memvalidasi postur keamanan dan tingkat kepatuhan keamanan yang berlaku. Solusi seperti [Amazon Inspector](#) dapat membantu memvalidasi postur keamanan dan kepatuhan gambar.

IMDSv2 untuk pipeline Image Builder

Saat pipeline Image Builder Anda berjalan, ia mengirimkan permintaan HTTP untuk meluncurkan instans EC2 yang digunakan Image Builder untuk membangun dan menguji gambar Anda. Untuk mengonfigurasi versi IMDS yang digunakan pipeline Anda untuk permintaan peluncuran, setel `httpTokens` parameter dalam setelan metadata instans konfigurasi infrastruktur Image Builder.

Note

Sebaiknya Anda mengonfigurasi semua instans EC2 yang diluncurkan Image Builder dari build pipeline untuk menggunakan IMDSv2 sehingga permintaan pengambilan metadata instance memerlukan header token yang ditandatangani.

Untuk informasi selengkapnya tentang konfigurasi infrastruktur Image Builder, lihat [Kelola konfigurasi infrastruktur EC2 Image Builder](#). Untuk informasi selengkapnya tentang opsi metadata instans EC2 untuk image Linux, lihat [Mengonfigurasi opsi metadata instans di Panduan Pengguna Amazon EC2 untuk Instans Linux](#). Untuk gambar Windows, lihat [Mengonfigurasi opsi metadata instans di Panduan Pengguna Amazon EC2 untuk Instans Windows](#).

Diperlukan pembersihan pasca-build

Setelah Image Builder menyelesaikan semua langkah pembuatan untuk image kustom Anda, Image Builder menyiapkan instance build untuk pengujian dan pembuatan gambar. Sebelum mematikan instance build untuk membuat snapshot, Image Builder melakukan pembersihan berikut untuk memastikan keamanan gambar Anda:

Linux

Pipeline Image Builder menjalankan skrip pembersihan untuk membantu memastikan bahwa gambar akhir mengikuti praktik terbaik keamanan, dan untuk menghapus artefak atau pengaturan build apa pun yang tidak boleh terbawa ke snapshot Anda. Namun, Anda dapat melewati bagian skrip, atau mengganti data pengguna sepenuhnya. Oleh karena itu, gambar yang dihasilkan oleh pipeline Image Builder belum tentu sesuai dengan kriteria peraturan tertentu.

Saat pipeline menyelesaikan tahap pembuatan dan pengujiannya, Image Builder secara otomatis menjalankan skrip pembersihan berikut sebelum membuat gambar keluaran.

Important

Jika Anda mengganti data Pengguna dalam resep Anda, skrip tidak berjalan. Dalam hal ini, pastikan bahwa Anda menyertakan perintah dalam data pengguna Anda yang membuat file kosong bernama `perform_cleanup`. Image Builder mendeteksi file ini dan menjalankan skrip pembersihan sebelum membuat gambar baru.

```
#!/bin/bash
if [[ ! -f {{workingDirectory}}/perform_cleanup ]]; then
    echo "Skipping cleanup"
    exit 0
else
    sudo rm -f {{workingDirectory}}/perform_cleanup
fi
```

```
function cleanup() {
    FILES=("$@")
    for FILE in "${FILES[@]"; do
        if [[ -f "$FILE" ]]; then
            echo "Deleting $FILE";
            sudo shred -zuf $FILE;
        fi;
        if [[ -f $FILE ]]; then
            echo "Failed to delete '$FILE'. Failing."
            exit 1
        fi;
    done
};

# Clean up for cloud-init files
CLOUD_INIT_FILES=(
    "/etc/sudoers.d/90-cloud-init-users"
    "/etc/locale.conf"
    "/var/log/cloud-init.log"
    "/var/log/cloud-init-output.log"
)
if [[ -f {{workingDirectory}}/skip_cleanup_clouddinit_files ]]; then
    echo "Skipping cleanup of cloud init files"
else
    echo "Cleaning up cloud init files"
    cleanup "${CLOUD_INIT_FILES[@]}"
    if [[ $( sudo find /var/lib/cloud -type f | sudo wc -l ) -gt 0 ]]; then
        echo "Deleting files within /var/lib/cloud/*"
        sudo find /var/lib/cloud -type f -exec shred -zuf {} \;
    fi;

    if [[ $( sudo ls /var/lib/cloud | sudo wc -l ) -gt 0 ]]; then
        echo "Deleting /var/lib/cloud/*"
        sudo rm -rf /var/lib/cloud/* || true
    fi;
fi;

# Clean up for temporary instance files
INSTANCE_FILES=(
    "/etc/.updated"
    "/etc/aliases.db"
    "/etc/hostname"
```



```
"/var/lib/misc/postfix.aliasedb-stamp"
"/var/lib/postfix/master.lock"
"/var/spool/postfix/pid/master.pid"
"/var/.updated"
"/var/cache/yum/x86_64/2/.gpgkeyschecked.yum"
)
if [[ -f {{workingDirectory}}/skip_cleanup_instance_files ]]; then
    echo "Skipping cleanup of instance files"
else
    echo "Cleaning up instance files"
    cleanup "${INSTANCE_FILES[@]}"
fi;

# Clean up for ssh files
SSH_FILES=(
    "/etc/ssh/ssh_host_rsa_key"
    "/etc/ssh/ssh_host_rsa_key.pub"
    "/etc/ssh/ssh_host_ecdsa_key"
    "/etc/ssh/ssh_host_ecdsa_key.pub"
    "/etc/ssh/ssh_host_ed25519_key"
    "/etc/ssh/ssh_host_ed25519_key.pub"
    "/root/.ssh/authorized_keys"
)
if [[ -f {{workingDirectory}}/skip_cleanup_ssh_files ]]; then
    echo "Skipping cleanup of ssh files"
else
    echo "Cleaning up ssh files"
    cleanup "${SSH_FILES[@]}"
    USERS=$(ls /home/)
    for user in $USERS; do
        echo Deleting /home/"$user"/.ssh/authorized_keys;
        sudo find /home/"$user"/.ssh/authorized_keys -type f -exec shred -zuf {} \;
    done
    for user in $USERS; do
        if [[ -f /home/"$user"/.ssh/authorized_keys ]]; then
            echo Failed to delete /home/"$user"/.ssh/authorized_keys;
            exit 1
        fi;
    done;
fi;

# Clean up for instance log files
```

```
INSTANCE_LOG_FILES=(
  "/var/log/audit/audit.log"
  "/var/log/boot.log"
  "/var/log/dmesg"
  "/var/log/cron"
)
if [[ -f {{workingDirectory}}/skip_cleanup_instance_log_files ]]; then
  echo "Skipping cleanup of instance log files"
else
  echo "Cleaning up instance log files"
  cleanup "${INSTANCE_LOG_FILES[@]}"
fi;

# Clean up for TOE files
if [[ -f {{workingDirectory}}/skip_cleanup_toe_files ]]; then
  echo "Skipping cleanup of TOE files"
else
  echo "Cleaning TOE files"
  if [[ $( sudo find {{workingDirectory}}/TOE_* -type f | sudo wc -l) -gt 0 ]];
  then
    echo "Deleting files within {{workingDirectory}}/TOE_*"
    sudo find {{workingDirectory}}/TOE_* -type f -exec shred -zuf {} \;
  fi
  if [[ $( sudo find {{workingDirectory}}/TOE_* -type f | sudo wc -l) -gt 0 ]];
  then
    echo "Failed to delete {{workingDirectory}}/TOE_*"
    exit 1
  fi
  if [[ $( sudo find {{workingDirectory}}/TOE_* -type d | sudo wc -l) -gt 0 ]];
  then
    echo "Deleting {{workingDirectory}}/TOE_*"
    sudo rm -rf {{workingDirectory}}/TOE_*
  fi
  if [[ $( sudo find {{workingDirectory}}/TOE_* -type d | sudo wc -l) -gt 0 ]];
  then
    echo "Failed to delete {{workingDirectory}}/TOE_*"
    exit 1
  fi
fi

# Clean up for ssm log files
if [[ -f {{workingDirectory}}/skip_cleanup_ssm_log_files ]]; then
  echo "Skipping cleanup of ssm log files"
else
```

```
echo "Cleaning up ssm log files"
if [[ $( sudo find /var/log/amazon/ssm -type f | sudo wc -l) -gt 0 ]]; then
    echo "Deleting files within /var/log/amazon/ssm/*"
    sudo find /var/log/amazon/ssm -type f -exec shred -zuf {} \;
fi
if [[ $( sudo find /var/log/amazon/ssm -type f | sudo wc -l) -gt 0 ]]; then
    echo "Failed to delete /var/log/amazon/ssm"
    exit 1
fi
if [[ -d "/var/log/amazon/ssm" ]]; then
    echo "Deleting /var/log/amazon/ssm/*"
    sudo rm -rf /var/log/amazon/ssm
fi
if [[ -d "/var/log/amazon/ssm" ]]; then
    echo "Failed to delete /var/log/amazon/ssm"
    exit 1
fi
fi

if [[ $( sudo find /var/log/sa/sa* -type f | sudo wc -l ) -gt 0 ]]; then
    echo "Deleting /var/log/sa/sa*"
    sudo shred -zuf /var/log/sa/sa*
fi
if [[ $( sudo find /var/log/sa/sa* -type f | sudo wc -l ) -gt 0 ]]; then
    echo "Failed to delete /var/log/sa/sa*"
    exit 1
fi

if [[ $( sudo find /var/lib/dhclient/dhclient*.lease -type f | sudo wc -l ) -gt
0 ]]; then
    echo "Deleting /var/lib/dhclient/dhclient*.lease"
    sudo shred -zuf /var/lib/dhclient/dhclient*.lease
fi
if [[ $( sudo find /var/lib/dhclient/dhclient*.lease -type f | sudo wc -l ) -gt
0 ]]; then
    echo "Failed to delete /var/lib/dhclient/dhclient*.lease"
    exit 1
fi

if [[ $( sudo find /var/tmp -type f | sudo wc -l) -gt 0 ]]; then
    echo "Deleting files within /var/tmp/*"
    sudo find /var/tmp -type f -exec shred -zuf {} \;
fi
```

```
if [[ $( sudo find /var/tmp -type f | sudo wc -l) -gt 0 ]]; then
    echo "Failed to delete /var/tmp"
    exit 1
fi
if [[ $( sudo ls /var/tmp | sudo wc -l ) -gt 0 ]]; then
    echo "Deleting /var/tmp/*"
    sudo rm -rf /var/tmp/*
fi

# Shredding is not guaranteed to work well on rolling logs

if [[ -f "/var/lib/rsyslog/imjournal.state" ]]; then
    echo "Deleting /var/lib/rsyslog/imjournal.state"
    sudo shred -zuf /var/lib/rsyslog/imjournal.state
    sudo rm -f /var/lib/rsyslog/imjournal.state
fi

if [[ $( sudo ls /var/log/journal/ | sudo wc -l ) -gt 0 ]]; then
    echo "Deleting /var/log/journal/*"
    sudo find /var/log/journal/ -type f -exec shred -zuf {} \;
    sudo rm -rf /var/log/journal/*
fi

sudo touch /etc/machine-id
```

Windows

Setelah pipeline Image Builder menyesuaikan gambar Windows, ia menjalankan utilitas Microsoft [Sysprep](#). Tindakan ini mengikuti [praktik AWS terbaik untuk pengerasan dan pembersihan gambar](#).

Ganti skrip pembersihan Linux

Image Builder membuat gambar yang aman secara default dan mengikuti praktik terbaik keamanan kami. Namun, beberapa kasus penggunaan yang lebih canggih mungkin mengharuskan Anda untuk melewati satu atau lebih bagian dari skrip pembersihan bawaan. Jika Anda perlu melewatkan beberapa pembersihan, kami sangat menyarankan Anda menguji AMI keluaran Anda untuk memastikan keamanan gambar Anda.

⚠ Important

Melewati bagian dalam skrip pembersihan dapat menghasilkan informasi sensitif, seperti detail akun pemilik atau kunci SSH yang disertakan dalam gambar akhir, dan dalam hal apa pun diluncurkan dari gambar itu. Anda mungkin juga mengalami masalah dengan peluncuran di Availability Zone, Region, atau akun yang berbeda.

Tabel berikut menguraikan bagian skrip pembersihan, file yang dihapus di bagian itu, dan nama file yang dapat Anda gunakan untuk menandai bagian yang harus dilewati oleh Image Builder. Untuk melewati bagian tertentu dari skrip pembersihan, Anda dapat menggunakan modul tindakan [CreateFile](#) komponen atau perintah dalam data pengguna Anda (jika diganti) untuk membuat file kosong dengan nama yang ditentukan dalam kolom Nama file bagian Lewati.

ℹ Note

File yang Anda buat untuk melewati bagian skrip pembersihan tidak boleh menyertakan ekstensi file. Misalnya, jika Anda ingin melewati `CLOUD_INIT_FILES` bagian skrip, tetapi Anda membuat file bernamaskip_cleanup_cloudinit_files.txt, Image Builder tidak akan mengenali file skip.

Input

Bersihkan bagian	File dihapus	Lewati nama file bagian
<code>CLOUD_INIT_FILES</code>	<code>/etc/sudoers.d/90-cloud-init-users</code> <code>/etc/locale.conf</code> <code>/var/log/cloud-init.log</code> <code>/var/log/cloud-init-output.log</code>	<code>skip_cleanup_cloudinit_files</code>
<code>INSTANCE_FILES</code>	<code>/etc/.updated</code>	<code>skip_cleanup_instances_files</code>

Bersihkan bagian	File dihapus	Lewati nama file bagian
	<pre data-bbox="591 212 956 968">/etc/aliases.db /etc/hostname /var/lib/misc/postfix.aliasesdb-stamp /var/lib/postfix/master.lock /var/spool/postfix/pid/master.pid /var/.updated /var/cache/yum/x86_64/2/.gpgkeychecked.yum</pre>	

Bersihkan bagian	File dihapus	Lewati nama file bagian
SSH_FILES	<pre> /etc/ssh/ssh_host_ rsa_key /etc/ssh/ssh_host_ rsa_key.pub /etc/ssh/ssh_host_ ecdsa_key /etc/ssh/ssh_host_ ecdsa_key.pub /etc/ssh/ssh_host_ ed25519_key /etc/ssh/ssh_host_ ed25519_key.pub /root/.ssh/authori zed_keys /home/<all users>/.s sh/authorized_keys; </pre>	skip_cleanup_ssh_f iles
INSTANCE_LOG_FILES	<pre> /var/log/audit/aud it.log /var/log/boot.log /var/log/dmesg /var/log/cron </pre>	skip_cleanup_insta nce_log_files
TOE_FILES	<pre> {{workingDirectory }}/TOE_* </pre>	skip_cleanup_toe_f iles
SSM_LOG_FILES	<pre> /var/log/amazon/ssm/ * </pre>	skip_cleanup_ssm_l og_files

Memecahkan Masalah EC2 Image Builder

EC2 Image Builder terintegrasi Layanan AWS dengan pemantauan dan pemecahan masalah untuk membantu Anda memecahkan masalah pembuatan gambar. Image Builder melacak dan menampilkan kemajuan untuk setiap langkah dalam proses pembuatan gambar. Selain itu, Image Builder dapat mengekspor log ke lokasi Amazon S3 yang Anda berikan.

[Untuk pemecahan masalah lanjutan, Anda dapat menjalankan perintah dan skrip yang telah ditentukan menggunakan Run Command. AWS Systems Manager](#)

Daftar Isi

- [Memecahkan masalah pembangunan pipa](#)
- [Skenario pemecahan masalah](#)

Memecahkan masalah pembangunan pipa

Jika build pipeline Image Builder gagal, Image Builder menampilkan pesan kesalahan yang menjelaskan kegagalan tersebut. Image Builder juga mengembalikan pesan kegagalan, seperti yang ada di output contoh berikut: `workflow execution ID`

```
Workflow Execution ID: wf-12345abc-6789-0123-abc4-567890123abc failed with reason: ...
```

Image Builder mengatur dan mengarahkan tindakan pembuatan gambar melalui serangkaian langkah yang ditentukan untuk tahapan runtime dalam proses pembuatan gambar standarnya. Tahapan pembuatan dan pengujian dari proses masing-masing memiliki alur kerja yang terkait. Saat Image Builder menjalankan alur kerja untuk membangun atau menguji gambar baru, Image Builder akan menghasilkan sumber daya metadata alur kerja yang melacak detail runtime.

Gambar kontainer memiliki alur kerja tambahan yang berjalan selama distribusi.

Detail penelitian untuk kegagalan instans runtime untuk alur kerja Anda

Untuk memecahkan masalah kegagalan runtime alur kerja Anda, Anda dapat memanggil tindakan [GetWorkflowExecution](#) dan [ListWorkflowStepExecutions](#) API dengan Anda. `workflow execution ID`

Tinjau log runtime alur kerja

- CloudWatch Log Amazon

Image Builder menerbitkan log eksekusi alur kerja terperinci ke grup dan aliran Image Builder CloudWatch Logs berikut:

LogGroup:

```
/aws/imagebuilder/ImageName
```

LogStream (x.x.x/x):

```
ImageVersion/ImageBuildVersion
```

Dengan CloudWatch Log, Anda dapat mencari data log dengan pola filter. Untuk informasi selengkapnya, lihat [Cari data log menggunakan pola filter](#) di Panduan Pengguna CloudWatch Log Amazon.

- AWS CloudTrail

Semua aktivitas build juga masuk CloudTrail jika diaktifkan di akun Anda. Anda dapat memfilter CloudTrail acara berdasarkan sumbernya `imagebuilder.amazonaws.com`. Atau, Anda dapat mencari ID instans Amazon EC2 yang dikembalikan dalam log eksekusi untuk melihat detail selengkapnya tentang eksekusi pipeline.

- Amazon Simple Storage Service (S3)

Jika Anda telah menetapkan nama bucket S3 dan key prefix dalam konfigurasi infrastruktur, jalur log runtime langkah alur kerja mengikuti pola ini:

```
S3://S3BucketName/KeyPrefix/ImageName/ImageVersion/ImageBuildVersion/WorkflowExecutionId/StepName
```

Log yang Anda kirim ke bucket S3 menunjukkan langkah-langkah dan pesan kesalahan untuk aktivitas pada instans EC2 selama proses pembuatan gambar. Log termasuk output log dari manajer komponen, definisi komponen yang dijalankan, dan output rinci (dalam JSON) dari semua langkah yang diambil pada instance. Jika Anda mengalami masalah, Anda harus meninjau file-file `iniapplication.log`, dimulai dengan, untuk mendiagnosis penyebab masalah pada instance.

Secara default, Image Builder mematikan build Amazon EC2 atau instance pengujian yang berjalan saat pipeline gagal. Anda dapat mengubah setelan instans untuk sumber daya konfigurasi infrastruktur yang digunakan pipeline, untuk mempertahankan instance build atau pengujian untuk pemecahan masalah.

Untuk mengubah setelan instans di konsol, Anda harus menghapus kotak centang Hentikan instans pada kegagalan yang terletak di bagian Pengaturan pemecahan masalah sumber daya konfigurasi infrastruktur Anda.

Anda juga dapat mengubah pengaturan instance dengan `update-infrastructure-configuration` perintah di AWS CLI. Tetapkan `terminateInstanceOnFailure` nilai ke `false` dalam file JSON yang direferensikan perintah dengan `--cli-input-json` parameter. Untuk detailnya, lihat [Memperbarui konfigurasi infrastruktur](#).

Skenario pemecahan masalah

Bagian ini mencantumkan skenario pemecahan masalah terperinci berikut:

- [Akses ditolak - kode status 403](#)
- [Waktu build habis saat memverifikasi ketersediaan Agen Systems Manager pada instance build](#)
- [Disk sekunder Windows sedang offline saat diluncurkan](#)
- [Build gagal dengan image dasar yang diperkeras CIS](#)
- [AssertInventoryCollection gagal \(Otomasi Systems Manager\)](#)

Untuk melihat detail skenario, pilih judul skenario untuk mengembangkannya. Anda dapat memiliki beberapa judul diperluas secara bersamaan.

Akses ditolak - kode status 403

Deskripsi

Pembuatan pipeline gagal dengan "AccessDenied: Akses Kode status Ditolak: 403".

Penyebab

Kemungkinan penyebabnya meliputi:

- Profil instance tidak memiliki [izin](#) yang diperlukan untuk mengakses API atau sumber daya komponen.

- Peran profil instance tidak memiliki izin yang diperlukan untuk masuk ke Amazon S3. Paling umum, ini terjadi ketika peran profil instance tidak memiliki PutObjectizin untuk bucket S3 Anda.

Solusi

Tergantung pada penyebabnya, masalah ini dapat diselesaikan sebagai berikut:

- Profil instans tidak memiliki kebijakan terkelola — Tambahkan kebijakan yang hilang ke peran profil instans Anda. Kemudian jalankan pipa lagi.
- Profil instans tidak memiliki izin tulis untuk bucket S3 — Tambahkan kebijakan ke peran profil instans Anda yang memberikan PutObjectizin untuk menulis ke bucket S3 Anda. Kemudian jalankan pipa lagi.

Waktu build habis saat memverifikasi ketersediaan Agen Systems Manager pada instance build

Deskripsi

Pembuatan pipeline gagal dengan “status = TimedOut ”” dan “failure message = 'Step timed out sementara step memverifikasi ketersediaan Agen Systems Manager pada instance target'”.

Penyebab

Kemungkinan penyebabnya meliputi:

- Instance yang diluncurkan untuk melakukan operasi build dan menjalankan komponen tidak dapat mengakses endpoint Systems Manager.
- Profil instance tidak memiliki [izin](#) yang diperlukan.

Solusi

Bergantung pada kemungkinan penyebabnya, masalah ini dapat diselesaikan sebagai berikut:

- Masalah akses, subnet pribadi - Jika Anda membangun subnet pribadi, pastikan Anda telah menyiapkan PrivateLink titik akhir untuk Systems Manager, Image Builder, dan, jika ingin masuk, Amazon S3/. CloudWatch Untuk informasi selengkapnya tentang pengaturan PrivateLink titik akhir, lihat konsep [titik akhir VPC](#) (). AWS PrivateLink

- Izin tidak ada - Tambahkan kebijakan terkelola berikut ke peran terkait layanan IAM untuk Image Builder:
 - EC2 InstanceProfileForImageBuilder
 - ECR EC2 InstanceProfileForImageBuilder ContainerBuilds
 - AmazonSSM ManagedInstanceCore

Untuk informasi selengkapnya tentang peran terkait layanan Image Builder, lihat. [Menggunakan peran terkait layanan untuk EC2 Image Builder](#)

Disk sekunder Windows sedang offline saat diluncurkan

Deskripsi

Ketika jenis instans yang digunakan untuk membangun Image Builder Windows AMI tidak cocok dengan jenis instans yang digunakan untuk diluncurkan dari AMI, masalah dapat terjadi ketika volume non-root offline saat peluncuran. Ini terutama terjadi ketika instance build menggunakan arsitektur yang lebih baru daripada instance peluncuran.

Contoh berikut menunjukkan apa yang terjadi ketika Image Builder AMI dibangun di atas tipe instans EC2 Nitro dan diluncurkan pada instance EC2 Xen:

Jenis instance build: m5.large (Nitro)

Jenis instance peluncuran: t2.medium (Xen)

```
PS C:\Users\Administrator> get-disk
Number  Friendly Name  Serial Number      Health Status  Operational Status  Total
Size   Partition Style
-----
-----
0       AWS PVDISK      vol0abc12d34e567f8a9  Healthy       Online              30
GB     MBR
1       AWS PVDISK      vol1bcd23e45f678a9b0  Healthy       Offline             8
GB     MBR
```

Penyebab

Karena pengaturan default Windows, disk yang baru ditemukan tidak secara otomatis dibawa online dan diformat. Ketika jenis instans diubah pada EC2, Windows memperlakukan ini sebagai disk baru yang ditemukan. Ini karena perubahan driver yang mendasarinya.

Solusi

Kami menyarankan Anda menggunakan sistem tipe instans yang sama saat membuat AMI Windows yang ingin Anda luncurkan. Jangan sertakan jenis instance yang dibangun pada sistem yang berbeda dalam konfigurasi infrastruktur Anda. Jika salah satu jenis instance yang Anda tentukan menggunakan sistem Nitro, maka mereka semua harus menggunakan sistem Nitro.

Untuk informasi selengkapnya tentang instans yang dibangun di sistem Nitro, lihat [Instans yang dibangun di Sistem Nitro di Panduan Pengguna Amazon EC2 untuk Instans Windows](#).

Build gagal dengan image dasar yang diperkeras CIS

Deskripsi

Anda menggunakan image dasar yang diperkeras CIS dan build gagal.

Penyebab

Ketika `/tmp` direktori diklasifikasikan sebagai `noexec`, itu dapat menyebabkan Image Builder gagal.

Solusi

Pilih lokasi yang berbeda untuk direktori kerja Anda di `workingDirectory` bidang resep gambar. Untuk informasi selengkapnya, lihat deskripsi tipe [ImageRecipe](#) data.

AssertInventoryCollection gagal (Otomasi Systems Manager)

Deskripsi

Systems Manager Automation menunjukkan kegagalan dalam langkah `AssertInventoryCollection` otomatisasi.

Penyebab

Anda atau organisasi Anda mungkin telah membuat asosiasi Manajer Negara Bagian Systems Manager yang mengumpulkan informasi inventaris untuk instans EC2. Jika koleksi metadata gambar yang disempurnakan diaktifkan untuk pipeline Image Builder Anda (ini adalah default), Image Builder mencoba membuat asosiasi inventaris baru untuk instance build. Namun, Systems Manager tidak mengizinkan beberapa asosiasi inventaris untuk instans terkelola, dan mencegah asosiasi baru jika sudah ada. Hal ini menyebabkan operasi gagal, dan menghasilkan pembuatan pipa yang gagal.

Solusi

Untuk mengatasi masalah ini, matikan pengumpulan metadata gambar yang disempurnakan, menggunakan salah satu metode berikut:

- Perbarui pipeline gambar Anda di konsol, untuk menghapus kotak centang Aktifkan pengumpulan metadata yang disempurnakan. Simpan perubahan Anda dan jalankan pipeline build.

Untuk informasi selengkapnya tentang memperbarui pipeline gambar AMI menggunakan konsol EC2 Image Builder, [Perbarui saluran pipa gambar AMI \(konsol\)](#) lihat. Untuk informasi selengkapnya tentang memperbarui pipeline image container menggunakan konsol EC2 Image Builder, [Perbarui pipeline gambar kontainer \(konsol\)](#) lihat.

- Anda juga dapat memperbarui pipeline gambar Anda dengan `update-image-pipeline` perintah di `fileAWS CLI`. Untuk melakukan ini, sertakan `EnhancedImageMetadataEnabled` properti dalam file JSON Anda, atur ke `false`. Contoh berikut menunjukkan properti disetel ke `false`.

```
{
  "name": "MyWindows2019Pipeline",
  "description": "Builds Windows 2019 Images",
  "enhancedImageMetadataEnabled": false,
  "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-recipe/2020.12.03",
  "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:infrastructure-configuration/my-example-infrastructure-configuration",
  "distributionConfigurationArn": "arn:aws:imagebuilder:us-west-2:123456789012:distribution-configuration/my-example-distribution-configuration",
  "imageTestsConfiguration": {
    "imageTestsEnabled": true,
    "timeoutMinutes": 60
  },
  "schedule": {
    "scheduleExpression": "cron(0 0 * * SUN *)",
    "pipelineExecutionStartCondition": "EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE"
  },
  "status": "ENABLED"
}
```

Untuk mencegah hal ini terjadi pada pipeline baru, kosongkan kotak centang Aktifkan pengumpulan metadata yang disempurnakan saat Anda membuat pipeline baru menggunakan konsol EC2 Image Builder, atau setel nilai properti dalam file JSON saat Anda `false` membuat pipeline menggunakan `EnhancedImageMetadataEnabled` AWS CLI

Riwayat dokumen untuk panduan pengguna EC2 Image Builder

Tabel berikut menjelaskan perubahan penting pada dokumentasi berdasarkan tanggal. Untuk notifikasi tentang pembaruan-pembaruan dokumentasi ini, Anda dapat berlangganan ke sebuah umpan RSS.

- Versi API: 2023-12-12

Perubahan	Deskripsi	Tanggal
Rilis fitur: Manajemen alur kerja gambar	Dengan alur kerja gambar, Anda memiliki lebih banyak fleksibilitas, visibilitas, dan kontrol atas proses pembuatan gambar. Anda dapat menyesuaikan langkah pembuatan dan pengujian untuk alur kerja, atau Anda dapat menggunakan alur kerja default Image Builder.	Desember 12, 2023
Rilis fitur: Manajemen siklus hidup gambar	Dengan kebijakan dan aturan manajemen siklus hidup gambar, Anda dapat menentukan strategi manajemen sumber daya untuk memastikan bahwa gambar yang sudah ketinggalan zaman dan sumber daya terkait melalui proses penandaan dan penghapusan.	17 November 2023
Pembaruan STIG Q3	Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal ketiga 2023. Selain	5 Oktober 2023

itu pesan diperbarui untuk mengklarifikasi bahwa paket pihak ketiga tidak diinstal secara otomatis, dengan sangat sedikit pengecualian. Semua STIG yang dilewati dicatat.

[Versi STIG Baru](#)

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal kedua 2023.

3 Mei 2023

[Versi STIG Baru](#)

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal pertama 2023. Menambahkan dukungan untuk AL2023.

April 14, 2023

[Perbarui Wilayah yang didukung untuk AWSTOE](#)

AWSTOEMenambahkan dukungan untuk hal-hal berikutWilayah AWS: Asia Pasifik (Hyderabad), Asia Pasifik (Jakarta), Eropa (Zurich), Eropa (Spanyol), dan Timur Tengah (UEA).

13 April 2023

[AWSTOEpembaruan unduhan aplikasi](#)

Memperbarui tanda tangan untuk unduhan AWSTOE instalasi di Windows. Juga memperbarui catatan TLS bahwa unduhan aplikasi dari bucket S3 sekarang memerlukan TLS versi 1.2 atau yang lebih baru.

Maret 31, 2023

Rilis fitur: Alur kerja build yang ditingkatkan	Menambahkan detail runtime untuk build gambar di tab alur kerja baru di detail versi pembuatan gambar. Informasi yang disempurnakan untuk build pemecahan masalah.	30 Maret 2023
Rilis fitur: Deteksi dan pelaporan CVE	Untuk akun yang telah mengaktifkan pemindaian Amazon Inspector, Image Builder dapat menangkap temuan common vulnerability and exposure (CVE) dari Amazon Inspector selama tahap pengujian proses pembuatan gambar baru, termasuk image container yang disimpan di Amazon ECR. Image Builder membuat snapshot dari temuan untuk mendukung analisis detail. Image Builder juga melaporkan jumlah temuan yang dapat difilter berdasarkan akun, pipa, atau gambar, dengan kemampuan untuk menelusuri detail.	30 Maret 2023
Ditambahkan riwayat versi	Menambahkan riwayat versi ke bagian Windows dan Linux.	17 Februari 2023
Versi STIG Baru	Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal keempat 2022.	Februari 1, 2023

Rilis fitur: AWS Marketplace integrasi dan pengerasan CIS	Menambahkan AWS Marketplace integrasi untuk dengan mudah menemukan dan menggunakan gambar berlangganan sebagai dasar untuk gambar kustom baru, termasuk CIS Hardened Images dan komponen CIS Hardening baru dari Center for Internet Security.	Januari 13, 2023
Komponen pengerasan CIS	Ditambahkan komponen pengerasan CIS yang dimiliki dan dipelihara oleh CIS.	Januari 13, 2023
Versi STIG Baru	Memperkenalkan dukungan Ubuntu, versi STIG yang diperbarui, dan menerapkan STIGS untuk rilis kuartal kedua 2022.	20 Juli 2022
Pembaruan dokumen: Navigasi untuk Buat halaman dokumen komponen YAMB	Memindahkan konten dokumen komponen Create YAMB ke halamannya sendiri, dan memperbarui halaman lain untuk mereferensikannya.	Juni 7, 2022
Versi STIG Baru	Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal pertama 2022.	April 25, 2022
Ditambahkan modul ExecuteDocument tindakan	Ditambahkan dokumentasi untuk modul ExecuteDocument tindakan di bawahGeneral execution .	28 Maret 2022

Rilis fitur: Support untuk peluncuran Windows AMI yang lebih cepat	Menambahkan pengaturan konfigurasi distribusi untuk mendukung peluncuran yang lebih cepat untuk AMI Windows.	Februari 21, 2022
Rilis pemeliharaan: Perbarui sidik AWSTOE jari biner	Cetak jempol biner yang diperbarui untuk sertifikat AWSTOE penandatanganan.	Februari 18, 2022
Rilis fitur: Konfigurasi input untuk AWSTOE	Ditambahkan dukungan untuk menggunakan file konfigurasi JSON sebagai input untuk AWSTOE run perintah.	3 Februari 2022
Versi STIG Baru	Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal keempat 2021. Juga menambahkan bagian untuk komponen SCAP Compliance Checker (SCC) baru.	Desember 22, 2021
Rilis fitur: Integrasi Impor/Ekspor VM (VMIE)	Menambahkan dukungan untuk impor VM melalui semua saluran (konsol, API/CLI, dll.), Dan untuk ekspor VM melalui API/CLI. Ekspor VM saat ini tidak tersedia dari konsol Image Builder.	Desember 20, 2021
Rilis fitur: Berbagi AMI untuk AWS Organizations dan OU	Konfigurasi distribusi yang diperbarui untuk menambahkan dukungan untuk berbagi AMI keluaran dengan AWS Organizations dan OU.	24 November 2021

Pembaruan dokumen: Perbarui tahapan dan fase komponen	Konten yang diperluas untuk tahapan komponen di Image Builder, dan bagaimana konten tersebut berinteraksi dengan fase AWSTOE komponen.	22 September 2021
Pembaruan dokumen: Tambahkan konten CloudTrail integrasi	Menambahkan ringkasan pemantauan dan konten CloudTrail integrasi.	September 17, 2021
Versi STIG Baru	Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal ketiga 2021.	September 10, 2021
Rilis fitur: EventBridge Integrasi Amazon	Menambahkan EventBridge dukungan yang memungkinkan Anda menghubungkan Image Builder dengan peristiwa dari terkait Layanan AWS, dan memulai peristiwa berdasarkan aturan yang ditentukan. EventBridge	Agustus 18, 2021
Pembaruan dokumen: Menyusun ulang halaman AWSTOE	AWSTOE Halaman yang disusun ulang untuk kejelasan.	Agustus 11, 2021

Rilis fitur: Komponen parameter dan konfigurasi instance tambahan	Ditambahkan dukungan untuk menentukan parameter untuk menyesuaikan komponen untuk resep. Konfigurasi instans EC2 yang diperluas yang digunakan untuk membangun dan menguji gambar, termasuk kemampuan untuk menentukan perintah yang akan dijalankan saat peluncuran, dan kontrol lebih besar atas instalasi dan penghapusan agen Systems Manager.	7 Juli 2021
Versi STIG baru	Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal kedua 2021.	30 Juni 2021
Peningkatan: Menandai perangkat tambahan	Perpesanan yang ditingkatkan seputar penandaan sumber daya.	25 Juni 2021
Rilis fitur: Luncurkan integrasi template	Menambahkan dukungan untuk menggunakan templat peluncuran Amazon EC2 untuk distribusi AMI di pengaturan Distribusi.	7 April 2021
Rilis fitur: Penyempurnaan pembuatan kontainer	Menambahkan dukungan untuk mengonfigurasi pemetaan perangkat blok dan menentukan AMI untuk digunakan sebagai gambar dasar untuk pembuatan kontainer.	7 April 2021

Versi STIG baru	Diperbarui versi STIG dan menerapkan STIGS.	5 Maret 2021
Perbarui ekspresi cron	Pemrosesan cron Image Builder diperbarui untuk meningkatkan granularitas ekspresi cron ke menit, dan menggunakan mesin penjadwalan cron standar. Contoh diperbarui dengan format baru.	8 Februari 2021
Rilis fitur: Dukungan kontainer	Menambahkan dukungan untuk membuat gambar kontainer Docker menggunakan Image Builder, dengan pendaftaran dan penyimpanan gambar yang dihasilkan di Amazon Elastic Container Registry (Amazon ECR). Konten telah disusun ulang untuk mencerminkan fungsionalitas baru dan mengakomodasi pertumbuhan masa depan.	17 Desember 2020
Dokumentasi cron yang direstrukturisasi	Halaman ini sekarang menyoroti informasi lebih lanjut tentang cara kerja cron dengan pembuatan pipeline Image Builder, dan menyertakan detail tentang waktu UTC. Wildcard yang tidak diizinkan untuk bidang tertentu telah dihapus. Contohnya sekarang termasuk sampel ekspresi untuk konsol dan CLI.	13 November 2020

Versi konsol 2.0: pengeditan saluran yang diperbarui	Perubahan konten dalam memulai dan membuat tutorial pipeline, ditambah halaman kelola pipeline gambar, untuk menggabungkan fitur dan alur konsol baru.	13 November 2020
Versi STIG baru	Diperbarui versi STIG dan menerapkan STIGS. Catatan - format daftar diubah untuk menunjukkan STIG yang diterapkan secara default.	15 Oktober 2020
Support untuk konstruksi perulangan di AWSTOE	Buat konstruksi perulangan untuk menentukan urutan instruksi berulang dalam aplikasi. AWSTOE	29 Juli 2020
Support untuk pengembangan AWSTOE komponen lokal	Kembangkan dan uji komponen gambar secara lokal dengan AWSTOE aplikasi.	28 Juli 2020
AMI terenkripsi	EC2 Image Builder menambahkan dukungan untuk distribusi AMI terenkripsi.	1 Juli 2020
AutoScaling penghentian	Pengakhiran penggunaan AutoScaling untuk meluncurkan instance.	Juni 15, 2020

[Support untuk konektivitas melalui AWS PrivateLink](#)

Anda dapat membuat koneksi pribadi antara VPC dan EC2 Image Builder dengan membuat antarmuka VPC endpoint. Endpoint antarmuka didukung oleh AWS PrivateLink, teknologi yang memungkinkan Anda mengakses Image Builder API secara pribadi tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi Direct AWS Connect. Instans di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi dengan Image Builder API. Lalu lintas antara VPC dan Image Builder Anda tidak meninggalkan jaringan Amazon.

10 Juni 2020

[Versi STIG baru](#)

Diperbarui versi STIG dan menerapkan STIGS.

23 Januari 2020

[Pemecahan Masalah](#)

Menambahkan skenario pemecahan masalah umum.

22 Januari 2020

[Komponen STIG](#)

Anda dapat membuat gambar yang sesuai dengan STIG dengan AWSTOE komponen STIG.

22 Januari 2020

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