aws

Panduan Pengguna

EC2 Image Builder

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

EC2 Image Builder: Panduan Pengguna

Copyright © 2025 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 pelanggan, atau dengan cara apa pun yang merendahkan atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu Image Builder? 1
Fitur Image Builder 2
Sistem operasi yang didukung 3
Format gambar yang didukung 4
Kuota default 4
AWS Wilayah dan Titik Akhir 4
Konsep 5
Harga 8
Terkait Layanan AWS 9
Cara kerja Image Builder 9
Elemen AMI 10
Manajemen komponen 11
Sumber daya dibuat 11
Distribusi 12
Berbagi Sumber Daya 13
Kepatuhan 13
Penentuan versi semantik 13
Menyiapkan 15
Peran terkait layanan Image Builder 15
Persyaratan konfigurasi 16
Repositori kontainer untuk pipa gambar kontainer 17
Host khusus untuk gambar macOS 17
Prasyarat IAM 17
Prasyarat Agen Systems Manager 19
Tutorial Image Builder 20
Bangun gambar pertama Anda 20
Buat komponen khusus dengan parameter input 20
Gunakan parameter Systems Manager dengan Image Builder 21
Wisaya pipa: Buat AMI 21
Langkah 1: Tentukan detail pipa 21
Langkah 2: Pilih resep 22
Langkah 3: Tentukan konfigurasi infrastruktur - opsional 25
Langkah 4: Tentukan pengaturan distribusi - opsional 25
Langkah 5: Tinjau 26

Langkah 6: Bersihkan 26
Wisaya pipa: Buat gambar wadah 28
Langkah 1: Tentukan detail pipa 29
Langkah 2: Pilih resep 29
Langkah 3: Tentukan konfigurasi infrastruktur - opsional 33
Langkah 4: Tentukan pengaturan distribusi - opsional 33
Langkah 5: Tinjau 34
Langkah 6: Bersihkan 34
Komponen khusus dengan parameter 36
Gunakan parameter dalam dokumen komponen YAMAL Anda 37
Mengatur parameter komponen dalam resep Image Builder dari konsol 41
Gunakan parameter gambar dasar dalam resep Anda 42
Langkah 1: Temukan atau buat parameter Parameter Store 42
Langkah 2: Konfigurasikan izin IAM 43
Langkah 3: Buat Resep Gambar yang menggunakan parameter 44
Komponen-komponen 46
Daftar dan lihat komponen 48
Daftar komponen Image Builder 49
Daftar versi pembuatan komponen dari AWS CLI 51
Dapatkan detail komponen dari AWS CLI 52
Dapatkan detail kebijakan komponen dari AWS CLI 52
Gunakan AWS Marketplace komponen 52
Temukan AWS Marketplace komponen 53
Berlangganan AWS Marketplace komponen 54
Gunakan AWS Marketplace komponen dalam resep 55
Gunakan komponen terkelola 56
Paket distributor menginstal aplikasi Windows 56
Pengerasan CIS 61
Komponen pengerasan STIG 62
Kembangkan komponen khusus 104
Membuat dokumen komponen YAMAL 104
Buat komponen kustom dengan Image Builder 108
AWSTOE aplikasi manajer komponen 118
AWSTOE download 119
Wilayah yang Didukung 121
Referensi perintah 123

Pengaturan manual 129
Gunakan kerangka dokumen komponen 142
Modul aksi 196
Konfigurasikan masukan 303
Sumber daya gambar 308
Buat daftar gambar dan versi build 308
Daftar gambar 309
Daftar gambar menunggu tindakan 315
Daftar versi pembuatan gambar 317
Lihat detail sumber daya gambar 320
Melihat detail gambar di konsol Image Builder 320
Dapatkan detail kebijakan gambar dari AWS CLI 328
Buat gambar kustom dengan Image Builder 328
Batalkan pembuatan gambar dari AWS CLI 331
Impor dan ekspor gambar VM 331
Impor VM ke Image Builder 331
Mendistribusikan disk VM dari build gambar Anda dari AWS CLI 336
Impor gambar disk ISO 337
Sistem operasi yang didukung untuk impor gambar disk ISO 337
Prasyarat 337
Impor image disk ISO ke Image Builder 338
Luncurkan instance dari output AMI 342
Kelola temuan keamanan 343
Konfigurasikan pemindaian keamanan 343
Kelola temuan keamanan 345
Bersihkan sumber daya Image Builder 347
Mengelola siklus hidup gambar 348
Prasyarat 349
Membuat peran IAM untuk manajemen siklus hidup Image Builder 350
Membuat peran IAM untuk manajemen siklus hidup lintas akun Image Builder 351
Daftar kebijakan siklus hidup 353
Lihat kebijakan siklus hidup 356
Melihat detail kebijakan siklus hidup di konsol Image Builder 356
Buat kebijakan siklus hidup 358
Buat kebijakan siklus hidup (gambar AMI) 359
Buat kebijakan siklus hidup (gambar kontainer) 362

Cara kerja aturan siklus hidup ... 364
Aturan pengecualian siklus hidup AMI ... 365
Lihat detail aturan manajemen siklus hidup ... 366
Konfigurasikan gambar khusus ... 368
Resep ... 368
Daftar dan lihat resep gambar ... 369
Daftar dan lihat resep wadah ... 371
Buat versi baru dari resep gambar ... 373
Buat versi baru dari resep wadah ... 386
Pembersihan sumber daya ... 396
Konfigurasi infrastruktur ... 396
Daftar dan lihat konfigurasi infrastruktur ... 398
Membuat konfigurasi infrastruktur ... 399
Memperbarui konfigurasi infrastruktur ... 404
AWS PrivateLink Titik akhir VPC ... 406
Pengaturan distribusi ... 411
Daftar dan lihat konfigurasi distribusi ... 413
Buat dan perbarui distribusi AMI ... 415
Membuat dan memperbarui distribusi gambar kontainer ... 428
Mengatur distribusi AMI lintas akun ... 432
Tentukan template peluncuran AMI ... 440
Bagikan sumber daya ... 444
Pemilik sumber daya ... 445
Prasyarat untuk berbagi sumber daya Image Builder ... 445
Konsumen sumber daya ... 446
Buat berbagi sumber daya ... 446
Opsi 1: Buat berbagi sumber daya RAM ... 447
Opsi 2: Terapkan kebijakan sumber daya dan promosikan ke pembagian sumber daya yang ada ... 447
Batalkan pembagian sumber daya ... 450
Memberi tanda pada sumber daya ... 451
Tag sumber daya dari AWS CLI ... 451
Hapus tag sumber daya dari AWS CLI ... 452
Daftar semua tag untuk sumber daya tertentu dari AWS CLI ... 452
Hapus sumber daya ... 454
Konsol: Hapus sumber daya ... 454

Hapus sumber daya (AWS CLI) 456
Alur kerja gambar 458
Kerangka alur kerja: Tahapan 459
Akses layanan 459
Gunakan alur kerja terkelola 460
Daftar alur kerja gambar 460
Buat alur kerja gambar 463
Membuat dokumen alur kerja YAMB 466
Struktur dokumen alur kerja YAMB 467
Tindakan langkah 478
Variabel dinamis 496
Pernyataan bersyarat 500
Kelola saluran pipa 506
Daftar dan lihat saluran pipa 507
Daftar pipa gambar dari AWS CLI 507
Dapatkan detail pipeline gambar dari AWS CLI 507
Membuat dan memperbarui pipeline (AMI) 507
Buat pipeline AMI dari AWS CLI 508
Perbarui pipeline dari konsol 510
Perbarui pipa dari AWS CLI 514
Buat dan perbarui pipeline (kontainer) 515
Buat pipeline dari AWS CLI 516
Perbarui pipeline dari konsol 518
Perbarui pipa dari AWS CLI 521
Konfigurasikan alur kerja pipa 523
Tentukan grup uji untuk alur kerja pengujian 524
Mengatur parameter alur kerja dalam pipeline Image Builder dari konsol 525
Tentukan peran layanan IAM yang digunakan Image Builder untuk menjalankan tindakan alur kerja 525
Jalankan saluran pipa 525
Gunakan ekspresi cron 526
Nilai yang didukung untuk ekspresi cron di Image Builder 527
Contoh ekspresi cron di Image Builder 529
Ekspresi rate 531
Gunakan EventBridge aturan 532
EventBridge istilah 533

Lihat EventBridge aturan untuk pipeline Image Builder 534
Gunakan EventBridge aturan untuk menjadwalkan pembuatan pipeline 534
Integrasikan produk dan layanan 536
Amazon EventBridge 539
Pesan peristiwa yang dikirim Image Builder 540
Amazon Inspector 542
AWS Marketplace 543
AWS Marketplace langganan di Image Builder 544
Temukan produk AWS Marketplace gambar dari konsol Image Builder 545
Gunakan produk AWS Marketplace gambar dalam resep Image Builder 547
Amazon Simple Notification Service 548
Topik SNS Terenkripsi 549
Format pesan SNS 550
Produk kepatuhan 556
Memantau peristiwa dan log 558
CloudTrail log 558
Informasi Image Builder di CloudTrail 558
CloudWatch Log 560
Keamanan di Image Builder 561
Perlindungan data 562
Enkripsi dan Manajemen Kunci 563
Penyimpanan data 568
Privasi Lalu Lintas Antar Jaringan 568
Identity and Access Management 568
Audiens 568
Mengautentikasi dengan identitas 569
Cara Image Builder bekerja dengan kebijakan dan peran IAM 569
Kelola perimeter data 581
Kebijakan Berbasis Identitas 582
Kebijakan Berbasis Sumber Daya 585
Kebijakan terkelola 586
Peran terkait layanan 601
Pemecahan Masalah 603
Validasi kepatuhan 605
Ketahanan 606
Keamanan infrastruktur 607

Manajemen tambalan 608
Praktik terbaik 609
Diperlukan pembersihan pasca-build 610
Ganti skrip pembersihan Linux 621
Memecahkan Masalah Image Builder 625
Memecahkan masalah pembangunan pipa 625
Skenario pemecahan masalah 627
Riwayat dokumen 633
..... dcxlvii

Apa itu Image Builder?

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

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

Langkah 1: Tentukan detail pipa

- Beri nama pipeline Anda dan tambahkan tag.
- Tentukan pengaturan pemindaian metadata dan kerentanan.
- Siapkan jadwal untuk pipeline Anda.

Langkah 2: Sesuaikan gambar Anda

Anda dapat memilih resep yang ada atau membuat yang baru.

- Pilih gambar dasar untuk kustomisasi Anda.
- Tambahkan atau hapus perangkat lunak dari gambar dasar Anda.
- Sesuaikan pengaturan dan skrip dengan komponen build.
- Pilih komponen uji untuk dijalankan.

Langkah 3: Tentukan alur kerja Anda

Alur kerja image menentukan urutan langkah yang dilakukan Image Builder selama tahap pembuatan dan pengujian proses pembuatan gambar. Ini adalah bagian dari keseluruhan kerangka kerja Image Builder.

Langkah 4: Konfigurasikan infrastruktur build

- Pilih peran IAM untuk dikaitkan dengan profil instance untuk instance yang diluncurkan Image Builder selama proses pembuatan gambar.
- Pilih satu atau beberapa jenis instance yang dapat diterapkan saat peluncuran.
- Pilih Topik Amazon Simple Notification Service (SNS) untuk menerima notifikasi dari Image Builder.
- Tentukan VPC, subnet, dan grup keamanan yang berlaku untuk proses pembuatan gambar.
- Pilih setelan pemecahan masalah seperti di mana Image Builder menulis log, dan apakah akan menghentikan instance build pada kegagalan (default) atau tetap berjalan untuk pemecahan masalah lebih lanjut.

Langkah 5: Tentukan distribusi gambar

- Pilih Wilayah AWS tempat Image Builder mendistribusikan Amazon Machine Image (AMI) atau gambar kontainer Anda.
- Jika pipeline Image Builder membuat AMI, Image Builder juga mendukung konfigurasi berikut:
  - Pilih kunci KMS yang akan digunakan untuk enkripsi.
  - Konfigurasikan berbagi AMI di seluruh Akun AWS dan Organizations.
  - Kaitkan lisensi yang dikelola sendiri License Manager dengan gambar terdistribusi Anda.
  - Konfigurasikan template peluncuran untuk gambar Anda.

Fitur 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, lihatAmazon mengelola komponen pengerasan STIG untuk 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 yang disetujui. AMIs

Berbagi sumber daya yang disederhanakan di seluruh Akun AWS

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

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

Untuk informasi selengkapnya, lihat Bagikan sumber daya Image Builder dengan AWS RAM.

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
macOS	12.x (Monterey), 13.x (Ventura), 14.x (Sonoma), 15.x (Sequoia)
Red Hat Enterprise Linux (RHEL)	7, 8, 9, dan 10
SUSE Linux Server Perusahaan (SUSE)	12 dan 15
Ubuntu	18.04 LTS, 20.04 LTS, 22.04 LTS, dan 24.04 LTS
Windows Server	2012 R2, 2016, 2019, 2022, dan 2025

Format gambar yang didukung

Untuk gambar kustom Anda yang membuat Amazon Machine Image (AMI), 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 sebagai titik awal Anda.

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.

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 yang sudah diinstal sebelumnya untuk menyebarkan EC2 instance. Untuk informasi selengkapnya, lihat Gambar Mesin Amazon (AMI).

Pipa gambar

Pipeline gambar menyediakan kerangka kerja otomatisasi untuk membangun gambar yang aman AMIs dan kontainer 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 wadah 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 konsol AWS 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 YAMG 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 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 oleh AWS, atau Anda dapat membuat sendiri. Untuk informasi selengkapnya tentang komponen, lihatBagaimana Image Builder menggunakan AWS Task Orchestrator and Executor aplikasi untuk mengelola komponen.

Dokumen komponen

Dokumen YAML 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.

Harga

Tidak ada biaya untuk menggunakan EC2 Image Builder untuk membuat AMI kustom atau gambar kontainer. 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 sebuah EC2 instance
- Menyimpan log di Amazon S3
- Memvalidasi gambar dengan Amazon Inspector
- Menyimpan Snapshot Amazon EBS untuk Anda AMIs
- Menyimpan gambar kontainer di Amazon ECR
- Mendorong dan menarik gambar kontainer masuk dan keluar dari Amazon ECR
- Jika Tingkat Lanjut Systems Manager diaktifkan, dan EC2 instans Amazon dijalankan dengan aktivasi lokal, Anda mungkin akan dikenakan biaya untuk sumber daya melalui Systems Manager

Terkait Layanan AWS

EC2 Image Builder menggunakan yang lain Layanan AWS untuk membuat gambar, tergantung pada konfigurasi resep Image Builder Anda. Untuk informasi selengkapnya tentang integrasi produk dan layanan untuk gambar kustom Anda, lihatIntegrasikan produk dan layanan di Image Builder.

Cara kerja EC2 Image Builder

Saat Anda menggunakan konsol EC2 Image Builder untuk membuat pipeline gambar kustom, sistem 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 wadah. 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 EC2 instance 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 Anda Akun 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 integration/continuous penerapan berkelanjutan (CI/CD) untuk otomatisasi berulang. Anda dapat berbagi dan membuat versi baru resep Anda.

Isi bagian

- Elemen AMI
- Manajemen komponen
- 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 EC2 instance.

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 EC2 Root Amazon.
- Saat Amazon EBS digunakan, perangkat root adalah volume EBS yang dibuat dari snapshot EBS.
- Luncurkan izin yang menentukan Akun AWS yang dapat diluncurkan VMs 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.

Manajemen komponen

EC2 Image Builder menggunakan aplikasi manajemen komponen AWS Task Orchestrator and Executor (AWSTOE) yang membantu Anda mengatur alur kerja yang kompleks, memodifikasi konfigurasi sistem, dan menguji sistem Anda dengan komponen skrip berbasis YAMM. 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, lihatPengaturan manual untuk mengembangkan komponen khusus 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, lihatGunakan komponen untuk menyesuaikan image Image Builder. Untuk informasi selengkapnya tentang membuat komponen dengan AWSTOE, lihatBagaimana Image Builder menggunakan AWS Task Orchestrator and Executor aplikasi untuk mengelola 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.

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

- EC2 contoh (sementara)
- Systems Manager Inventory Association (melalui Systems Manager State Manager jika `EnhancedImageMetadata` Diaktifkan) pada EC2 instance
- Amazon EC2 AMI
- Snapshot Amazon EBS yang terkait dengan Amazon AMI EC2

Pipa gambar kontainer

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

Setelah gambar dibuat, semua sumber daya sementara dihapus.

Distribusi

EC2 Image Builder dapat mendistribusikan AMIs atau mengkontainer gambar ke AWS Wilayah 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 EC2 instance 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 disebarkan ke seluruh Wilayah tempat AMI didistribusikan. AMIs

Anda juga dapat menggunakan AWS Organizations akun Anda untuk memberlakukan batasan pada akun anggota untuk meluncurkan instans hanya dengan disetujui dan sesuai. AMIs Untuk informasi selengkapnya, lihat Mengelola Akun AWS di Organisasi Anda.

Untuk memperbarui setelan distribusi menggunakan konsol Image Builder, ikuti langkah-langkah untukBuat versi resep gambar baru dari konsol, atauBuat versi resep kontainer baru dengan konsol.

Berbagi Sumber Daya

Untuk berbagi komponen, resep, atau gambar dengan akun lain atau di dalamnya AWS Organizations, lihatBagikan sumber daya Image Builder dengan AWS RAM.

Kepatuhan

Untuk Tolok Ukur Center for Internet Security (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, tidak ditambal, konektivitas internet publik CVEs, 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 informasi selengkapnya, lihat Tolok Ukur Pusat Keamanan Internet (CIS).

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. Untuk daftar lengkap komponen STIG yang tersedia melalui Image Builder, lihatAmazon mengelola komponen pengerasan STIG untuk Image Builder.

Pembuatan versi semantik di Image Builder

Image Builder menggunakan versi semantik untuk mengatur sumber daya dan memastikan bahwa mereka memiliki keunikan. IDs 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. Versionless ARNs dan Name ARNs tidak menyertakan nilai spesifik di salah satu node. Node ditinggalkan seluruhnya, atau mereka ditentukan sebagai wildcard, misalnya: x.x.x.
2. Versi hanya ARNs memiliki tiga node pertama:<major>. <minor>. <patch>
3. Versi build ARNs memiliki keempat node, dan arahkan ke build tertentu untuk versi objek tertentu.

Penugasan: Untuk tiga node pertama Anda dapat menetapkan nilai integer positif, atau 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

Siapkan untuk membangun gambar kustom dengan Image Builder

Sebelum Anda membuat EC2 gambar dengan Image Builder, verifikasi bahwa Anda telah memenuhi prasyarat berikut untuk membuat pipeline gambar. Kecuali dinyatakan lain secara khusus, prasyarat ini diperlukan untuk semua jenis jaringan pipa.

Prasyarat

- Peran terkait layanan Image Builder
- Persyaratan konfigurasi
- Repositori kontainer untuk pipa gambar kontainer
- Host khusus untuk gambar macOS
- Prasyarat IAM
- Prasyarat Agen Systems Manager

Setelah memenuhi prasyarat, Anda dapat mengelola Image Builder EC2 dari salah satu antarmuka berikut.

- EC2 Konsol Image Builder
- Perintah Image Builder di AWS CLI
- EC2 Referensi API Image Builder
- AWS SDKs dan Alat

Peran terkait layanan Image Builder

EC2 Image Builder menggunakan peran terkait layanan untuk memberikan izin ke AWS layanan 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 IAM untuk Image Builder

Persyaratan konfigurasi

- Image Builder mendukung AWS PrivateLink. Untuk informasi selengkapnya tentang mengonfigurasi titik akhir VPC untuk Image Builder, lihat. Image Builder dan AWS PrivateLink antarmuka titik akhir VPC
- 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 berlaku. 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 penginstalan 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 Systems Manager 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 Instal Systems Manager Agent secara manual pada EC2 instance untuk Windows Server di Panduan AWS Systems Manager Pengguna.

  macOS

  TBD - butuh sesuatu tentang prasyarat

Repositori kontainer untuk 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 Anda 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.

Host khusus untuk gambar macOS

Instans Amazon EC2 Mac memerlukan Host Khusus yang berjalan pada jenis instans logam. Sebelum membuat image macOS kustom, Anda harus Mengalokasikan Host Khusus ke akun Anda. Untuk informasi selengkapnya tentang instance Mac dan daftar jenis instans yang mendukung sistem operasi macOS secara native, lihat instans Amazon EC2 Mac di Panduan Pengguna Amazon. EC2

Ketika Anda telah membuat Host Khusus, Anda dapat mengonfigurasi pengaturan di sumber daya konfigurasi infrastruktur untuk gambar Anda. Konfigurasi infrastruktur mencakup properti penempatan tempat Anda dapat menentukan host, grup penempatan host, atau Availability Zone tempat instance yang diluncurkan dari gambar Anda harus pergi.

Prasyarat 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
- Amazon SSMManaged InstanceCore

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

JSON

```
{
"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. Masuk ke AWS Management Console 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 SSMManaged InstanceCore kebijakan Amazon EC2InstanceProfileForImageBuilderECRContainerBuildsdan Amazon.

> **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 Image Builder dengan AWS RAM.

Prasyarat Agen Systems Manager

EC2 Image Builder menjalankan Agen AWS Systems Manager (Systems Manager) pada EC2 instance 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.

Untuk memilih keluar dari pengumpulan informasi ini, pilih metode yang cocok dengan lingkungan pilihan Anda:

- Konsol Image Builder - Hapus pilihan kotak centang Aktifkan koleksi metadata yang disempurnakan.
- AWS CLI— Tentukan `--no-enhanced-image-metadata-enabled` opsi
- Image Builder API atau SDKs - Atur `enhancedImageMetadataEnabled` parameter ke`false`.

Image Builder menggunakan `RunCommand` untuk mengirim tindakan ke build dan instance pengujian Anda sebagai bagian dari alur kerja build dan pengujian image. Anda tidak dapat memilih keluar dari penggunaan `RunCommand` untuk mengirim tindakan ke instance build dan pengujian Anda.

Pelajari cara membuat gambar kustom dengan tutorial Image Builder

Ada banyak cara untuk membangun gambar dan komponen khusus dengan EC2 Image Builder. Tutorial membantu Anda mempelajari tentang konsep utama Image Builder. Setiap tutorial menyajikan kasus penggunaan dengan langkah-langkah yang dapat Anda ikuti untuk pertama kalinya. Instruksi menggunakan default jika memungkinkan untuk membantu mempelajari keseluruhan proses. Setelah Anda menggunakan salah satu tutorial, Anda dapat menjelajahi lebih banyak cara untuk menyesuaikan gambar Anda sendiri.

Bangun gambar pertama Anda

Tutorial berikut menunjukkan cara membuat gambar pertama Anda dengan wizard konsol Image Builder. Di akhir tutorial Anda akan membuat kumpulan sumber daya Image Builder berikut. Langkah terakhir dalam tutorial ini adalah membersihkan sumber daya yang Anda buat.

- Resep gambar untuk Gambar Mesin Amazon (AMI) Anda atau resep wadah untuk gambar wadah Anda.
- Sumber daya konfigurasi infrastruktur dengan pengaturan default.
- Sumber daya pengaturan distribusi dengan pengaturan default yang mendistribusikan output ke wilayah sumber (akun Anda di Wilayah yang Anda gunakan untuk menjalankan wizard konsol).
- Pipeline image yang menggunakan resource yang terdaftar untuk membuat image keluaran Anda dengan alur kerja build image default.
- AMI keluaran atau gambar kontainer.

Tutorial wizard konsol

- Wisaya pipa: Buat AMI
- Wisaya pipa: Buat gambar wadah

Buat komponen khusus dengan parameter input

Tutorial berikut menunjukkan cara membuat komponen kustom yang mendefinisikan parameter input, dan kemudian mengatur nilai dari resep Image Builder Anda.

Komponen khusus dengan parameter

Gunakan parameter Systems Manager dengan Image Builder

Tutorial berikut menunjukkan cara membuat AWS Systems Manager parameter Parameter Store dan menggunakannya dalam resep gambar.

Gunakan parameter gambar dasar dalam resep Anda

Anda juga dapat menggunakan parameter Parameter Store dalam pengaturan distribusi AMI untuk menyimpan ID gambar keluaran Anda, dan dalam komponen khusus. Untuk informasi selengkapnya, lihat Membuat dan memperbarui konfigurasi distribusi AMI untuk distribusi, dan Gunakan parameter Systems Manager Parameter Store untuk komponen kustom.

Tutorial: Membuat pipeline gambar dengan output AMI dari wizard konsol Image Builder

Tutorial ini memandu Anda melalui pembuatan pipeline otomatis untuk membangun dan memelihara EC2 image Image Builder 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: Bersihkan

Langkah 1: Tentukan detail pipa

1. Buka konsol EC2 Image Builder di 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 Anda 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. Pembuatan versi semantik di Image Builder

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 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 Linux AMIs yang dikelola oleh Amazon. Untuk tutorial ini, pilih `Amazon Linux 2 x86` gambar.

   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. Pembuatan versi semantik di Image Builder

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, pilih Add build components, dan pilih `Amazon managed` dari daftar filter pemilik komponen. Ini membuka panel pilihan di sisi kanan antarmuka konsol tempat Anda dapat menelusuri dan memfilter komponen yang tersedia.

   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. Pertahankan default untuk opsi Pembuatan Versi (Gunakan versi 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. Pembuatan versi semantik di Image Builder

d. Pilih Tambahkan ke resep untuk menambahkan komponen ke resep Anda. Ini menutup panel pemilihan komponen.

e. Kembali ke panel Build components, komponen yang Anda tambahkan akan ditampilkan.

7. 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.

a. Ulangi langkah sebelumnya untuk menambahkan `update-linux-kernel-5` komponen ke resep Anda.

b. Komponen yang baru saja Anda tambahkan memiliki parameter input untuk versi kernel. 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. Untuk informasi selengkapnya tentang penggunaan parameter input dalam komponen Anda, dan menyetelnya di resep, lihatTutorial: Buat komponen kustom dengan parameter input.

c. Pilih salah satu komponen, dan seret ke atas atau ke bawah untuk mengubah urutan komponen akan berjalan.

d. Untuk menghapus `update-linux-kernel-5` komponen, pilih X dari sudut kanan atas kotak komponen.

   Ulangi langkah ini untuk menghapus komponen lain yang mungkin telah Anda tambahkan, hanya menyisakan `update-linux` komponen yang dipilih.

8. Pilih Berikutnya untuk melanjutkan ke langkah berikutnya.

Langkah 3: Tentukan konfigurasi infrastruktur - opsional

Image Builder meluncurkan EC2 instance 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 untuk instance EC2 build dan test yang digunakan untuk mengonfigurasi image Anda. Untuk informasi selengkapnya tentang setelan konfigurasi infrastruktur, lihat CreateInfrastructureConfigurationdi 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 (OUs) 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, lihatMengelola pengaturan distribusi 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 konfigurasikan. 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 pengenal 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: Bersihkan

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 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 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.

Tutorial: Buat pipeline gambar dengan output gambar kontainer Docker dari wizard konsol Image Builder

Tutorial ini memandu Anda melalui pembuatan pipeline otomatis untuk membangun dan memelihara EC2 image 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: Bersihkan

Langkah 1: Tentukan detail pipa

1. Buka konsol EC2 Image Builder di 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. Pembuatan versi semantik di Image Builder

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 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 gambar penampung 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. Pembuatan versi semantik di Image Builder

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. Pembuatan versi semantik di Image Builder

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, lihatTutorial: Buat komponen kustom dengan parameter input.

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 versi 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.

   Secara default, Image Builder menggunakan variabel kontekstual berikut di Dockerfile Anda.

   ParentImage (wajib)

   Pada waktu pembuatan, variabel ini menyelesaikan gambar dasar untuk resep Anda.

   Contoh:

   ```
   FROM
   {{{ imagebuilder:parentImage }}}
   ```

   lingkungan (diperlukan jika komponen ditentukan)

   Variabel ini akan menyelesaikan skrip yang menjalankan komponen.

   Contoh:

   ```
   {{{ imagebuilder:environments }}}
   ```

   komponen (opsional)

   Image Builder menyelesaikan skrip komponen build dan test untuk komponen yang disertakan dalam resep container. Variabel ini dapat ditempatkan di mana saja di Dockerfile, setelah variabel lingkungan.

   Contoh:

```
{{{ imagebuilder:components }}}
```

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 EC2 instance 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 CreateInfrastructureConfigurationdi 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 disebarkan 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 pengenal 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: Bersihkan

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 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`** 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.

Tutorial: Buat komponen kustom dengan parameter input

Anda dapat mengelola komponen Image Builder, termasuk membuat dan menyetel parameter komponen, langsung dari konsol EC2 Image Builder, dari AWS CLI, atau dari Image Builder API atau SDKs. Di bagian ini, kita akan membahas pembuatan dan penggunaan parameter di komponen Anda, dan mengatur parameter komponen melalui konsol Image Builder dan AWS CLI perintah saat runtime.

> **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, Anda harus menyediakan dokumen komponen aplikasi YAMG atau JSON. Dokumen berisi kode yang berjalan selama fase dan langkah-langkah yang Anda tentukan untuk memberikan penyesuaian untuk gambar Anda. 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 dokumen komponen. Contoh dokumentasi menggunakan file bernama *hello-world-test.yaml* yang menyertakan 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 dokumen komponen. Contoh dokumentasi menggunakan file bernama *`hello-world-test.yaml`* yang menyertakan 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.

macOS

Membuat dokumen komponen YAMG

Gunakan alat pengeditan file untuk membuat file dokumen komponen. Contoh dokumentasi menggunakan file bernama *hello-world-test.yaml* yang menyertakan konten berikut:

```
# Document Start
#
name: "HelloWorldTestingDocument-macOS"
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.

Untuk informasi selengkapnya tentang fase, langkah, dan sintaks untuk dokumen komponen AWSTOE aplikasi, 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 membuat komponen langsung dari dokumen YAMAL Anda dengan konsol Image Builder, lihatBuat komponen khusus dari konsol.
- Untuk membuat komponen dari baris perintah dengan create-component perintah Image Builder, lihatBuat komponen kustom dari 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 dari 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, mereka 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 parameter input, nilai default muncul di kotak Nilai dengan teks berwarna abu-abu. Jika tidak ada nilai lain yang dimasukkan, Image Builder menggunakan nilai default.

Gunakan parameter gambar dasar dalam resep Anda

Saat Anda membuat resep untuk penyesuaian gambar, ada beberapa cara untuk mengidentifikasi gambar dasar yang Anda mulai. Jika Anda menentukan ID Amazon Machine Image (AMI) untuk gambar dasar Anda dan gambar dasar itu diperbarui, ID AMI-nya mungkin berubah dan Anda perlu memperbarui resep agar sesuai.

Alih-alih mengubah resep Anda setiap kali ID gambar dasar berubah, Anda dapat menentukan AWS Systems Manager parameter Parameter Store (parameter SSM) untuk menyimpan nilai ID AMI gambar dasar Anda, dan kemudian menggunakan parameter untuk menentukan gambar dasar dalam resep Anda. Untuk AWS dikelola AMIs, Anda dapat menggunakan parameter publik untuk versi terbaru.

Tutorial ini memandu Anda melalui proses pembuatan parameter ID AMI dan menggunakannya dalam resep gambar. Langkah-langkah Image Builder dalam tutorial ini berbasis konsol.

Daftar Isi

- Langkah 1: Temukan atau buat parameter Parameter Store
- Langkah 2: Konfigurasikan izin IAM
- Langkah 3: Buat Resep Gambar yang menggunakan parameter

Langkah 1: Temukan atau buat parameter Parameter Store

Proses untuk langkah ini tergantung pada jenis AMI yang Anda tentukan untuk gambar dasar Anda. Untuk AWS dikelola AMIs, Anda dapat menggunakan parameter publik yang mengacu pada versi saat ini. Beberapa parameter mungkin tidak tersedia di semua Wilayah AWS.

Untuk memulai, buka tab yang sesuai dengan AMI Anda.

AWS managed AMI

Jika image dasar Anda adalah AMI AWS terkelola, Anda dapat menggunakan parameter publik untuk menentukan ID AMI, daripada membuat parameter Anda sendiri. Untuk menemukan parameter publik untuk AMI Anda, lihat Menemukan parameter publik di Panduan AWS Systems Manager Pengguna.

Custom AMI

Untuk membuat parameter ID AMI, ikuti petunjuk untuk Membuat parameter Parameter Store di Systems Manager dengan konsol, AWS CLI, atau PowerShell. Berikan nilai berikut untuk memastikan bahwa nilai parameter adalah ID AMI.

Tingkat parameter: `Standard`

Jenis: `String`

Tipe data: Pilih`aws:ec2:image`. Saat Anda menentukan jenis ini, sistem memvalidasi nilai yang dimasukkan untuk memastikan bahwa itu adalah ID AMI.

Nilai: Masukkan ID AMI yang valid (misalnya,***`ami-1234567890abcdef1`***).

Langkah 2: Konfigurasikan izin IAM

Untuk menggunakan parameter Systems Manager Parameter Store (parameter SSM), baik publik maupun pribadi, Anda harus menentukan tindakan Systems Manager Parameter Store berikut dalam peran eksekusi Image Builder, dengan parameter yang tercantum sebagai sumber daya.

- `ssm:GetParameter`— Tindakan ini memungkinkan Anda untuk menggunakan parameter SSM untuk menentukan gambar dasar dalam resep Anda.
- `ssm:PutParameter`— Tindakan ini memungkinkan Anda untuk menyimpan ID AMI keluaran dalam parameter SSM selama distribusi. Definisi kebijakan terlihat sama, tetapi tutorial ini tidak menyertakan tindakan put dalam kebijakan contoh.

Untuk menggunakan parameter SSM dalam komponen kustom, Anda harus menentukan `ssm:GetParameter` dalam peran profil instance sebagai gantinya. Untuk informasi selengkapnya, lihat Gunakan parameter Systems Manager Parameter Store.

Saat membuat pipeline atau menggunakan perintah create-image di dalamnya AWS CLI, Anda hanya dapat menentukan satu peran eksekusi Image Builder. Jika Anda telah menentukan peran eksekusi

alur kerja Image Builder, Anda akan menambahkan izin parameter ke peran tersebut. Jika tidak, Anda akan membuat peran kustom baru yang mencakup izin yang diperlukan untuk parameter SSM.

1. Buat peran khusus (opsional)

   Jika Anda sudah memiliki peran kustom yang ditentukan untuk izin Image Builder, Anda dapat melewati langkah ini.

   Ikuti proses untuk Membuat peran untuk mendelegasikan izin ke AWS layanan di AWS Identity and Access Management Panduan Pengguna.

2. Tambahkan izin ke peran kustom Anda

   Untuk menambahkan izin parameter SSM ke peran kustom Anda, ikuti kebijakan Perbarui izin untuk proses peran di Panduan Pengguna.AWS Identity and Access Management

   Contoh kebijakan berikut menunjukkan `ssm:GetParameter` tindakan dengan parameter yang dibuat di akun Anda.

   JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": "ssm:GetParameter",
   "Resource": "arn:aws:ssm:*:111122223333:parameter/ImageBuilder-*"
  }
 ]
}
```

Untuk informasi selengkapnya tentang sumber daya parameter publik, lihat Memanggil parameter publik AMI di Panduan AWS Systems Manager Pengguna.

Langkah 3: Buat Resep Gambar yang menggunakan parameter

1. Buka konsol EC2 Image Builder di https://console.aws.amazon.com/imagebuilder/.
2. Pilih Resep gambar, lalu pilih Buat resep gambar dari halaman daftar.

3. Isi bagian Base image, sebagai berikut:

   a. Pilih opsi Gunakan AMI kustom. Ini menampilkan bidang tambahan tempat Anda dapat memasukkan ID AMI atau parameter SSM yang berisi ID AMI.

   b. Pilih opsi parameter SSM.

   c. Di bidang parameter SSM, masukkan nama parameter atau Nama Sumber Daya Amazon (ARN) dari parameter yang Anda buat di Langkah 1. Jika Anda memasukkan nama, itu tidak akan memiliki awalan di konsol.

4. Lengkapi konfigurasi resep yang tersisa sesuai kebutuhan.

> **Note**
>
> Jika Anda mengatur gambar induk melalui antarmuka lain, seperti AWS CLI, nama parameter harus memiliki awalan `ssm:` (misalnya,. `ssm:/ImageBuilder-Tutorial/BaseAMI`

Gunakan komponen untuk menyesuaikan image 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 AWSTOE komponen dan aplikasi manajemen komponen pada instance EC2 build dan pengujian Amazon. 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 AWS Health, lihat Panduan AWS Health Pengguna.

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 EC2 build Amazon yang menjalankan image dasar Anda, untuk membuat baseline untuk gambar baru Anda. Misalnya, resep Anda dapat menyertakan komponen yang menginstal aplikasi atau memodifikasi pengaturan firewall sistem operasi.

   Fase berikut dari dokumen komponen Anda berjalan selama tahap pembuatan:

   - 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 dibuat AMIs dan gambar kontainer. Untuk alur kerja AMI, Image Builder meluncurkan EC2 instance 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 berikut dari dokumen komponen Anda berjalan untuk setiap komponen yang disertakan dalam resep selama tahap pengujian build gambar:

   - 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**
>
> Sementara kerangka AWSTOE aplikasi 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 gambar, dokumen komponen harus mendefinisikan setidaknya satu dari fase ini: `build` atau`validate`. Agar komponen dapat berjalan selama tahap pengujian gambar, 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, lihatTutorial: Buat komponen kustom dengan parameter input.

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, lihatGunakan kerangka dokumen AWSTOE komponen untuk komponen kustom.

> **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 di AWS 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 di AWS CLI.

Topik

- Daftar dan lihat detail komponen
- Gunakan AWS Marketplace komponen untuk menyesuaikan gambar Anda
- Menggunakan komponen terkelola untuk menyesuaikan image Image Builder
- Mengembangkan komponen kustom untuk image Image Builder Anda
- Bagaimana Image Builder menggunakan AWS Task Orchestrator and Executor aplikasi untuk mengelola komponen

Daftar dan lihat detail komponen

Bagian ini menjelaskan bagaimana Anda dapat menemukan informasi dan melihat detail untuk komponen yang Anda gunakan dalam resep EC2 Image Builder.

Detail komponen

- Daftar komponen Image Builder
- Daftar versi pembuatan komponen dari AWS CLI
- Dapatkan detail komponen dari AWS CLI
- Dapatkan detail kebijakan komponen dari AWS CLI

Daftar komponen Image Builder

Anda dapat menggunakan salah satu metode berikut untuk membuat daftar dan memfilter komponen Image Builder.

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/.
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:

   - AWS Marketplace— Komponen yang terkait langsung dengan langganan AWS Marketplace produk.
   - 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 komponen Image Builder 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

- `AWSMarketplace`
- `Amazon`
- `Self`
- `Shared`
- `ThirdParty`

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 pembuatan komponen dari 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. Pembuatan versi semantik di Image Builder

```
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 dari 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 dari 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
```

Gunakan AWS Marketplace komponen untuk menyesuaikan gambar Anda

Selain banyak pilihan gambar yang dibuat oleh Independent Software Vendors (ISVs), AWS Marketplace menawarkan komponen yang dapat Anda gunakan untuk menyesuaikan gambar Image

Builder Anda sendiri. Anda harus berlangganan AWS Marketplace komponen-komponen ini sebelum Anda dapat menggunakannya dalam resep gambar Anda untuk membangun gambar baru.

Saat Anda menentukan AWS Marketplace komponen dalam resep gambar, Image Builder memvalidasi langganan dan melakukan pemeriksaan dependensi untuk memastikan bahwa Anda memiliki sumber daya yang Anda perlukan untuk menggunakannya. Ketika validasi berhasil, Image Builder membuat unduhan aman untuk komponen dan artefaknya untuk digunakan oleh build pipeline gambar.

Temukan AWS Marketplace komponen

Anda dapat menemukan komponen AWS Marketplace perangkat lunak untuk digunakan dalam resep Anda dari halaman Temukan produk di konsol Image Builder, sebagai berikut.

1. Buka konsol EC2 Image Builder di https://console.aws.amazon.com/imagebuilder/.
2. Dari panel navigasi, pilih Temukan produk di AWS Marketplacebagian.
3. Pilih tab Komponen. Tab ini mencantumkan semua AWS Marketplace produk yang menggunakan opsi pengiriman yang menyertakan komponen terkait AWS Marketplace.
4. Untuk mencari produk perangkat lunak tertentu yang menyertakan komponen, Anda dapat memasukkan bagian dari nama di bilah pencarian atau memfilter berdasarkan`Status`,`Operating System`,`Publisher`, atau`Categories`. Bilah pencarian juga berisi kontrol pagination untuk hasil Anda.

Hasil

Setiap AWS Marketplace produk memiliki panel detailnya sendiri yang mencakup informasi berikut.

Nama AWS Marketplace produk dan logo

Nama produk perangkat lunak terkait dengan detail produk di AWS Marketplace. Anda dapat memilih tautan untuk mempelajari lebih lanjut tentang produk di AWS Marketplace. Atau, Anda dapat melihat ringkasan opsi berlangganan dan berlangganan langsung dari hasil pencarian dengan tombol Lihat opsi langganan jika Anda sudah melakukan riset.

Versi

Ini berisi versi utama komponen.

Sistem operasi

Sistem operasi yang dirancang komponen untuk dijalankan.

Penerbit

Penerbit komponen. Ini terkait dengan halaman detail penerbit di AWS Marketplace. Halaman detail penerbit terbuka di tab baru di browser Anda.

Kategori

Satu atau lebih kategori AWS Marketplace produk yang berlaku untuk komponen.

Status

Menunjukkan apakah Anda berlangganan produk ini. Jika Anda tidak berlangganan, Anda dapat memilih Lihat opsi langganan untuk melihat ringkasan opsi langganan AWS Marketplace produk, dan secara opsional berlangganan langsung dari konsol Image Builder.

Komponen terkait

Jika AWS Marketplace produk memiliki satu atau beberapa versi yang disertakan dengan langganan Anda, mereka ditampilkan di bagian Komponen terkait. Bagian ini awalnya diciutkan, dan menampilkan hitungan komponen terkait. Anda dapat memperluas bagian untuk melihat detail lebih lanjut.

> **Note**
>
> Komponen Center for Internet Security (CIS) yang terkait dengan produk AWS Marketplace gambar mereka tidak ditampilkan dalam hasil produk Discover. Jika Anda berlangganan produk gambar mereka, komponen terkait ditampilkan di halaman Langganan, dan sebagai komponen pihak ketiga dalam dialog Buat resep gambar. Untuk informasi selengkapnya tentang komponen, lihatKomponen pengerasan CIS.

Berlangganan AWS Marketplace komponen

Setelah Anda menemukan AWS Marketplace produk dengan komponen yang ingin Anda gunakan dalam resep Anda, Anda dapat berlangganan langsung dari konsol Image Builder, sebagai berikut, atau Anda dapat berlangganan dari AWS Marketplace konsol.

1. Buka konsol EC2 Image Builder di https://console.aws.amazon.com/imagebuilder/.
2. Dari panel navigasi, pilih Temukan produk di AWS Marketplacebagian.
3. Pilih tab Komponen. Tab ini mencantumkan semua AWS Marketplace produk yang menggunakan opsi pengiriman yang menyertakan komponen terkait AWS Marketplace.

4. Untuk mencari AWS Marketplace produk tertentu, masukkan bagian nama di bilah pencarian. Jika Anda tahu penerbit, tetapi bukan nama produk yang tepat atau cara mengejanya, Anda juga dapat memfilter `Publisher` untuk mendapatkan daftar produk yang tersedia penerbit.
5. Pilih produk yang ingin Anda berlangganan dari daftar hasil, dan pilih Lihat opsi langganan. Ini menunjukkan ringkasan opsi berlangganan untuk AWS Marketplace produk.
6. Pilih Berlangganan untuk berlangganan produk tanpa meninggalkan konsol Image Builder. Anda diberi tahu bahwa langganan sedang diproses. Setelah Anda berlangganan, Status diperbarui ke`Subscribed`.

Untuk informasi selengkapnya tentang AWS Marketplace produk yang saat ini Anda berlangganan, lihat langkah-langkah konsol yang dijelaskan diAWS Marketplace langganan di Image Builder.

Menggunakan AWS Marketplace komponen dalam resep gambar Image Builder

Anda dapat menggunakan AWS Marketplace komponen dalam resep gambar Image Builder dengan cara yang sama seperti Anda menggunakan jenis komponen lainnya. Untuk sebagian besar komponen yang terkait dengan produk AWS Marketplace gambar, kategori kepemilikannya adalah`AWS Marketplace`. Misalnya, untuk menggunakan komponen build dari AWS Marketplace produk yang Anda berlangganan, pilih Tambahkan komponen build, lalu pilih `AWS Marketplace` dari daftar. Ini membuka panel pilihan di sisi kanan antarmuka konsol yang mencantumkan AWS Marketplace komponen.

> **Note**
>
> Jika Anda mencari komponen pengerasan CIS, pilih`Third party managed`, dari daftar kepemilikan, bukan. `AWS Marketplace`

Untuk informasi selengkapnya tentang cara memilih, mengatur, dan mengonfigurasi parameter untuk komponen Anda, lihatBuat versi baru dari resep gambar.

Menggunakan komponen terkelola untuk menyesuaikan image Image Builder

Komponen yang dikelola dibuat oleh AWS, kadang-kadang dalam kemitraan dengan organisasi pihak ketiga, seperti Pusat Keamanan Internet (CIS), misalnya. Saat Anda menggunakan komponen terkelola dalam resep gambar atau wadah, Amazon menyediakan versi komponen terbaru yang memiliki tambalan dan pembaruan lain yang diterapkan. Untuk mendapatkan daftar komponen, atau untuk mendapatkan informasi komponen, lihatDaftar dan lihat detail komponen.

Daftar komponen AWS terkelola unggulan berikut mencakup komponen yang tersedia untuk Anda gunakan saat Anda berlangganan CIS yang dikeraskan AMIs melalui. AWS Marketplace

Komponen unggulan

- Penginstalan aplikasi komponen terkelola paket distributor untuk image Image Builder Windows
- Komponen pengerasan CIS
- Amazon mengelola komponen pengerasan STIG untuk Image Builder

Penginstalan aplikasi komponen terkelola paket distributor untuk image Image Builder 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, lihatKonfigurasikan `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, lihatKonfigurasikan `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 dari resep wadah resep kontainer. Untuk informasi selengkapnya tentang *`AwsVssComponents`* paket, lihat Membuat snapshot yang konsisten dengan aplikasi VSS di Panduan Pengguna Amazon. EC2

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, lihatKonfigurasikan 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.

Konfigurasikan izin Distributor Systems Manager

`distributor-package-windows`Komponen dan komponen lain yang menggunakannya, sepertiaws-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.

   JSON

```
{
 "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 sebelah Amazon SSMManaged InstanceCore. (Jenis `SSM` di kotak pencarian jika Anda perlu mempersempit daftar.)
6. Dalam daftar kebijakan ini, pilih kotak di sebelah EC2InstanceProfileForImageBuilder. (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.

Konfigurasikan **`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, lihatTemukan 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.

Konfigurasikan **`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 di AWS Management Console, masuk ke AWS Systems Manager konsol dan pilih Distributor dari panel navigasi. Halaman Distributor menunjukkan semua paket yang tersedia untuk Anda. Untuk informasi selengkapnya tentang daftar paket yang tersedia dengan AWS 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 di AWS 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 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 terkait STIGs yang didukung Image Builder untuk paket itu, komponen pengerasan menerapkannya.

Halaman ini mencantumkan semua STIGs yang didukung Image Builder yang diterapkan ke EC2 instance 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, lihatGunakan komponen untuk menyesuaikan image Image Builder.

Saat Anda membuat gambar, komponen pengerasan STIG mencatat apakah didukung diterapkan atau STIGs 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, lihatMemecahkan 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

AWSTOE Komponen 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-Low versi 2025.2.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 lengkap Windows STIGs, lihat Perpustakaan STIGs Dokumen. Untuk informasi tentang cara melihat daftar lengkap, lihat Alat Melihat STIG.

- Windows Server 2022 STIG Versi 2 Rilis 4

  V-254335, V-254336, V-254337, V-254338, V-254351, V-254357, V-254363, dan V-254481
- Windows Server 2019 STIG Versi 3 Rilis 4

  V-205691, V-205819, V-205858, V-205859, V-205860, V-205870, V-205871, dan V-205923
- Windows Server 2016 STIG Versi 2 Rilis 10

  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-225250, V-225318, V-225319, V-225324, V-225327, V-225330, V-225331, V-225332, V-225333, V-225334, V-225335, V-225336, V-225342, V-225343, V-225355, V-225357, V-225358, V-225359, V-225360, V-225362, V-225363, V-225376, V-225392, V-225394, V-225412, V-225460, V-225462, V-225468, V-225473, V-225473, V-225473 76, V-225479, V-225480, V-225481, V-225482, V-225483, V-225484, V-225485, V-225487, V-225488, V-225489, V-225490, V-225511, V-225514, V-225525, V-225526, V-225526, V-225526, V-225526 25536, dan V-225537
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 6

  Tidak ada pengaturan STIG yang diterapkan ke kerentanan Microsoft .NET Framework untuk Kategori III.
- Windows Firewall STIG Versi 2 Rilis 2

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 5

  V-223016, V-223056, dan V-223078

- Microsoft Edge STIG Versi 2 Rilis 2 (hanya Windows Server 2022)

  V-235727, V-235731, V-235751, V-235752, dan V-235765

STIG-Build-Windows-Medium versi 2025.2.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 lengkap Windows STIGs, lihat Perpustakaan STIGs Dokumen. Untuk informasi tentang cara melihat daftar lengkap, lihat Alat Melihat STIG.

> **Note**
>
> Komponen STIG-Build-Windows-Medium pengerasan mencakup semua pengaturan STIG yang terdaftar yang AWSTOE berlaku untuk komponen STIG-Build-Windows-Low pengerasan, selain pengaturan STIG yang terdaftar khusus untuk kerentanan Kategori II.

- Windows Server 2022 STIG Versi 2 Rilis 4

  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-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-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-254356, V-254358, V-254359, V-254360, V-254361, V-254362, V-254364, V-254365, V-254365 366, V-254367, V-254368, V-254369, V-254370, V-254371, V-254372, V-254373, V-254375, V-254376, V-254377, V-254379, V-254380, V-254383, V-254384, V-254431, V-254432, 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-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-254485, V-254486, V-254486, V-254486 V-254488, V-254489, 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 3 Rilis 4

  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 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-205842, V-205861, V-205863, V-205866, V-205867, V-205868, 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, V-236001, dan V-257503

- Windows Server 2016 STIG Versi 2 Rilis 10

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 72, 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 24895, V-224896, V-224897, V-224898, V-224899, V-224900, V-224901, V-224902, V-224903, V-224904, V-224905, V-224906, V-224907, V-224908, V-224909, 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-225059, V-225061, V-225061, V-225062, V-225063, V-225064, V-225066, V-225067, V-225068, V-225069, V-225069, V-225069, 225072, V-225073, 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 6000, dan V-257502

- 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-225239, V-225259, V-225260, V-225261, V-225263, V-225264, V-225265, V-225266, V-225267, V-225268, V-225269, V-225270, V-225271, V-225272, V-225273, V-225275, V-225276, V-225277, V-225278, V-225279, V-225280, V-225281, V-225282, V-225283, V-225284, V-225285, V-225286, V-225287, V-225288, V-225289, V-225290, V-225296 1, V-225292, V-225293, V-225294, V-225295, V-225296, V-225297, V-225298, V-225299, V-225300, V-225301, V-225302, V-225303, V-225305, V-225314, V-225314, V-225314, V-225314 15, V-225316, V-225317, V-225325, V-225326, V-225329, V-225337, V-225338, V-225339, V-225340, V-225341, V-225344, V-225345, V-225346, V-225347, V-225348, V-225349, V-225350, V-225351, V-225352, V-225353, V-225356, V-225367, V-225368, V-225369, V-225370, V-225371, V-225372, V-225373, V-225374, V-225375, V-225377, V-225378, V-225379, V-225380, V-225381, V-225382, V-225383, V-225384, V-225385, V-225386, V-225386, V-225386 389, V-225391, V-225393, V-225395, V-225397, V-225398,

V-225400, V-225401, V-225402, V-225404, V-225405, V-225406, V-225407, V-225408, V-225409, V-225410, V-225410, V-225410 25411, V-225413, V-225414, V-225415, V-225441, V-225442, V-225443, V-225448, V-225452, V-225453, V-225454, V-225455, V-225456, V-225457, V-225458, V-225461, V-225463, V-225464, V-225469, V-225470, V-225471, V-225472, V-225474, V-225475, V-225477, V-225477, V-225477, V-225477, V-225477, V-225477 78, V-225486, V-225494, V-225500, V-225501, V-225502, V-225503, V-225504, V-225506, V-225508, V-225509, V-225510, V-225513, V-225515, V-225516, V-225517, V-225517, V-225517 518, V-225519, V-225520, V-225521, V-225522, V-225523, V-225524, V-225527, V-225528, V-225529, V-225530, V-225531, V-225532, V-225533, V-225534, V-225535, V-225535, V-225535, V-225535, V-225535, V-225535, V-225538, V-225539, V-225540, V-225541, V-225542, V-225543, V-225544, V-225545, V-225546, V-225548, V-225549, V-225550, V-225551, V-225553, V-225554, V-225555, V-225557, V-225558, V-225559, V-225560, V-225561, V-225562, V-225563, V-225564, V-225565, V-225566, V-225567, V-225568, V-225569, V-225570, V-225571, V-225572, V-225573, dan V-225574

- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 6

  Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah), ditambah V-225238

- Windows Firewall STIG Versi 2 Rilis 2

  Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah), ditambah:

  V-241989, V-241990, V-241991, V-241993, V-241993, V-241998, dan V-242003

- Internet Explorer 11 STIG Versi 2 Rilis 5

  Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah), ditambah:

  V-223015, V-223017, V-223018, V-223019, V-223020, V-223021, V-223022, V-223023, V-223024, V-223025, V-223026, V-223027, V-223028, V-223029, V-223030, V-223031, V-223032, V-223033, V-223034, V-223035, V-223036, V-223037, V-223038, V-223039, V-223040, V-223041, V-223042, V-223043, V-223044, V-223045, V-223046, V-223048, V-223049, V-223050, V-223051, V-223052, V-223053, V-223053, V-223053 54, V-223055, V-223057, V-223058, V-223059, V-223060, V-223061, V-223062, V-223063, V-223064, V-223065, V-223066, V-223067, V-223068, V-223069, V-223070, V-223071, V-223072, V-223073, V-223073, V-223073, V-223073, V-223073 3074, V-223075, V-223076, V-223077, V-223079, V-223080, V-223081, V-223082, V-223083, V-223084, V-223085, V-223086, V-223087, V-223088, V-223089, V-223090, V-223091, V-223092, V-223093, V-223093, V-223094, V-223095, V-223096, V-223097, V-223098, V-223099, V-223100, V-223101,

V-223102, V-223103, V-223104, V-223105, V-223106, V-223107, V-223108, V-223109, V-223110, V-223111, V-223112, V-223113, V-223114, V-223115, V-223116, V-223117, V-223118, V-223119, V-223120, V-223121, V-223122, V-223124, V-223125, V-223126, V-223127, V-223128, V-223128, V-223128, V-223128 23129, V-223130, V-223131, V-223132, V-223133, V-223134, V-223135, V-223136, V-223137, V-223138, V-223139, V-223140, V-223141, V-223142, V-223143, V-223144, V-223145, V-223146, V-223147, V-223148, V-223148 223149, V-250540, dan V-250541

- Microsoft Edge STIG Versi 2 Rilis 2 (hanya Windows Server 2022)

  Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan 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, V-235768 69, V-235770, V-235771, V-235772, V-235773, V-235774, dan V-246736

- Microsoft Defender STIG Versi 2 Rilis 4

  Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan 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-High versi 2025.2.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 lengkap Windows STIGs, lihat Perpustakaan STIGs Dokumen. Untuk informasi tentang cara melihat daftar lengkap, lihat Alat Melihat STIG.

> **Note**
>
> Komponen STIG-Build-Windows-High pengerasan mencakup semua pengaturan STIG yang terdaftar yang AWSTOE berlaku untuk STIG-Build-Windows-Low dan komponen STIG-Build-Windows-Medium pengerasan, selain pengaturan STIG yang terdaftar khusus untuk kerentanan Kategori I.

- Windows Server 2022 STIG Versi 2 Rilis 4

  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 3 Rilis 4

  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 10

  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-225274, V-225354, V-225364, V-225365, V-225366, V-225390, V-225396, V-225399, V-225444, V-225449, V-225491, V-225492, V-225493, V-225496, V-225497, V-225498, V-225505, V-225507, V-225547, V-225552, dan V-225556

- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 6

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan 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 2

  Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori II dan III (Sedang dan Rendah), ditambah:

  V-241992, V-241997, dan V-242002

- Internet Explorer 11 STIG Versi 2 Rilis 5

  V-252910

- Microsoft Edge STIG Versi 2 Rilis 2 (hanya Windows Server 2022)

  Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori II dan III (Sedang dan Rendah), ditambah:

  V-235758 dan V-235759

- Microsoft Defender STIG Versi 2 Rilis 4

  Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan 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 pengerasan Windows untuk pembaruan STIG triwulanan. Untuk melihat perubahan dan versi yang diterbitkan selama seperempat, pilih judul untuk memperluas informasi.

Perubahan Q2 2025 - 26/06/2025:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis Q2 2025 sebagai berikut:

STIG-Build-Windows-Low versi 2025.2.x

- Windows Server 2022 STIG Versi 2 Rilis 4

- Windows Server 2019 STIG Versi 3 Rilis 4
- Windows Server 2016 STIG Versi 2 Rilis 10
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 5
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 6
- Windows Firewall STIG Versi 2 Rilis 2
- Internet Explorer 11 STIG Versi 2 Rilis 5
- Microsoft Edge STIG Versi 2 Rilis 2 (hanya Windows Server 2022)

STIG-Build-Windows-Medium versi 2025.2.x

- Windows Server 2022 STIG Versi 2 Rilis 4
- Windows Server 2019 STIG Versi 3 Rilis 4
- Windows Server 2016 STIG Versi 2 Rilis 10
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 5
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 6
- Windows Firewall STIG Versi 2 Rilis 2
- Internet Explorer 11 STIG Versi 2 Rilis 5
- Microsoft Edge STIG Versi 2 Rilis 2 (hanya Windows Server 2022)
- Defender STIG Versi 2 Rilis 4

STIG-Build-Windows-High versi 2025.2.x

- Windows Server 2022 STIG Versi 2 Rilis 4
- Windows Server 2019 STIG Versi 3 Rilis 4
- Windows Server 2016 STIG Versi 2 Rilis 10
- Windows Server 2012 R2 MS STIG Versi 3 Rilis 5
- Microsoft .NET Framework 4.0 STIG Versi 2 Rilis 6
- Windows Firewall STIG Versi 2 Rilis 2
- Internet Explorer 11 STIG Versi 2 Rilis 5
- Microsoft Edge STIG Versi 2 Rilis 2 (hanya Windows Server 2022)
- Defender STIG Versi 2 Rilis 4

Perubahan Q1 2025 - 05/04/2025:

STIGS yang diperbarui untuk Internet Explorer 11 STIG Versi 2 Rilis 5 untuk semua komponen STIG untuk rilis kuartal pertama 2025.

- STIG-Build-Windows-Low versi 2025.1.x
- STIG-Build-Windows-Medium versi 2025.1.x
- STIG-Build-Windows-High versi 2025.1.x

Perubahan Q4 2024 - 02/04/2025:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis Q4 2024 sebagai berikut:

STIG-Build-Windows-Low versi 2024.4.0

- Windows Server 2022 STIG Versi 2 Rilis 2
- Windows Server 2019 STIG Versi 3 Rilis 2
- Windows Server 2016 STIG Versi 2 Rilis 9
- 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 2
- Internet Explorer 11 STIG Versi 2 Rilis 5
- Microsoft Edge STIG Versi 2 Rilis 2 (hanya Windows Server 2022)

STIG-Build-Windows-Medium versi 2024.4.0

- Windows Server 2022 STIG Versi 2 Rilis 2
- Windows Server 2019 STIG Versi 3 Rilis 2
- Windows Server 2016 STIG Versi 2 Rilis 9
- 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 2
- Internet Explorer 11 STIG Versi 2 Rilis 5
- Microsoft Edge STIG Versi 2 Rilis 2 (hanya Windows Server 2022)
- Defender STIG Versi 2 Rilis 4

STIG-Build-Windows-High versi 2024.4.0

- Windows Server 2022 STIG Versi 2 Rilis 2
- Windows Server 2019 STIG Versi 3 Rilis 2
- Windows Server 2016 STIG Versi 2 Rilis 9
- 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 2
- Internet Explorer 11 STIG Versi 2 Rilis 5
- Microsoft Edge STIG Versi 2 Rilis 2 (hanya Windows Server 2022)
- Defender STIG Versi 2 Rilis 4

Perubahan Q3 2024 - 10/04/2023 (tidak ada perubahan):

Tidak ada perubahan untuk komponen Windows STIGS untuk rilis kuartal ketiga 2024.

Perubahan Q2 2024 - 05/10/2024 (tidak ada perubahan):

Tidak ada perubahan untuk komponen Windows STIGS untuk rilis kuartal kedua 2024.

Perubahan Q1 2024 - 02/06/2024 (tidak ada perubahan):

Tidak ada perubahan untuk komponen Windows STIGS untuk rilis kuartal pertama 2024.

Perubahan Q4 2023 - 12/04/2023 (tidak ada perubahan):

Tidak ada perubahan untuk komponen Windows STIGS untuk rilis kuartal keempat 2023.

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-Low versi 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-Medium versi 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-High versi 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-High 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

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-High 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

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-High 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

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 Linux memiliki parameter input opsional yang membantu Anda menyesuaikan perilaku berikut untuk instance Linux Anda.

- InstallPackages (string) Jika nilainya`No`, komponen tidak menginstal paket perangkat lunak tambahan apa pun. Jika nilainya`Yes`, komponen menginstal paket perangkat lunak tambahan yang diperlukan untuk kepatuhan maksimum. Nilai default-nya `No`.
- SetDoDConsentBanner (string) Jika nilainya`No`, spanduk persetujuan DoD tidak ditampilkan saat Anda melampirkan ke instance yang memiliki salah satu komponen STIG Linux diinstal. Jika nilainya`Yes`, spanduk persetujuan DoD ditampilkan sebelum Anda masuk saat Anda melampirkan ke instance yang memiliki salah satu komponen STIG Linux yang diinstal. Anda harus mengakui spanduk sebelum Anda dapat masuk. Nilai default-nya `No`.

  Untuk contoh spanduk persetujuan, lihat Pemberitahuan Privasi dan Persetujuan Departemen Pertahanan Penafian yang muncul saat Anda mengakses situs web Layanan Dokumen DLA.

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)

Pengaturan RHEL 9 STIG

- RHEL 9
- CentOS Aliran 9

STIG-Build-Linux-Low versi 2025.2.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 lengkap, lihat Perpustakaan STIGs Dokumen. Untuk informasi tentang cara melihat daftar lengkap, lihat Alat Melihat STIG.

RHEL 7 STIG Versi 3 Rilis 15

- RHEL 7/CentOS 7/ AL2

  V-204452, V-204576, dan V-204605

RHEL 8 STIG Versi 2 Rilis 3

- RHEL 8/CentOS 8/AL 2023

  V-230241, 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-230495, V-230494 6, V-230497, V-230498, V-230499, V-244527, dan V-274141

RHEL 9 STIG Versi 2 Rilis 4

- RHEL 9/CentOS Aliran 9

  V-257782, V-257795, V-257796, V-257824, V-257880, V-257946, V-257947, V-258037, V-258067, V-258069, V-258076, V-258138, dan V-258173

Ubuntu 18.04 STIG Versi 2 Rilis 15

V-219163, V-219164, V-219165, V-219172, V-219173, V-219174, V-219175, V-219178, V-219179, V-219180, V-219210, V-219301, V-219322, V-219327, V-219332, dan V-219333

Ubuntu 20.04 STIG Versi 2 Rilis 2

V-238202, V-238203, 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

Ubuntu 22.04 STIG Versi 2 Rilis 4

V-260472, V-260476, V-260479, V-260480, V-260481, V-260520, V-260521, V-260549, V-260550, V-260551, V-260552, V-260581, dan V-260596

Ubuntu 24.04 STIG Versi 1 Rilis 1

V-270645, V-270646, V-270664, V-270677, V-270690, V-270695, V-270706, V-270710, V-270749, V-270752, V-270818, dan V-270820

STIG-Build-Linux-Medium versi 2025.2.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 lengkap, lihat Perpustakaan STIGs Dokumen. Untuk informasi tentang cara melihat daftar lengkap, lihat Alat Melihat STIG.

> **Note**
>
> Komponen STIG-Build-Linux-Medium pengerasan mencakup semua pengaturan STIG yang terdaftar yang AWSTOE berlaku untuk komponen STIG-Build-Linux-Low pengerasan, selain pengaturan STIG yang terdaftar khusus untuk kerentanan Kategori II.

RHEL 7 STIG Versi 3 Rilis 15

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah) untuk distribusi Linux ini, ditambah:

- RHEL 7/CentOS 7/ 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-204431, V-204434, V-204435, V-204437, V-204449, V-204450, V-204451, V-204457, V-204466, V-204490, V-204491, V-204503, V-204507, V-204508, V-204510, V-204511, V-204512, V-204512, V-204512, V-204512, V-204512 4514, V-204515, 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-204547, V-204547, 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-204568, V-204572, V-204579, V-204584, V-204585, V-204587, V-204588, V-204589, V-204590, V-204591, V-204592, V-204593, V-204596, V-204597, 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-204616, V-204617, V-204619, V-204622, V-204625, V-204630, V-204631, V-204633, V-233307, V-237634, V-237635, V-251703, V-255925, V-255925, V-255925, V-255925 27, V-255928, dan V-256970

RHEL 8 STIG Versi 2 Rilis 3

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-230238, V-230239, V-230240, V-230244, V-230245, V-230246, V-230247, V-230248, V-230249, V-230250, V-230255, V-230257, V-230258, V-230259, V-230261, V-230262, V-230266, V-230267, V-230268, V-230273, V-230275, V-230277, V-230278, V-230279, V-230280, V-230282, V-230286, V-230287, V-230288, V-230290, V-230291, V-230296, V-230298, V-230310, V-230311, V-230312, V-230313, V-230314, V-230315, V-230316, V-230324, V-230325, V-230325, V-230325 326, V-230327, V-230330, V-230332, V-230333, V-230335, V-230337, V-230339, V-230341, V-230343, V-230345, V-230348, V-230353, V-230354, V-230356, V-230357, V-230358, V-230359, V-230360, V-230361, V-230362, V-230363, V-230365, V-230366, V-230368, V-230369, V-230370, V-230373, V-230375, V-230376, V-230377, V-230378, V-230382, V-230383, V-230386, V-230387, V-230390, V-230392, V-230393, V-230394, V-230396, V-230396, V-230396, V-230396 7, V-230398, V-230399, V-230400, V-230401, V-230402, V-230403, V-230404, V-230405, V-230406, V-230407, V-230408, V-230409, V-230410, V-230411, V-230412, V-230412, V-230412, V-230412, V-230412 413, V-230418, V-230419, V-230421, V-230422, V-230423, V-230424, V-230425, V-230426, V-230427, 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-230447, V-230447, V-230447, V-230447, V-230447 48, V-230449, V-230455, V-230456, V-230462, V-230463, V-230464, V-230465, V-230466, V-230467, V-230471, V-230472, V-230473, V-230474, V-230478, V-230480, V-230480, V-230480 30482, V-230483, V-230488, V-230489, V-230502, V-230503, V-230505, V-230507, V-230523, V-230525, V-230526, V-230527, V-230532, V-230535, V-230536, V-230537, V-230538, V-230539, V-230540, V-230541, V-230542, V-230543, V-230544, V-230545, V-230546, V-230547, V-230548, V-230549, V-230550, V-230555, V-230556, V-230557, V-230559, V-230560, V-230561, V-237640, V-237642, V-237643, V-244523, V-244524, V-244525, V-244526, V-244528, V-244528, V-244533, V-244538, V-244542, V-244543, V-244544, V-244547, V-244550, V-244551, V-244552, V-244553, V-244554, V-250315, V-250316, V-250317, V-251710, V-251711, V-251713, V-251714, V-251715, V-251716, V-251717, V-251718, V-256974, dan V-257258

RHEL 9 STIG Versi 2 Rilis 4

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah) untuk distribusi Linux ini, ditambah:

- RHEL 9/CentOS Aliran 9

V-257780, V-257781, V-257783, V-257786, V-257788, V-257790, V-257791, V-257792, V-257793, V-257794, V-257797, V-257798, V-257799, V-257801, V-257802, V-257803, V-257804, V-257804, V-257805, V-257806, V-257807, V-257808, V-257809, V-257810, V-257811, V-257812, V-257813, V-257814, V-257815, V-257816, V-257817, V-257825, V-257827, V-257828, V-257829, V-257830, V-257831, V-257832, V-257833, V-257834, V-257836, V-257838, V-257839, V-257840, V-257841, V-257842, V-257849, V-257882, V-257883, V-257884, V-257885, V-257886, V-257886, V-257886 887, V-257888, V-257889, V-257890, V-257891, V-257892, V-257893, V-257894, V-257895, V-257896, V-257897, V-257898, V-257899, V-257900, V-257901, V-257902, V-257903, V-257904, V-257905, V-257906, V-257907, V-257907, V-257907, V-257907, V-257907 7908, V-257909, V-257910, V-257911, V-257912, V-257913, V-257914, V-257915, V-257916, V-257917, V-257918, V-257919, V-257920, V-257921, V-257922, V-257923, V-257924, V-257925, V-257926, V-257927, V-257928, V-257929, V-257930, V-257933, V-257934, V-257935, V-257936, V-257939, V-257940, V-257942, V-257943, V-257944, V-257948, V-257951, V-257952, V-257953, V-257953, V-257953, V-257953 54, V-257957, V-257958, V-257959, V-257960, V-257961, V-257962, V-257963, V-257964, V-257965, V-257966, V-257967, V-257968, V-257969, V-257970, V-257971, V-257972, V-257974, V-257975, V-257976, V-257977, V-257978, V-257979, V-257980, V-257982, V-257983, V-257985, V-257987, V-257988, V-257992, V-257993, V-257994, V-257995, V-257996, V-257998,

V-257999, V-258000, V-258000, V-258000, V-258000 8001, V-258002, V-258003, V-258004, V-258005, V-258006, V-258007, V-258008, V-258009, V-258010, V-258011, V-258028, V-258034, V-258035, V-258038, V-258039, V-258040, V-258041, V-258043, V-258043, V-258043, V-258043, V-258043, V-258043, V-258046, V-258049, V-258052, V-258054, V-258055, V-258056, V-258057, V-258060, V-258063, V-258064, V-258065, V-258066, V-258068, V-258070, V-258071, V-258072, V-258073, V-258074, V-258075, V-258077, V-258079, V-258080, V-258081, V-258082, V-258083, V-258084, V-258085, V-258088, V-258089, V-258091, V-258092, V-258093, V-258095, V-258097, V-258098, V-258099, V-258100, V-258101, V-258102, V-258102, V-258102 103, V-258104, V-258105, V-258107, V-258108, V-258109, V-258110, V-258111, V-258112, V-258113, V-258114, V-258115, V-258116, V-258117, V-258118, V-258119, V-258120, V-258120, V-258120, V-258120, V-258120, V-258120 8122, V-258123, V-258124, V-258125, V-258126, V-258128, V-258129, V-258130, V-258133, V-258137, V-258140, V-258141, V-258142, V-258144, V-258145, V-258146, V-258147, V-258148, V-258150, V-258152 1, V-258152, V-258153, V-258154, V-258156, V-258157, V-258158, V-258159, V-258160, V-258161, V-258162, V-258163, V-258164, V-258165, V-258166, V-258167, V-258167, V-258167 168, V-258169, V-258170, V-258171, V-258172, V-258175, V-258176, V-258177, V-258178, V-258179, V-258180, V-258181, V-258182, V-258183, V-258184, V-258185, V-258185, V-258185 6, V-258187, V-258188, V-258189, V-258190, V-258191, V-258192, V-258193, V-258194, V-258195, V-258196, V-258197, V-258198, V-258199, V-258200, V-258201, V-258202, V-258203, V-258204, V-258206, V-258207, V-258208, V-258209, V-258210, V-258211, V-258212, V-258213, V-258214, V-258215, V-258216, V-258217, V-258218, V-258219, V-258220, V-258220, V-258221, V-258222, V-258223, V-258225, V-258226, V-258227, V-258228, V-258229, V-258232, V-258233, V-258234, V-258237, V-258239, V-258240, dan V-272488

Ubuntu 18.04 STIG Versi 2 Rilis 15

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-219168, V-219176, V-219181, V-219184, V-219186, 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-219225, V-219226, V-219227, V-219227, V-219227, V-219227 28, 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-219297, V-219298, V-219299, V-219300, V-219303, V-219304, V-219306, V-219309, V-219310, V-219311, V-219315, V-219318, V-219319, V-219323, V-219326, V-219328, V-219330, V-219331, V-219335, V-219335, V-219335 36, V-219337, V-219338, V-219339, V-219342, V-219344, V-233779, V-233780, dan V-255906

Ubuntu 20.04 STIG Versi 2 Rilis 2

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-238210, V-238211, V-238212, V-238213, V-238220, V-238225, V-238227, V-238228, V-238229, V-238230, V-238231, V-238232, V-238236, V-238238, V-238239, V-238240, 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-238288, V-238289, V-238290, V-238291, V-238292, V-238293, V-238294, V-238295, V-238297, V-238298, V-238300, V-238301, V-238302, V-238303, V-238303, V-238309, V-238310, V-238315, V-238316, V-238317, V-238318, V-238318, V-238318 238319, V-238320, V-238324, V-238325, V-238329, V-238333, V-238334, V-238337, V-238338, V-238339, V-238340, V-238341, 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-238355, V-238356, V-238359, V-238369, V-238370, V-238371, V-238376, V-238377, V-238378, V-251505, dan V-255912

Ubuntu 22.04 STIG Versi 2 Rilis 4

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah) untuk distribusi Linux ini, ditambah:

V-260471, V-260473, V-260474, V-260475, V-260477, V-260478, V-260485, V-260486, V-260487, V-260488, V-260489, V-260490, V-260491, V-260492, V-260493, V-260493, V-260493, V-260493, V-260493, V-260493, V-260493 4, V-260495, V-260496, V-260497, V-260498, V-260499, V-260500, V-260505, V-260506, V-260507, V-260508, V-260509, V-260510, V-260511, V-260512, V-260513, V-260513, V-260513, V-260513 514, V-260522, V-260527, V-260528, V-260530, V-260533, V-260534, V-260535, V-260537, V-260538, V-260540, V-260542, V-260543, V-260545, V-260546, V-260547, V-260547, V-260547, V-260547, V-260553, V-260554, V-260555, V-260556, V-260557, V-260560, V-260561, V-260562, V-260563, V-260564, V-260565, V-260566, V-260567, V-260569, V-260572, V-260573, V-260574, V-260575, V-260576, V-260582, V-260584, V-260585, V-260586,

V-260588, V-260589, V-260589, V-260589, V-260589, V-260589 90, V-260591, V-260594, V-260597, V-260598, V-260599, V-260600, V-260601, V-260602, V-260603, V-260604, V-260605, V-260606, V-260607, V-260608, V-260609, V-260609, V-260609, V-260609 610, V-260611, V-260612, V-260613, V-260614, V-260615, V-260616, V-260617, V-260618, V-260619, V-260620, V-260621, V-260622, V-260623, V-260624, V-260625, V-260625, V-260625, V-260625, V-260626, V-260627, V-260628, V-260629, V-260630, V-260631, V-260632, V-260633, V-260634, V-260635, V-260636, V-260637, V-260638, V-260639, V-260640, V-260641, V-260642, V-260643, V-260644, V-260645, V-260646, V-260647, V-260648, dan V-260649

Ubuntu 24.04 STIG Versi 1 Rilis 1

Termasuk semua pengaturan STIG yang didukung yang diterapkan komponen pengerasan untuk kerentanan Kategori III (Rendah) untuk distribusi Linux ini, ditambah:

V-270649, V-270651, V-270652, V-270653, V-270654, V-270656, V-270657, V-270659, V-270660, V-270661, V-270662, V-270663, V-270669, V-270672, V-270673, V-270674, V-270676, V-270678, V-270679, V-270680, V-270681, V-270683, V-270684, V-270685, V-270686, V-270687, V-270688, V-270689, V-270692, V-270693, V-270696, V-270697, V-270698, V-270700, V-270701, V-270702, V-270702, V-270702, V-270702, V-270702, V-270703, V-270704, V-270705, V-270709, V-270715, V-270716, V-270718, V-270720, V-270721, V-270722, V-270723, V-270724, V-270725, V-270726, V-270727, V-270728, V-270729, V-270730, V-270730, V-270730, V-270730, V-270730, V-270731, V-270732, V-270733, V-270737, V-270739, V-270740, V-270741, V-270742, V-270743, V-270746, V-270750, V-270753, V-270755, V-270756, V-270757, V-270758, V-270759, V-270760, V-270765, V-270766, V-270767, V-270768, V-270769, V-270770, V-270771, V-270772, V-270773, V-270775, V-270776, V-270777, V-270778, V-270779, V-270780, V-270781, V-270782, V-270783, V-270784, V-270785, V-270786, V-270786, V-270786 87, V-270788, V-270789, V-270790, V-270791, V-270792, V-270793, V-270794, V-270795, V-270796, V-270797, V-270798, V-270799, V-270800, V-270801, V-270802, V-270803, V-270804, V-270804, V-270804, V-270804, V-270804 805, V-270806, V-270807, V-270808, V-270809, V-270810, V-270811, V-270813, V-270814, V-270815, V-270821, V-270822, V-270823, V-270824, V-270825, V-270826, V-270827, V-270828, V-270828, V-270829, V-270829, 270830, V-270831, dan V-270832

STIG-Build-Linux-High versi 2025.2.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 lengkap, lihat Perpustakaan STIGs Dokumen. Untuk informasi tentang cara melihat daftar lengkap, lihat Alat Melihat STIG.

> **Note**
>
> Komponen STIG-Build-Linux-High pengerasan mencakup semua pengaturan STIG yang terdaftar yang AWSTOE berlaku untuk STIG-Build-Linux-Low dan komponen STIG-Build-Linux-Medium pengerasan, selain pengaturan STIG yang terdaftar yang berlaku khusus untuk kerentanan Kategori I.

RHEL 7 STIG Versi 3 Rilis 15

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/ AL2

  V-204424, V-204425, V-204442, V-204443, V-204447, V-204448, V-204455, V-204497, V-204502, V-204594, V-204620, dan V-204621

RHEL 8 STIG Versi 2 Rilis 3

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-230223, V-230226, V-230264, V-230265, V-230487, V-230492, V-230529, V-230530, V-230531, V-230533, V-230558, dan V-244540

RHEL 9 STIG Versi 2 Rilis 4

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 9/CentOS Aliran 9

V-257784, V-257785, V-257820, V-257821, V-257826, V-257835, V-257955, V-257956, V-257984, V-257986, V-258059, V-258078, V-258094, V-258230, V-258235, dan V-258238

Ubuntu 18.04 STIG Versi 2 Rilis 15

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-219314, V-219316, V-251507, dan V-264388

Ubuntu 20.04 STIG Versi 2 Rilis 2

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-238218, V-238219, V-238326, V-238327, V-238380, dan V-251504

Ubuntu 22.04 STIG Versi 2 Rilis 4

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-260469, V-260482, V-260483, V-260523, V-260524, V-260526, V-260529, V-260539, V-260570, V-260571, dan V-260579

Ubuntu 24.04 STIG Versi 1 Rilis 1

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-270647, V-270648, V-270665, V-270666, V-270708, V-270711, V-270712, V-270713, V-270714, dan V-270717

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 Q2 2025 - 26/06/2025:

Memperbarui versi STIG berikut, menerapkan STIGS untuk rilis kuartal kedua 2025:

STIG-Build-Linux-Low versi 2025.2.x

- RHEL 7 STIG Versi 3 Rilis 15
- RHEL 8 STIG Versi 2 Rilis 3
- RHEL 9 STIG Versi 2 Rilis 4
- Ubuntu 18.04 STIG Versi 2 Rilis 15
- Ubuntu 20.04 STIG Versi 2 Rilis 2
- Ubuntu 22.04 STIG Versi 2 Rilis 4
- Ubuntu 24.04 STIG Versi 1 Rilis 1

STIG-Build-Linux-Medium versi 2025.2.x

- RHEL 7 STIG Versi 3 Rilis 15
- RHEL 8 STIG Versi 2 Rilis 3
- RHEL 9 STIG Versi 2 Rilis 4
- Ubuntu 18.04 STIG Versi 2 Rilis 15
- Ubuntu 20.04 STIG Versi 2 Rilis 2
- Ubuntu 22.04 STIG Versi 2 Rilis 4
- Ubuntu 24.04 STIG Versi 1 Rilis 1

STIG-Build-Linux-High versi 2025.2.x

- RHEL 7 STIG Versi 3 Rilis 15
- RHEL 8 STIG Versi 2 Rilis 3
- RHEL 9 STIG Versi 2 Rilis 4
- Ubuntu 18.04 STIG Versi 2 Rilis 15
- Ubuntu 20.04 STIG Versi 2 Rilis 2
- Ubuntu 22.04 STIG Versi 2 Rilis 4
- Ubuntu 24.04 STIG Versi 1 Rilis 1

Perubahan Q1 2025 - 04/11/2025:

Memperbarui versi STIG berikut, menerapkan STIGS untuk rilis kuartal pertama 2025, dan menambahkan dukungan untuk Ubuntu 24.04:

STIG-Build-Linux-Low versi 2025.1.x

- RHEL 7 STIG Versi 3 Rilis 15
- RHEL 8 STIG Versi 2 Rilis 2
- RHEL 9 STIG Versi 2 Rilis 3
- Ubuntu 18.04 STIG Versi 2 Rilis 15
- Ubuntu 20.04 STIG Versi 2 Rilis 2
- Ubuntu 22.04 STIG Versi 2 Rilis 3
- Ubuntu 24.04 STIG Versi 1 Rilis 1

STIG-Build-Linux-Medium versi 2025.1.x

- RHEL 7 STIG Versi 3 Rilis 15
- RHEL 8 STIG Versi 2 Rilis 2
- RHEL 9 STIG Versi 2 Rilis 3
- Ubuntu 18.04 STIG Versi 2 Rilis 15
- Ubuntu 20.04 STIG Versi 2 Rilis 2
- Ubuntu 22.04 STIG Versi 2 Rilis 3
- Ubuntu 24.04 STIG Versi 1 Rilis 1

STIG-Build-Linux-High versi 2025.1.x

- RHEL 7 STIG Versi 3 Rilis 15
- RHEL 8 STIG Versi 2 Rilis 2
- RHEL 9 STIG Versi 2 Rilis 3
- Ubuntu 18.04 STIG Versi 2 Rilis 15
- Ubuntu 20.04 STIG Versi 2 Rilis 2
- Ubuntu 22.04 STIG Versi 2 Rilis 3

- Ubuntu 24.04 STIG Versi 1 Rilis 1

Perubahan Q4 2024 - 12/10/2024:

Memperbarui versi STIG berikut, menerapkan STIGS untuk rilis kuartal keempat 2024, dan menambahkan informasi tentang dua parameter input baru untuk komponen Linux:

STIG-Build-Linux-Low versi 2024.4.x

- RHEL 7 STIG Versi 3 Rilis 15
- RHEL 8 STIG Versi 2 Rilis 1
- RHEL 9 STIG Versi 2 Rilis 2
- Ubuntu 18.04 STIG Versi 2 Rilis 15
- Ubuntu 20.04 STIG Versi 2 Rilis 1
- Ubuntu 22.04 STIG Versi 2 Rilis 2

STIG-Build-Linux-Medium versi 2024.4.x

- RHEL 7 STIG Versi 3 Rilis 15
- RHEL 8 STIG Versi 2 Rilis 1
- RHEL 9 STIG Versi 2 Rilis 2
- Ubuntu 18.04 STIG Versi 2 Rilis 15
- Ubuntu 20.04 STIG Versi 2 Rilis 1
- Ubuntu 22.04 STIG Versi 2 Rilis 2

STIG-Build-Linux-High versi 2024.4.x

- RHEL 7 STIG Versi 3 Rilis 15
- RHEL 8 STIG Versi 2 Rilis 1
- RHEL 9 STIG Versi 2 Rilis 2
- Ubuntu 18.04 STIG Versi 2 Rilis 15
- Ubuntu 20.04 STIG Versi 2 Rilis 1
- Ubuntu 22.04 STIG Versi 2 Rilis 2

Perubahan Q3 2024 - 10/04/2024 (tidak ada perubahan):

Tidak ada perubahan untuk komponen Linux STIGS untuk rilis kuartal ketiga 2024.

Perubahan Q2 2024 - 05/10/2024:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal kedua 2024. Juga menambahkan dukungan untuk RHEL 9, CentOS Stream 9, dan Ubuntu 22.04, sebagai berikut:

STIG-Build-Linux-Low versi 2024.2.x

- RHEL 7 STIG Versi 3 Rilis 14
- RHEL 8 STIG Versi 1 Rilis 14
- RHEL 9 STIG Versi 1 Rilis 3
- Ubuntu 18.04 STIG Versi 2 Rilis 14
- Ubuntu 20.04 STIG Versi 1 Rilis 12
- Ubuntu 22.04 STIG Versi 1 Rilis 1

STIG-Build-Linux-Medium versi 2024.2.x

- RHEL 7 STIG Versi 3 Rilis 14
- RHEL 8 STIG Versi 1 Rilis 14
- RHEL 9 STIG Versi 1 Rilis 3
- Ubuntu 18.04 STIG Versi 2 Rilis 14
- Ubuntu 20.04 STIG Versi 1 Rilis 12
- Ubuntu 22.04 STIG Versi 1 Rilis 1

STIG-Build-Linux-High versi 2024.2.x

- RHEL 7 STIG Versi 3 Rilis 14
- RHEL 8 STIG Versi 1 Rilis 14
- RHEL 9 STIG Versi 1 Rilis 3
- Ubuntu 18.04 STIG Versi 2 Rilis 14
- Ubuntu 20.04 STIG Versi 1 Rilis 12
- Ubuntu 22.04 STIG Versi 1 Rilis 1

Perubahan Q1 2024 - 02/06/2024:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal pertama 2024 sebagai berikut:

STIG-Build-Linux-Low versi 2024.1.x

- RHEL 7 STIG Versi 3 Rilis 14
- RHEL 8 STIG Versi 1 Rilis 13
- Ubuntu 18.04 STIG Versi 2 Rilis 13
- Ubuntu 20.04 STIG Versi 1 Rilis 11

STIG-Build-Linux-Medium versi 2024.1.x

- RHEL 7 STIG Versi 3 Rilis 14
- RHEL 8 STIG Versi 1 Rilis 13
- Ubuntu 18.04 STIG Versi 2 Rilis 13
- Ubuntu 20.04 STIG Versi 1 Rilis 11

STIG-Build-Linux-High versi 2024.1.x

- RHEL 7 STIG Versi 3 Rilis 14
- RHEL 8 STIG Versi 1 Rilis 13
- Ubuntu 18.04 STIG Versi 2 Rilis 13
- Ubuntu 20.04 STIG Versi 1 Rilis 11

Perubahan Q4 2023 - 12/07/2023:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal keempat 2023 sebagai berikut:

STIG-Build-Linux-Low versi 2023.4.x

- RHEL 7 STIG Versi 3 Rilis 13
- RHEL 8 STIG Versi 1 Rilis 12
- Ubuntu 18.04 STIG Versi 2 Rilis 12
- Ubuntu 20.04 STIG Versi 1 Rilis 10

STIG-Build-Linux-Medium versi 2023.4.x

- RHEL 7 STIG Versi 3 Rilis 13
- RHEL 8 STIG Versi 1 Rilis 12
- Ubuntu 18.04 STIG Versi 2 Rilis 12
- Ubuntu 20.04 STIG Versi 1 Rilis 10

STIG-Build-Linux-High versi 2023.4.x

- RHEL 7 STIG Versi 3 Rilis 13
- RHEL 8 STIG Versi 1 Rilis 12
- Ubuntu 18.04 STIG Versi 2 Rilis 12
- Ubuntu 20.04 STIG Versi 1 Rilis 10

Perubahan Q3 2023 - 10/04/2023:

Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal ketiga 2023 sebagai berikut:

STIG-Build-Linux-Low versi 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-Medium versi 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-High versi 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-Low versi 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-Medium versi 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-High versi 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-Low versi 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-Medium versi 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-High versi 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-Low versi 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-Medium versi 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-High versi 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-Low versi 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-Medium versi 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-High versi 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 AL2 skrip sebelumnya dengan RHEL 7. Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal pertama 2022 sebagai berikut:

STIG-Build-Linux-Low versi 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-High versi 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-Low versi 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-High versi 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-Low versi 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-High versi 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-windows versi 2024.03.0

`scap-compliance-checker-windows`Komponen berjalan pada EC2 instance yang dibuat Image Builder untuk membangun dan menguji image. 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 MS, 2016, 2019, dan 2022.

Komponen pemeriksa kepatuhan SCAP untuk Windows mencakup tolok ukur berikut:

Versi SCC: 5.10

Tolok Ukur Q4 2023:

- u_ms_defender_antivirus_v2r5_stig_scap_1-2_patokan
- U_MS_ _Framework_4-0_v2r2_stig_scap_1-2_benchmark DotNet
- U_MS_ _v2r6_STIG_SCAP_1-2_Benchmark IE11
- u_ms_windows_2012_dan_2012_r2_dc_v3r5_stig_scap_1-2_patokan
- u_ms_windows_2012_dan_2012_r2_ms_v3r5_stig_scap_1-2_patokan
- u_ms_windows_defender_firewall_v2r3_stig_scap_1-2_patokan
- u_ms_windows_server_2016_v2r7_stig_scap_1-2_patokan
- u_ms_windows_server_2019_v3r2_stig_scap_1-2_patokan
- u_ms_windows_server_2022_v2r2_stig_scap_1-2_patokan
- u_can_ubuntu_20-04_lts_v1r10_stig_scap_1-2_patokan
- u_rhel_7_v3r15_stig_scap_1-3_patokan
- u_rhel_8_v1r13_stig_scap_1-3_patokan
- u_rhel_9_v2r1_stig_scap_1-3_patokan

scap-compliance-checker-linux versi 2021.04.0

`scap-compliance-checker-linux`Komponen berjalan pada EC2 instance yang dibuat Image Builder untuk membangun dan menguji image. 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 lokasi berikut pada instance build:`/opt/scc/SCCResults`.
5. Mencatat skor kepatuhan dari laporan lokal ke file log AWSTOE aplikasi.

> **Note**
>
> AWSTOE saat ini mendukung validasi kepatuhan SCAP untuk RHEL 7/8 dan Ubuntu 18.04/20.04. Aplikasi SCC saat ini mendukung arsitektur x86 untuk validasi.

Komponen pemeriksa kepatuhan SCAP untuk Linux mencakup tolok ukur berikut:

Versi SCC: 5.10

Tolok Ukur Q4 2023:

- u_can_ubuntu_20-04_lts_v1r10_stig_scap_1-2_patokan
- u_rhel_7_v3r15_stig_scap_1-3_patokan
- u_rhel_8_v1r13_stig_scap_1-3_patokan
- u_rhel_9_v2r1_stig_scap_1-3_patokan
- u_ms_defender_antivirus_v2r5_stig_scap_1-2_patokan
- U_MS_ _Framework_4-0_v2r2_stig_scap_1-2_benchmark DotNet
- U_MS_ _v2r6_STIG_SCAP_1-2_Benchmark IE11
- u_ms_windows_2012_dan_2012_r2_dc_v3r5_stig_scap_1-2_patokan
- u_ms_windows_2012_dan_2012_r2_ms_v3r5_stig_scap_1-2_patokan
- u_ms_windows_defender_firewall_v2r3_stig_scap_1-2_patokan
- u_ms_windows_server_2016_v2r7_stig_scap_1-2_patokan
- u_ms_windows_server_2019_v3r2_stig_scap_1-2_patokan
- u_ms_windows_server_2022_v2r2_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
Pembaruan SCAP Q1 2025	•	April 11, 2025

Perubahan	Deskripsi	Tanggal
	scap-compliance-checker-windows versi 2024.03.0 (Versi SCC: 5.10) • scap-compliance-checker-windows versi 2025.01.0 (Versi SCC: 5.10	
Pembaruan SCAP Q4 2023	• scap-compliance-checker-windows versi 2023.04.0 (Versi SCC: 5.8) • scap-compliance-checker-windows versi 2023.04.0 (Versi SCC: 5.8)	Desember 20, 2021
Pembaruan SCAP Q3 2023	• scap-compliance-checker-windows versi 2023.03.0 (Versi SCC: 5.7.2) • scap-compliance-checker-linux versi 2023.03.0 (Versi SCC: 5.7.2	13 November 2023
Ditambahkan komponen SCAP	Memperkenalkan komponen SCAP berikut: • scap-compliance-checker-linuxVersi yang dibuat 2021.04.0 (SCC Version: 5.4.2) • scap-compliance-checker-linuxVersi yang dibuat 2021.04.0 (SCC Version: 5.4.2)	Desember 20, 2021

Mengembangkan komponen kustom untuk image Image Builder Anda

Anda dapat membuat komponen Anda sendiri untuk menyesuaikan gambar Image Builder Anda sesuai dengan spesifikasi Anda yang tepat. Gunakan langkah-langkah berikut untuk mengembangkan komponen kustom untuk gambar Image Builder atau resep kontainer Anda.

1. Jika Anda ingin mengembangkan dokumen komponen Anda dan memvalidasinya secara lokal, Anda dapat menginstal aplikasi AWS Task Orchestrator and Executor (AWSTOE) dan mengaturnya di mesin lokal Anda. Untuk informasi selengkapnya, lihat Pengaturan manual untuk mengembangkan komponen khusus dengan AWSTOE.
2. Buat dokumen komponen yang menggunakan kerangka dokumen AWSTOE komponen. Untuk informasi selengkapnya tentang kerangka dokumen, lihatGunakan kerangka dokumen AWSTOE komponen untuk komponen kustom.
3. Tentukan dokumen komponen Anda saat membuat komponen kustom. Untuk informasi selengkapnya, lihat Buat komponen kustom dengan Image Builder.

Topik

- Membuat dokumen komponen YAMAL untuk komponen kustom di Image Builder
- Buat komponen kustom dengan Image Builder

Membuat dokumen komponen YAMAL untuk komponen kustom di Image Builder

Untuk membangun komponen, Anda harus menyediakan dokumen komponen aplikasi YAMAL atau JSON. Dokumen berisi kode yang berjalan selama fase dan langkah-langkah yang Anda tentukan untuk memberikan penyesuaian untuk gambar Anda.

Beberapa contoh di bagian ini membuat komponen build yang memanggil modul `Update0S` tindakan dalam aplikasi manajemen AWSTOE komponen. Modul memperbarui sistem operasi. Untuk informasi selengkapnya tentang modul `Update0S` tindakan, lihatUpdateOS.

Contoh sistem operasi macOS menggunakan modul `ExecuteBash` tindakan untuk menginstal dan memverifikasi utilitas. `wget` Modul `Update0S` tindakan tidak mendukung macOS. Untuk informasi selengkapnya tentang modul `ExecuteBash` tindakan, lihatExecuteBash. Untuk informasi

selengkapnya tentang fase, langkah, dan sintaks untuk dokumen komponen AWSTOE aplikasi, lihat Menggunakan dokumen di AWSTOE.

> **Note**
>
> Image Builder menentukan jenis komponen dari fase yang didefinisikan dalam dokumen 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 image. 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, lihatGunakan komponen untuk menyesuaikan image 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 dokumen komponen Anda. Contoh dokumentasi menggunakan file bernama*`update-linux-os.yaml`*, dengan 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 dokumen komponen Anda. Contoh dokumentasi menggunakan file bernama*update-windows-os.yaml*, dengan 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.

macOS

Buat file komponen YAMAL

Gunakan alat pengeditan file untuk membuat dokumen komponen Anda. Contoh dokumentasi menggunakan file bernama*wget-macos.yaml*, dengan konten berikut:

```
name: WgetInstallDocument
description: This is wget installation document.
schemaVersion: 1.0

phases:
  - name: build
    steps:
      - name: WgetBuildStep
        action: ExecuteBash
        inputs:
```

```
        commands:
          - |
            PATH=/usr/local/bin:$PATH
            sudo -u ec2-user brew install wget


- name: validate
  steps:
    - name: WgetValidateStep
      action: ExecuteBash
      inputs:
        commands:
          - |
            function error_exit {
              echo $1
              echo "{\"failureMessage\":\"$2\"}"
              exit 1
            }

            type wget
            if [ $? -ne 0 ]; then
              error_exit "$stderr" "Wget installation failed!"
            fi

- name: test
  steps:
    - name: WgetTestStep
      action: ExecuteBash
      inputs:
        commands:
          - wget -h
```

> **Tip**
>
> Gunakan alat seperti Validator YAMAL online ini, atau ekstensi lint YAMAL di lingkungan kode Anda untuk memverifikasi bahwa YAMAL Anda terbentuk dengan baik.

Buat komponen kustom dengan Image Builder

Setelah menyelesaikan dokumen komponen, Anda dapat menggunakannya untuk membuat komponen kustom yang dapat digunakan resep Image Builder Anda. Anda dapat membuat

komponen kustom dari konsol Image Builder, dari API atau SDKs, atau dari baris perintah. Untuk informasi selengkapnya tentang cara membuat komponen kustom dengan parameter input dan menggunakannya dalam resep Anda, lihatTutorial: Buat komponen kustom dengan parameter input.

Bagian berikut menunjukkan cara membuat komponen dari konsol atau dari AWS CLI.

Daftar Isi

- Buat komponen khusus dari konsol
- Buat komponen kustom dari AWS CLI
- Impor skrip untuk membuat komponen dari AWS CLI

Buat komponen khusus dari konsol

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/.
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 pengujian untuk membuat image Anda.

    Di kotak Konten, masukkan konten dokumen komponen YAMB Anda. Untuk memulai dengan contoh Hello World untuk Linux, pilih opsi Use example. Untuk mempelajari lebih lanjut tentang cara membuat dokumen komponen YAMB, atau menyalin dan menempelkan contoh UpdateOS

dari halaman tersebut, lihat. Membuat dokumen komponen YAMAL untuk komponen kustom di Image Builder

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.

Perbarui 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. Pembuatan versi semantik di Image Builder

Buat komponen kustom dari 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://amzn-s3-demo-destination-bucket/my-path/update-linux-os.yaml
  ```

Buat komponen dari dokumen YAMB

Untuk merampingkan create-component perintah yang Anda gunakan di 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 sebelumnya. Pasangan `uri` kunci-nilai berisi referensi file.

> **Note**
>
> Konvensi penamaan untuk nilai data dalam file JSON mengikuti pola yang ditentukan untuk parameter permintaan operasi Image Builder API. Untuk meninjau parameter permintaan perintah API, lihat CreateComponentperintah 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://amzn-s3-demo-destination-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"
 }
}
```

2. Buat komponen

   Gunakan perintah berikut untuk membuat komponen, mereferensikan 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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

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://amzn-s3-demo-destination-bucket/my-path/update-windows-os.yaml
  ```

Buat komponen dari dokumen YAMB

Untuk merampingkan create-component perintah yang Anda gunakan di 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 sebelumnya. Pasangan `uri` kunci-nilai berisi referensi file.

> **Note**
>
> Konvensi penamaan untuk nilai data dalam file JSON mengikuti pola yang ditentukan untuk parameter permintaan operasi Image Builder API. Untuk meninjau parameter permintaan perintah API, lihat CreateComponentperintah 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://amzn-s3-demo-destination-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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

2. Buat komponen

Gunakan perintah berikut untuk membuat komponen, mereferensikan 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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

macOS

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 wget-macos.yaml s3://amzn-s3-demo-destination-bucket/my-path/wget-macos.yaml
  ```

Buat komponen dari dokumen YAMB

Untuk merampingkan create-component perintah yang Anda gunakan di AWS CLI, buat file JSON yang berisi semua parameter komponen yang ingin Anda lewatkan ke perintah. Sertakan lokasi *wget-macos.yaml* dokumen yang Anda buat sebelumnya. Pasangan `uri` kunci-nilai berisi referensi file.

> **Note**
>
> Konvensi penamaan untuk nilai data dalam file JSON mengikuti pola yang ditentukan untuk parameter permintaan operasi Image Builder API. Untuk meninjau parameter permintaan perintah API, lihat CreateComponentperintah 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*install-wget-macos-component.json*. Sertakan konten berikut:

   ```
   {
   ```

```
 "name": "install install-wget-macos-component",
 "semanticVersion": "1.1.2",
 "description": "An example component that installs and verifies the wget
 utility on macOS.",
 "changeDescription": "Initial version.",
 "platform": "macOS",
 "uri": "s3://amzn-s3-demo-destination-bucket/my-path/wget-macos.yaml",
 "kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/98765432-
b123-456b-7f89-0123456f789c",
 "tags": {
  "MyTagKey-purpose": "install-software"
 }
}
```

2. Buat komponen

   Gunakan perintah berikut untuk membuat komponen, mereferensikan nama file untuk file JSON yang Anda buat pada langkah sebelumnya:

   ```
   aws imagebuilder create-component --cli-input-json file://install-wget-macos-
   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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

AWSTOE versi komponen untuk pembaruan dari AWS CLI

AWSTOE nama 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. Pembuatan versi semantik di Image Builder

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 apa yang seharusnya.

Impor skrip untuk membuat komponen dari 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. *amzn-s3-demo-source-bucket* Saat `SHELL` ini, didukung untuk komponen`format`.

```
{
"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://amzn-s3-demo-source-bucket/AdminConfig.ps1",
"kmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/60763706-
b131-418b-8f85-3420912f020c"
}
```

Untuk membuat komponen dari skrip yang diimpor, jalankan perintah berikut.

```
aws imagebuilder import-component --cli-input-json file://import-component.json
```

> **Note**
>
> 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, lihatHapus sumber daya Image Builder yang sudah usang atau tidak digunakan.

Bagaimana Image Builder menggunakan AWS Task Orchestrator and Executor aplikasi untuk mengelola komponen

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

AWSTOE adalah aplikasi mandiri yang diinstal Image Builder pada build dan instance pengujiannya saat Anda membuat gambar. Anda juga dapat menginstalnya secara manual pada EC2 instance untuk membuat komponen kustom Anda sendiri. Itu tidak memerlukan pengaturan tambahan, dan juga dapat berjalan di tempat.

Daftar Isi

- AWSTOE download
- Wilayah yang Didukung
- AWSTOE referensi perintah
- Pengaturan manual untuk mengembangkan komponen khusus dengan AWSTOE
- Gunakan kerangka dokumen AWSTOE komponen untuk komponen kustom
- Modul aksi didukung oleh manajer AWSTOE komponen
- Konfigurasikan input untuk perintah AWSTOE run

AWSTOE download

Untuk menginstal AWSTOE, 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 mengetahui informasi selengkapnya, lihat Membuat kebijakan titik akhir VPC untuk Image Builder.

> **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.

Arsitektur	Platform	Tautan unduhan	Contoh
386	AL 2 dan 2023 RHEL 7, 8, dan 9	`https://awstoe-<region>.s3`	https://awstoe-us-east-1.s3.us-east

Arsitektur	Platform	Tautan unduhan	Contoh
	Ubuntu 16.04, 18.04, 20.04, 22.04, dan 24.04 CentOS 7 dan 8 SUSE 12 dan 15	`s.com/latest/linux/386/awstoe`	-1.amazonaws.com/latest/linux/386/awstoe
AMD64	AL 2 dan 2023 RHEL 7, 8, dan 9 Ubuntu 16.04, 18.04, 20.04, 22.04, dan 24.04 CentOS 7 dan 8 CentOS Aliran 8 SUSE 12 dan 15	`https://awstoe-<region>.s3.s.com/latest/linux/amd64/awstoe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/amd64/awstoe
AMD64	macOS 10.14.x (Mojave), 10.15.x (Catalina), 11.x (Big Sur), 12.x (Monterey)	`https://awstoe-region.s3.s.com/latest/darwin/amd64/awstoe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/darwin/amd64/awstoe
AMD64	Windows Server 2012 R2, 2016, 2019, dan 2022	`https://awstoe-<region>.s3.s.com/latest/windows/amd64/awstoe.exe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/windows/amd64/awstoe.exe

Arsitektur	Platform	Tautan unduhan	Contoh
ARM64	AL 2 dan 2023 RHEL 7, 8, dan 9 Ubuntu 16.04, 18.04, 20.04, 22.04, dan 24.04 CentOS 7 dan 8 CentOS Aliran 8 SUSE 12 dan 15	`https://awstoe- <region>.s3.amazonaws.com/latest/linux/arm64/awstoe`	https://awstoe-us-east-1.s3.us-east-1.amazonaws.com/latest/linux/arm64/awstoe

Wilayah yang Didukung

AWSTOE didukung sebagai aplikasi mandiri di Wilayah berikut.

Wilayah AWS nama	Wilayah AWS
AS Timur (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
AS Barat (California Utara)	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

Wilayah AWS nama	Wilayah AWS
Asia Pacific (Seoul)	ap-northeast-2
Asia Pasifik (Mumbai)	ap-south-1
Asia Pasifik (Hyderabad)	ap-south-2
Asia Pasifik (Singapura)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pasifik (Jakarta)	ap-southeast-3
Asia Pasifik (Tokyo)	ap-northeast-1
Canada (Central)	ca-sentral-1
Europe (Frankfurt)	eu-central-1
Eropa (Zürich)	eu-central-2
Eropa (Stockholm)	eu-north-1
Eropa (Milan)	eu-south-1
Eropa (Spanyol)	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)	me-central-1
Timur Tengah (Bahrain)	me-selatan-1
Amerika Selatan (São Paulo)	sa-east-1

Wilayah AWS nama	Wilayah AWS
Tiongkok (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1

AWSTOE referensi perintah

AWSTOE adalah aplikasi manajemen komponen baris perintah yang berjalan pada EC2 instance Amazon. Saat Image Builder meluncurkan instance EC2 build atau test, Image Builder akan diinstal AWSTOE pada instance. Kemudian menjalankan AWSTOE perintah di AWS CLI untuk menginstal atau memvalidasi komponen yang ditentukan dalam resep gambar atau wadah.

> **Note**
>
> Beberapa modul AWSTOE tindakan memerlukan izin tinggi untuk berjalan di server Linux. Untuk menggunakan izin yang ditinggikan, awali sintaks perintah dengansudo, 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, lihatModul 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, lihatKonfigurasikan 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

`LogGroup`Nama untuk CloudWatch Log.

--cw-log-region

Bentuk pendek: N/A

AWS Wilayah yang berlaku untuk CloudWatch Log.

--cw-log-stream

Bentuk pendek: N/A

`LogStream`Nama 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*).
- S3 URIs (`s3://`*bucket/key*).
- Versi pembuatan komponen Image Builder ARNs (arn:aws:imagebuilder:us-west-:component/ /2021.12.02/1). *2:123456789012 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 yang 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*)
- S3 URIs () s3://*bucket/key*
- Versi pembuatan komponen Image Builder ARNs (arn:aws:imagebuilder:us-west-:component/ /2021.12.02/1) *2:123456789012 my-example-component*

> **Note**
>
> Tidak ada spasi di antara item dalam daftar, hanya koma.

Opsi

--help

Bentuk pendek: -h

Menampilkan manual bantuan untuk menggunakan opsi aplikasi manajemen komponen.

--jejak

Bentuk pendek: -t

Mengaktifkan pencatatan verbose ke konsol.

Pengaturan manual untuk mengembangkan komponen khusus 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 secara manual dan menggunakannya sebagai aplikasi yang berdiri sendiri untuk mengembangkan komponen khusus. Image Builder menangani langkah-langkah ini untuk Anda, jika Anda menggunakan konsol Image Builder atau AWS CLI perintah untuk membuat komponen kustom. Untuk informasi selengkapnya, lihat Buat komponen kustom dengan Image Builder.

Verifikasi tanda tangan unduhan AWSTOE instalasi

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

Topik

- Verifikasi tanda tangan unduhan AWSTOE instalasi di Linux atau macOS
- Verifikasi tanda tangan unduhan AWSTOE instalasi di Windows

Verifikasi tanda tangan unduhan AWSTOE instalasi di Linux atau macOS

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

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 AWSTOE aplikasi diubah atau rusak, jangan jalankan file instalasi. Sebagai gantinya, hubungi Dukungan Untuk informasi selengkapnya tentang opsi dukungan Anda, lihat Dukungan.

AWSTOE file untuk sistem operasi berbasis Linux dan macOS ditandatangani menggunakanGnuPG, implementasi open source dari standar Pretty Good Privacy (OpenPGP) untuk tanda tangan digital

yang aman. GnuPG(juga dikenal sebagaiGPG) 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 CLI EC2 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, Unix, atau macOS, alat GPG kemungkinan sudah diinstal. Untuk menguji apakah alat ini sudah terinstal di sistem Anda, ketikkan gpg pada jendela perintah. Jika alat GPG telah teriinstal, 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, pilih sistem operasi yang cocok dengan instans Anda.

Debian-based Linux

Dari terminal, jalankan perintah berikut:

```
apt-get install gnupg
```

Red Hat–based Linux

Dari terminal, jalankan perintah berikut:

```
yum install gnupg
```

macOS

Dari terminal, jalankan perintah berikut:

```
brew 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. Misalnya, 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
     Sle3l9hz6wCClz1l9LbBzTyHfSm5ucTXvNe88XX5Gmt37OCDM7vfli0Ctv8WFoLN
     6jbxuA/sV71yIkPm9IYp3+GvaKeT870+sn8/JOOKE/U4sJV1ppbqmuUzDfhrZUaw
     8eW8IN9A1FTIuWiZED/5L83UZuQs1S7s2PjlABEBAAG0GkFXU1RPRSA8YXdzdG9l
     QGFtYXpvbi5jb20+iQE5BBMBCAAjBQJfFKsLAhsDBwsJCAcDAgEGFQgCCQoLBBYC
     AwECHgECF4AACgkQ3r3BVvWuvFJGiwf9EVmrBR77+Qe/DUeXZJYoaFr7If/fVDZl
     6V3TC6p0J0Veme7uXleRUTFOjzbh+7e5sDX19HrnPquzCnzfMiqbp4lSoeUuNdOf
     FcpuTCQH+M+sIEIgPno4PLl0Uj2uE1o++mxmonBl/Krk+hly8hB2L/9n/vW3L7BN
     OMb1Ll9PmgGPbWipcT8KRdz4SUex9TXGYzjlWb3jU3uXetdaQY1M3kVKE1siRsRN
     YYDtpcjmwbhjpu4xm19aFqNoAHCDctEsXJA/mkU3erwIRocPyjAZE2dnlkL9ZkFZ
     z9DQkcIarbCnybDM5lemBbdhXJ6hezJE/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 F5AEBC52: 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 `F5AEBC52`.

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 Dukungan.

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
```

Misalnya:

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

Nilai yang didukung untuk **`architecture`** bisa`amd64`,`386`, 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
```

Misalnya:

```
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 F5AEBC52
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 Dukungan.

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 Dukungan.

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

BA 81 25 EE AC 64 2E A9 F3 C5 93 CA 6D 3E B7 93 7D 68 75 74

> **Note**
>
> Selama jendela peluncuran untuk biner baru, sertifikat penandatangan Anda mungkin tidak cocok dengan sidik jari baru. Jika sertifikat penandatangan Anda tidak cocok, verifikasi bahwa nilai cap jempol adalah:
> F8 83 11 EE F0 4A A2 91 E3 79 21 BA 6B FC AF F8 19 92 12 D7

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:

     BA 81 25 EE AC 64 2E A9 F3 C5 93 CA 6D 3E B7 93 7D 68 75 74

     maka AWSTOE biner yang Anda unduh asli dan dapat diinstal dengan aman.
   - 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 kredensyal
- 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 download

   > **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, lihatVerifikasi tanda tangan di Linux atau macOS. Untuk memverifikasi unduhan Anda di server Windows, lihatVerifikasi tanda tangan di Windows.

> **Note**
>
> 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 EC2 instance untuk mengembangkan dan menguji AWSTOE komponen.

Langkah 2: Tetapkan AWS kredensyal

AWSTOE memerlukan AWS kredensi 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 sebuah EC2 instance, maka menjalankan AWSTOE menggunakan izin yang sama dengan peran IAM yang dilampirkan ke instance. EC2

Untuk informasi selengkapnya tentang peran IAM EC2, lihat peran IAM untuk Amazon. EC2

Contoh berikut menunjukkan cara mengatur AWS kredensyal 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`.

```
$env:AWS_ACCESS_KEY_ID=your_access_key_id
```

```
$env:AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

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

```
set AWS_ACCESS_KEY_ID=your_access_key_id
```

```
set AWS_SECRET_ACCESS_KEY=your_secret_access_key
```

Langkah 3: Kembangkan dokumen komponen secara lokal

Komponen ditulis dengan dokumen YAMAL plaintext. Untuk informasi selengkapnya tentang sintaks dokumen, lihatGunakan kerangka dokumen AWSTOE komponen untuk komponen kustom.

Berikut ini adalah contoh dokumen komponen Hello World untuk membantu Anda memulai.

Linux

Beberapa contoh komponen Linux dalam panduan ini mengacu pada file dokumen komponen bernama`hello-world-linux.yml`. Anda dapat menggunakan dokumen berikut untuk memulai dengan contoh-contoh tersebut.

```
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.'
```

Windows

Beberapa contoh komponen Windows dalam panduan ini merujuk ke file dokumen komponen bernama`hello-world-windows.yml`. Anda dapat menggunakan dokumen berikut untuk memulai dengan contoh-contoh tersebut.

```
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.'
```

macOS

Beberapa contoh komponen macOS dalam panduan ini merujuk ke file dokumen komponen bernama. `hello-world-macos.yml` Anda dapat menggunakan dokumen berikut untuk memulai dengan contoh-contoh tersebut.

```
name: Hello World
description: This is hello world testing document for macOS.
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.

Linux atau macOS

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

Windows

```
awstoe.exe validate --documents C:\Users\user\Documents\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` are`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.yaml`, diikuti oleh `build` dan `validate` fase`document2.yaml`.

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 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 amzn-s3-demo-destination-bucket --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 build,test
```

Dapatkan bantuan dengan AWSTOE

```
awstoe --help
```

Gunakan kerangka dokumen AWSTOE komponen untuk komponen kustom

Untuk membangun komponen menggunakan framework komponen 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 gunakan komponen Anda saat mereka membuat Amazon Machine Image (AMI) atau image container baru.

Topik

- Alur kerja dokumen komponen
- Pencatatan komponen
- Input dan output rantai
- Skema dan definisi dokumen
- Contoh dokumen
- Gunakan variabel dalam dokumen komponen kustom Anda
- Gunakan konstruksi bersyarat di AWSTOE
- Gunakan operator perbandingan dalam dokumen AWSTOE komponen
- Gunakan operator logis dalam dokumen AWSTOE komponen
- 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, input/output nilai, dan contoh, lihatModul aksi didukung oleh manajer AWSTOE komponen.

Pencatatan komponen

AWSTOE membuat folder log baru pada EC2 instance 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 dan macOS: `/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:

File 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.

Input dan output rantai

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. Misalnya:
  - 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 YAMM 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	Wajib
name	Nama dokumen.	String	Tidak
deskripsi	Deskripsi dokumen.	String	Tidak
schemaVersion	Versi skema dokumen, saat ini 1.0.	Tali	Ya

Bidang	Deskripsi	Tipe	Wajib
fase	Daftar fase dengan langkah-langkahnya.	Daftar	Ya

Definisi skema untuk suatu fase adalah sebagai berikut.

Bidang	Deskripsi	Tipe	Wajib
name	Nama fase.	Tali	Ya
langkah	Daftar langkah-l angkah dalam fase.	Daftar	Ya

Definisi skema untuk suatu langkah adalah sebagai berikut.

Bidang	Deskripsi	Tipe	Wajib	Nilai default
name	Nama yang ditentukan pengguna untuk langkah tersebut.	String		
tindakan	Kata kunci yang berkaitan dengan modul yang menjalank an langkah.	String		
timeoutSeconds	Jumlah detik langkah berjalan sebelum gagal atau mencoba lagi. Juga, mendukung	Bilangan Bulat	Tidak	7.200 detik (120 menit)

Bidang	Deskripsi	Tipe	Wajib	Nilai default
	-1 nilai, yang menunjukkan batas waktu tak terbatas. 0 dan nilai negatif lainnya tidak diperbolehkan.			

Bidang	Deskripsi	Tipe	Wajib	Nilai default
onFailure	Menentukan apa yang harus dilakukan langkah jika terjadi kegagalan. Nilai yang valid adalah sebagai berikut: • Batalkan — Gagal langkah setelah jumlah upaya maksimum, dan berhenti berjalan. Menetapkan status untuk fase dan dokumen ke`Failed`. • Lanjutkan — Gagal langkah setelah jumlah upaya maksimum, dan terus menjalankan langkah-langkah yang tersisa. Menetapkan status untuk	String	Tidak	Batalkan

Bidang	Deskripsi	Tipe	Wajib	Nilai default
	fase dan dokumen ke`Failed`. • Abaikan - Tetapkan langkah ke `IgnoredFailure` setelah jumlah maksimum upaya yang gagal, dan terus menjalankan langkah-langkah yang tersisa. Menetapkan status untuk fase dan dokumen ke`SuccessWithIgnoredFailure` .			
maxAttempts	Jumlah maksimum upaya yang diizinkan sebelum gagal langkah.	Bilangan Bulat	Tidak	1

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

Contoh dokumen

Contoh berikut menunjukkan dokumen AWSTOE komponen yang melakukan tugas untuk sistem operasi target.

Linux

Contoh 1: Jalankan file biner kustom

Berikut ini adalah contoh dokumen yang men-download dan menjalankan file biner kustom pada instance Linux.

```
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>amzn-s3-demo-source-bucket</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 }}'
```

Windows

Contoh 1: Instal pembaruan Windows

Berikut ini adalah contoh dokumen yang 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}}'
```

Contoh 2: Instal AWS CLI pada instance Windows

Berikut ini adalah contoh dokumen yang 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 }}'
```

Contoh 3: Instal AWS CLI dengan installer MSI

Berikut ini adalah contoh dokumen yang menginstal AWS CLI dengan 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 }}'
```

macOS

Contoh 1: Jalankan file biner macOS kustom

Berikut ini adalah contoh dokumen yang mengunduh dan menjalankan file biner kustom pada instance macOS.

```
name: macOSBin
description: Download and run a binary file on macOS.
```

```
schemaVersion: 1.0
phases:
  - name: build
    steps:
      - name: Download
        action: S3Download
        inputs:
          - source: s3://<replaceable>amzn-s3-demo-source-bucket</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 }}'
```

Gunakan variabel dalam dokumen komponen kustom Anda

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.

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 YAMM.

Sintaksis

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

Nama kunci	Wajib	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`string`.

Referensi nilai konstan dalam dokumen

Anda dapat mereferensikan konstanta dalam input langkah atau loop di dalam dokumen YAMM 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 di 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 }}
```

Parameter

Parameter adalah variabel yang dapat berubah, dengan pengaturan yang dapat disediakan aplikasi pemanggilan 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 YAMM.

Sintaksis

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

Nama kunci	Wajib	Deskripsi
`name`	Ya	Nama parameter. Harus unik untuk dokumen (tidak boleh

Nama kunci	Wajib	Deskripsi
		sama dengan nama parameter atau konstanta lainnya).
`type`	Ya	Tipe data parameter. Jenis yang didukung meliputi: `string`.
`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 YAMM 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 di dalam kurung kurawal, dan secara otomatis dipangkas. Misalnya, semua referensi berikut valid:

  `{{ MyParameter }}`, `{{ MyParameter}}`, `{{MyParameter }}`, `{{MyParameter}}`
- Referensi dalam dokumen YAMAL 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 YAMM 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	Wajib	Deskripsi
--parameter *name* =*value*,...	Tidak	Opsi ini mengambil daftar pasangan kunci-nilai, dengan nama parameter sebagai kunci.

Contoh

Contoh berikut menunjukkan cara menggunakan parameter dalam dokumen YAMM 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 kunci-nilai parameter dengan `--parameter` opsi di: AWS CLI

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

Gunakan parameter Systems Manager Parameter Store

Anda dapat mereferensikan AWS Systems Manager parameter Parameter Store (parameter SSM) dalam dokumen komponen Anda dengan awalan variabel. `aws:ssm` Misalnya,

`{{ aws:ssm:/my/param }}`menyelesaikan ke nilai parameter SSM. `/my/param`

Fitur ini mendukung jenis parameter SSM berikut:

- String - Peta ke tipe AWSTOE string.
- StringList — Peta ke AWSTOE `stringList` jenis.
- SecureString — Peta ke tipe AWSTOE string.

Untuk informasi selengkapnya tentang Parameter Store lihat AWS Systems Manager Parameter Menyimpan di Panduan AWS Systems Manager Pengguna.

Anda juga dapat mereferensikan AWS Secrets Manager rahasia menggunakan parameter `SecureString` SSM. Sebagai contoh: `{{ aws:ssm:/aws/reference/secretsmanager/test/test-secret }}`. Untuk informasi selengkapnya, lihat Mereferensikan AWS Secrets Manager rahasia dari parameter Parameter Store.

> **Important**
>
> Image Builder mengecualikan resolusi `SecureString` parameter dari lognya. Namun, Anda juga bertanggung jawab untuk memastikan bahwa informasi sensitif tidak dicatat melalui perintah yang dikeluarkan dalam dokumen komponen. Misalnya, jika Anda menggunakan `echo` perintah dengan string aman, perintah tersebut menulis nilai plaintext ke log.

Izin IAM yang diperlukan

Untuk menggunakan parameter Systems Manager di komponen Anda, peran instans Anda harus memiliki `ssm:GetParameter` izin untuk ARN sumber daya parameter. Misalnya:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": "ssm:GetParameter",
   "Resource": "arn:aws:ssm:*:111122223333:parameter/ImageBuilder-*"
  }
 ]
}
```

Untuk mengakses nilai terenkripsi, Anda juga memerlukan izin berikut:

- Tambahkan `kms:Decrypt` `SecureString` parameter atau AWS Secrets Manager nilai yang dienkripsi dengan pelanggan yang dikelola. AWS KMS key
- Tambahkan `secretsmanager:GetSecretValue` jika Anda mereferensikan rahasia Secrets Manager.

Referensi parameter SSM dalam dokumen komponen

Contoh berikut menunjukkan cara mereferensikan parameter Systems Manager Parameter Store parameter Systems Manager dalam komponen:

```
name: UseSSMParameterVariable
description: This is a sample component document that prints out the value of an SSM
 Parameter. Never do this for a SecureString parameter.
schemaVersion: 1.0

phases:
  - name: verify
    steps:
      - name: EchoParameterValue
        action: ExecuteBash
```

```
        inputs:
          commands:
            - echo "Log SSM parameter name: /my/test/param, value {{ aws:ssm:/my/test/
param }}."
```

Resolusi variabel runtime dinamis untuk parameter SSM

AWSTOE menyediakan fungsi bawaan berikut yang dapat Anda gunakan dalam referensi variabel untuk memanipulasi atau mengubah nilai saat runtime.

menyelesaikan fungsi

`resolve`Fungsi ini menyelesaikan referensi variabel di dalam referensi variabel lain, memungkinkan untuk referensi nama variabel dinamis. Ini berguna saat bekerja dengan parameter SSM di mana bagian dari jalur parameter dapat bervariasi dan diteruskan sebagai parameter dokumen.

`resolve`Fungsi ini hanya mendukung resolusi dinamis dari bagian nama parameter SSM.

Sintaksis

`dynamic_variable`Dalam contoh berikut mewakili nama parameter SSM, dan harus menjadi salah satu dari berikut ini:

- Referensi parameter SSM (misalnya,`aws:ssm:`*`/my/param`*)
- Referensi parameter dokumen komponen (misalnya,*`parameter-name`*)

```
{{ aws:ssm:resolve(dynamic_variable) }}
```

Contoh: Menyelesaikan parameter SSM saat runtime

Contoh berikut menunjukkan cara menggunakan `resolve` fungsi dalam dokumen komponen YAMM:

```
name: SsmParameterTest
description: This component verifies an SSM parameter variable reference with the echo
 command.
schemaVersion: 1.0

parameters:
  - parameter-name:
      type: string
      description: "test"
```

```
phases:
  - name: validate
    steps:
      - name: PrintDynamicVariable
        action: ExecuteBash
        inputs:
          commands:
            - echo "{{ aws:ssm:resolve(parameter-name) }}"
```

Gunakan konstruksi bersyarat di AWSTOE

Konstruksi bersyarat melakukan tindakan yang berbeda dalam dokumen komponen Anda berdasarkan apakah ekspresi kondisional yang ditentukan mengevaluasi atau. `true` `false` Anda dapat menggunakan `if` konstruksi untuk mengontrol aliran eksekusi dalam dokumen komponen Anda.

jika Membangun

Anda dapat menggunakan `if` konstruksi untuk mengevaluasi apakah suatu langkah harus dijalankan atau tidak. Secara default, ketika ekspresi `if` kondisional mengevaluasi`true`, AWSTOE menjalankan langkah, dan ketika kondisi dievaluasi`false`, AWSTOE lewati langkah. Jika sebuah langkah dilewati, itu diperlakukan sebagai langkah yang berhasil ketika AWSTOE mengevaluasi apakah fase dan dokumen berjalan dengan sukses.

> **Note**
>
> `if`Pernyataan hanya dievaluasi satu kali, bahkan jika langkah tersebut memicu restart. Jika sebuah langkah dimulai ulang, ia mengakui bahwa `if` pernyataan tersebut telah dievaluasi, dan berlanjut di mana ia berhenti.

Sintaksis

```
if:
  - <conditional expression>:
      [then: <step action>]
      [else: <step action>]
```

Nama kunci	Wajib	Deskripsi
ekspresi bersyarat	Ya	Ekspresi kondisional dapat berisi persis salah satu jenis operator berikut di tingkat atas. • Operator perbandingan — Untuk daftar operator perbandingan, dan informasi tentang cara kerjanya dalam dokumen AWSTOE komponen, lihatOperator perbandingan. • Operator logis — Operator logis mencakup `and`or`,, dan`not`, dan beroperasi pada satu atau lebih operator pembanding. Untuk informasi selengkapnya tentang cara kerja operator logis dalam dokumen AWSTOE komponen, lihatOperator logis. Jika ekspresi Anda harus memenuhi beberapa kondisi, gunakan operator logis untuk menentukan kondisi Anda.
lalu	Tidak	Mendefinisikan tindakan yang harus diambil jika ekspresi

Nama kunci	Wajib	Deskripsi
		kondisional mengevaluasi. `true`
lain	Tidak	Mendefinisikan tindakan yang harus diambil jika ekspresi kondisional mengevaluasi. `false`
langkah tindakan	Bersyarat	Saat Anda menggunakan `then` atau`else`, Anda harus menentukan salah satu tindakan langkah berikut: • Batalkan — AWSTOE menandai langkah sebagai gagal. • Jalankan — AWSTOE jalankan langkahnya. • Lewati — AWSTOE lewati langkahnya.

Contoh 1: Instal paket

Contoh langkah berikut dari dokumen AWSTOE komponen menggunakan operator logis untuk menguji nilai parameter dan menjalankan perintah manajer paket yang sesuai untuk menginstal aplikasi jika paket di-unzip.

```
- name: InstallUnzipAptGet
    action: ExecuteBash
    if:
      and:
          - binaryExists: 'apt-get'
          - not:
```

```
                binaryExists: 'unzip'
      inputs:
        commands:
            - sudo apt-get update
            - sudo apt-get install -y unzip

    - name: InstallUnzipYum
      action: ExecuteBash
      if:
        and:
            - binaryExists: 'yum'
            - not:
                binaryExists: 'unzip'
      inputs:
        commands:
            - sudo yum install -y unzip

    - name: InstallUnzipZypper
      action: ExecuteBash
      if:
        and:
            - binaryExists: 'zypper'
            - not:
                binaryExists: 'unzip'
      inputs:
        commands:
            - sudo zypper refresh
            - sudo zypper install -y unzip
```

Contoh 2: Lewati satu langkah

Contoh berikut menunjukkan dua cara untuk melewati satu langkah. Satu menggunakan operator logis, dan satu menggunakan operator perbandingan dengan tindakan `Skip` langkah.

```
# Creates a file if it does not exist using not
- name: CreateMyConfigFile-1
  action: ExecuteBash
  if:
    not:
      fileExists: '/etc/my_config'
  inputs:
    commands:
      - echo "Hello world" > '/etc/my_config'
```

```
# Creates a file if it does not exist using then and else
- name: CreateMyConfigFile-2
  action: ExecuteBash
  if:
    fileExists: '/etc/my_config'
    then: Skip
    else: Execute
  inputs:
    commands:
      - echo "Hello world" > '/etc/my_config'
```

Gunakan operator perbandingan dalam dokumen AWSTOE komponen

Anda dapat menggunakan operator perbandingan berikut dengan modul Menegaskan tindakan dan dengan ekspresi bersyarat yang menggunakan. jika Membangun Operator perbandingan dapat beroperasi pada satu nilai, misalnya`stringIsEmpty`, atau dapat membandingkan nilai dasar dengan nilai kedua (nilai variabel) untuk menentukan apakah ekspresi kondisional mengevaluasi atau. `true` `false`

Jika perbandingan beroperasi pada dua nilai, nilai kedua dapat berupa variabel rantai.

Saat membandingkan nilai dari jenis yang berbeda, konversi nilai berikut dapat terjadi sebelum perbandingan:

- Untuk perbandingan numerik, jika nilai variabel adalah string, AWSTOE mengkonversi string ke angka sebelum evaluasi. Jika konversi tidak memungkinkan, perbandingan kembali`false`. Misalnya, jika nilai variabelnya`"1.0"`, konversi berfungsi, tetapi jika nilai variabelnya adalah `"a10"` konversi gagal.
- Untuk perbandingan string, jika nilai variabel adalah angka, AWSTOE mengubahnya menjadi string sebelum evaluasi.

Bandingkan string

Operator perbandingan berikut bekerja dengan string untuk membandingkan nilai, untuk menguji spasi atau string kosong, atau untuk membandingkan nilai input dengan pola regex. Perbandingan string tidak peka huruf besar/kecil, dan tidak memangkas spasi dari awal atau akhir input string.

Operator perbandingan string

- stringIsEmpty

- stringIsWhitespace
- stringEquals
- stringLessThan
- stringLessThanEquals
- stringGreaterThan
- stringGreaterThanEquals
- patternMatches

stringIsEmpty

`stringIsEmpty`Operator kembali `true` jika string tertentu tidak mengandung karakter apapun. Misalnya:

```
# Evaluates to true
stringIsEmpty: ""

# Evaluates to false
stringIsEmpty: " "

# Evaluates to false
stringIsEmpty: "Hello."
```

stringIsWhitespace

Tes jika string ditentukan untuk hanya `stringIsWhitespace` berisi spasi. Misalnya:

```
# Evaluates to true
stringIsWhitespace: "   "

# Evaluates to false
stringIsWhitespace: ""

# Evaluates to false
stringIsWhitespace: " Hello?"
```

StringEquals

Tes jika string ditentukan untuk `stringEquals` adalah sama persis untuk string yang ditentukan dalam `value` parameter. Misalnya:

```
# Evaluates to true
stringEquals: 'Testing, testing...'
value: 'Testing, testing...'

# Evaluates to false
stringEquals: 'Testing, testing...'
value: 'Hello again.'

# Evaluates to false
stringEquals: 'Testing, testing...'
value: 'TESTING, TESTING....'

# Evaluates to false
stringEquals: 'Testing, testing...'
value: '   Testing, testing...'

# Evaluates to false
stringEquals: 'Testing, testing...'
value: 'Testing, testing...   '
```

stringLessThan

Tes jika string yang `stringLessThan` ditentukan untuk kurang dari string yang ditentukan dalam `value` parameter. Misalnya:

```
# Evaluates to true
# This comparison operator isn't case sensitive
stringlessThan: 'A'
value: 'a'

# Evaluates to true - 'a' is less than 'b'
stringlessThan: 'b'
value: 'a'

# Evaluates to true
# Numeric strings compare as less than alphabetic strings
stringlessThan: 'a'
value: '0'

# Evaluates to false
stringlessThan: '0'
value: 'a'
```

stringLessThanSama dengan

Tes jika string yang `stringLessThanEquals` ditentukan untuk kurang dari atau sama dengan string yang ditentukan dalam `value` parameter. Misalnya:

```
# Evaluates to true - 'a' is equal to 'a'
stringLessThanEquals: 'a'
value: 'a'

# Evaluates to true - since the comparison isn't case sensitive, 'a' is equal to 'A'
stringLessThanEquals: 'A'
value: 'a'

# Evaluates to true - 'a' is less than 'b'
stringLessThanEquals: 'b'
value: 'a'

# Evaluates to true - '0' is less than 'a'
stringLessThanEquals: 'a'
value: '0'

# Evaluates to false - 'a' is greater than '0'
stringLessThanEquals: '0'
value: 'a'
```

stringGreaterThan

Tes jika string ditentukan untuk `stringGreaterThan` lebih besar dari string yang ditentukan dalam `value` parameter. Misalnya:

```
# Evaluates to false - since the comparison isn't case sensitive, 'A' is equal to
 'a'
stringGreaterThan: 'a'
value: 'A'

# Evaluates to true - 'b' is greater than 'a'
stringGreaterThan: 'a'
value: 'b'

# Evaluates to true - 'a' is greater than '0'
stringGreaterThan: '0'
value: 'a'
```

```
# Evaluates to false - '0' is less than 'a'
stringGreaterThan: 'a'
value: '0'
```

stringGreaterThanSama dengan

Tes jika string yang `stringGreaterThanEquals` ditentukan untuk lebih besar dari atau sama dengan string yang ditentukan dalam `value` parameter. Misalnya:

```
# Evaluates to true - 'a' is equal to 'A'
stringGreaterThanEquals: 'A'
value: 'a'

# Evaluates to true - 'b' is greater than 'a'
stringGreaterThanEquals: 'a'
value: 'b'

# Evaluates to true - 'a' is greater than '0'
stringGreaterThanEquals: '0'
value: 'a'

# Evaluates to false - '0' is less than 'a'
stringGreaterThanEquals: 'a'
value: '0'
```

Pola Pertandingan

Menguji apakah string yang ditentukan dalam `value` parameter cocok dengan pola regex yang ditentukan untuk. `patternMatches` Perbandingan menggunakan paket regexp Golang, yang sesuai dengan sintaks. RE2 Untuk informasi selengkapnya tentang RE2 aturan, lihat repositori google/re2 di. GitHub

Contoh berikut menunjukkan kecocokan pola yang mengembalikan`true`:

```
patternMatches: '^[a-z]+$'
value: 'ThisIsValue'
```

Bandingkan angka

Operator perbandingan berikut bekerja dengan angka. Nilai yang diberikan untuk operator ini harus salah satu dari jenis berikut, sesuai dengan spesifikasi YAMAL. Support untuk perbandingan numerik menggunakan operator perbandingan paket besar golang, misalnya: func (*Float) Cmp.

- Bilangan Bulat
- Float (berdasarkan float64, yang mendukung angka dari -1.7e+308 hingga+1.7e+308)
- String yang cocok dengan pola regex berikut: `^[-+]?([0-9]+[.])?[0-9]+$`

Operator perbandingan angka

- numberEquals
- numberLessThan
- numberLessThanEquals
- numberGreaterThan
- numberGreaterThanEquals

NumberSama

Tes jika nomor yang `numberEquals` ditentukan untuk sama dengan angka yang ditentukan dalam `value` parameter. Semua contoh perbandingan berikut kembali: `true`

```
# Values provided as a positive number
numberEquals: 1
value: 1

# Comparison value provided as a string
numberEquals: '1'
value: 1

# Value provided as a string
numberEquals: 1
value: '1'

# Values provided as floats
numberEquals: 5.0
value: 5.0
```

```
# Values provided as a negative number
numberEquals: -1
value: -1
```

numberLessThan

Tes jika nomor yang `numberLessThan` ditentukan untuk kurang dari angka yang ditentukan dalam `value` parameter. Misalnya:

```
# Evaluates to true
numberLessThan: 2
value: 1

# Evaluates to true
numberLessThan: 2
value: 1.9

# Evaluates to false
numberLessThan: 2
value: '2'
```

numberLessThanSama dengan

Tes jika angka yang `numberLessThanEquals` ditentukan untuk kurang dari atau sama dengan angka yang ditentukan dalam `value` parameter. Misalnya:

```
# Evaluates to true
numberLessThanEquals: 2
value: 1

# Evaluates to true
numberLessThanEquals: 2
value: 1.9

# Evaluates to true
numberLessThanEquals: 2
value: '2'

# Evaluates to false
numberLessThanEquals: 2
value: 2.1
```

numberGreaterThan

Tes jika angka yang `numberGreaterThan` ditentukan untuk lebih besar dari angka yang ditentukan dalam `value` parameter. Misalnya:

```
# Evaluates to true
numberGreaterThan: 1
value: 2

# Evaluates to true
numberGreaterThan: 1
value: 1.1

# Evaluates to false
numberGreaterThan: 1
value: '1'
```

numberGreaterThanSama dengan

Tes jika angka yang `numberGreaterThanEquals` ditentukan untuk lebih besar dari atau sama dengan angka yang ditentukan dalam `value` parameter. Misalnya:

```
# Evaluates to true
numberGreaterThanEquals: 1
value: 2

# Evaluates to true
numberGreaterThanEquals: 1
value: 1.1

# Evaluates to true
numberGreaterThanEquals: 1
value: '1'

# Evaluates to false
numberGreaterThanEquals: 1
value: 0.8
```

Periksa file

Operator perbandingan berikut memeriksa hash file atau memeriksa apakah ada file atau folder.

Operator file dan folder

- binaryExists
- fileExists
- folderExists
- fileMD5Equals
- fileSHA1Equals
- fileSHA256Equals
- fileSHA512Equals

BinaryExists

Menguji apakah aplikasi tersedia di jalur saat ini. Misalnya:

```
binaryExists: 'foo'
```

> **Note**
>
> Pada sistem Linux dan macOS, untuk aplikasi bernama*foo*, ini bekerja sama dengan perintah bash berikut:type ***foo*** >/dev/null 2>&1, di mana $? == 0 menunjukkan perbandingan yang berhasil.
> Pada sistem Windows, untuk aplikasi bernama*foo*, ini berfungsi sama dengan PowerShell perintah di & C:\Windows\System32\where.exe /Q ***foo*** mana $LASTEXITCODE = 0 menunjukkan perbandingan yang berhasil.

FileExists

Menguji apakah file ada di jalur yang ditentukan. Anda dapat memberikan jalur absolut atau relatif. Jika lokasi yang Anda tentukan ada dan merupakan file, perbandingan akan dievaluasi. `true` Misalnya:

```
fileExists: '/path/to/file'
```

> **Note**
>
> Pada sistem Linux dan macOS, ini bekerja sama dengan perintah bash berikut:-d ***/path/to/file***, di mana $? == 0 menunjukkan perbandingan yang berhasil.
> Pada sistem Windows, ini bekerja sama dengan PowerShell perintahTest-Path -Path '***C:\path\to\file***' -PathType 'Leaf'.

FolderExists

Menguji apakah folder ada di jalur yang ditentukan. Anda dapat memberikan jalur absolut atau relatif. Jika lokasi yang Anda tentukan ada dan merupakan folder, perbandingan akan dievaluasi. `true` Misalnya:

```
folderExists: '/path/to/folder'
```

> **Note**
>
> Pada sistem Linux dan macOS, ini bekerja sama dengan perintah bash berikut:-d ***/path/to/folder***, di mana $? == 0 menunjukkan perbandingan yang berhasil.
> Pada sistem Windows, ini bekerja sama dengan PowerShell perintahTest-Path -Path '***C:\path\to\folder***' -PathType 'Container'.

Berkas MD5 Sama

Menguji apakah MD5 hash file sama dengan nilai tertentu. Misalnya:

```
fileMD5Equals: '<MD5Hash>'
path: '/path/to/file'
```

Berkas SHA1 Sama

Menguji apakah SHA1 hash file sama dengan nilai tertentu. Misalnya:

```
fileSHA1Equals: '<SHA1Hash>'
path: '/path/to/file'
```

Berkas SHA256 Sama

Menguji apakah SHA256 hash file sama dengan nilai tertentu. Misalnya:

```
fileSHA256Equals: '<SHA256Hash>'
path: '/path/to/file'
```

Berkas SHA512 Sama

Menguji apakah SHA512 hash file sama dengan nilai tertentu. Misalnya:

```
fileSHA512Equals: '<SHA512Hash>'
path: '/path/to/file'
```

Gunakan operator logis dalam dokumen AWSTOE komponen

Anda dapat menggunakan operator logika berikut untuk menambahkan atau memodifikasi ekspresi bersyarat dalam dokumen komponen Anda. AWSTOE mengevaluasi ekspresi bersyarat dalam urutan kondisi yang ditentukan. Untuk informasi selengkapnya tentang operator perbandingan untuk dokumen komponen, lihatGunakan operator perbandingan dalam dokumen AWSTOE komponen.

dan

Dengan `and` operator, Anda dapat mengevaluasi dua atau lebih perbandingan sebagai ekspresi tunggal. Ekspresi mengevaluasi `true` kapan semua kondisi dalam daftar benar. Jika tidak, ekspresi mengevaluasi. `false`

Contoh:

Contoh berikut melakukan dua perbandingan - string dan angka. Kedua perbandingan itu benar, sehingga ekspresi dievaluasi menjadi benar.

```
and:
  - stringEquals: 'test_string'
    value: 'test_string'
  - numberEquals: 1
    value: 1
```

Contoh berikut juga melakukan dua perbandingan. Perbandingan pertama salah, di mana evaluasi berhenti dan perbandingan kedua dilewati. Ekspresi mengevaluasi. `false`

```
and:
  - stringEquals: 'test_string'
    value: 'Hello world!'
  - numberEquals: 1
    value: 1
```

atau

Dengan `or` operator, Anda dapat mengevaluasi dua atau lebih perbandingan sebagai ekspresi tunggal. Ekspresi mengevaluasi `true` kapan salah satu perbandingan yang ditentukan benar. Jika tidak ada perbandingan yang ditentukan yang dievaluasi`true`, ekspresi mengevaluasi. `false`

Contoh:

Contoh berikut melakukan dua perbandingan - string dan angka. Perbandingan pertama benar, sehingga ekspresi dievaluasi `true` dan perbandingan kedua dilewati.

```
or:
  - stringEquals: 'test_string'
    value: 'test_string'
  - numberEquals: 1
    value: 3
```

Contoh berikut juga melakukan dua perbandingan. Perbandingan pertama salah, dan evaluasi berlanjut. Perbandingan kedua benar, sehingga ekspresi mengevaluasi. `true`

```
or:
  - stringEquals: 'test_string'
    value: 'Hello world!'
  - numberEquals: 1
    value: 1
```

Dalam contoh terakhir, kedua perbandingan itu salah, sehingga ekspresi dievaluasi. `false`

```
or:
  - stringEquals: 'test_string'
    value: 'Hello world!'
  - numberEquals: 1
    value: 3
```

tidak

Dengan `not` operator, Anda dapat meniadakan satu perbandingan. Ekspresi mengevaluasi `true` apakah perbandingannya salah. Jika perbandingannya benar, maka ekspresi dievaluasi. `false`

Contoh:

Contoh berikut melakukan perbandingan string. Perbandingannya salah, sehingga ekspresi mengevaluasi`true`.

```
not:
  - stringEquals: 'test_string'
    value: 'Hello world!'
```

Contoh berikut juga melakukan perbandingan string. Perbandingannya benar, sehingga ekspresi mengevaluasi`false`.

```
not:
  - stringEquals: 'test_string'
    value: 'test_string'
```

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
  - `forEach`loop dengan daftar masukan - Iterasi atas kumpulan string yang terbatas.
  - `forEach`loop 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

AWSTOE Mengatasi 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 input
- forEachloop dengan daftar terbatas

`for`lingkaran

`for`Loop 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.

AWSTOE memverifikasi`start`,`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	Wajib	Default
`name`	Nama unik dari loop. Itu harus unik dibanding kan dengan nama loop lain dalam fase yang sama.	String	Tidak	""
`start`	Nilai awal iterasi. Tidak menerima ekspresi rantai.	Bilangan Bulat	Ya	T/A
`end`	Nilai akhir iterasi. Tidak menerima ekspresi rantai.	Bilangan Bulat	Ya	T/A
`updateBy`	Perbedaan dimana nilai iterasi diperbaru i melalui penambahan. Itu harus nilai negatif atau positif bukan nol. Tidak menerima ekspresi rantai.	Bilangan 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
```

`forEach`loop dengan daftar input

`forEach`Loop iterasi pada daftar eksplisit nilai, yang dapat berupa string dan ekspresi dirantai.

`forEach`loop dengan skema daftar masukan

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

`forEach`loop dengan masukan daftar masukan

Bidang	Deskripsi	Tipe	Wajib	Default
`name`	Nama unik dari loop. Itu harus unik dibanding kan dengan nama loop lain dalam fase yang sama.	String	Tidak	""
Daftar string loop `forEach`	Daftar string untuk iterasi. Menerima ekspresi berantai sebagai string dalam daftar. Ekspresi berantai harus diapit oleh tanda kutip ganda agar kompiler YAMAL dapat menafsirk annya dengan benar.	Daftar string	Ya	T/A

`forEach`loop 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
```

`forEach`loop 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 }}"
```

`forEach`loop 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 }}"
```

`forEach`loop dengan daftar terbatas

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.

`forEach`loop dengan skema daftar yang dibatasi

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

`forEach`loop dengan input daftar yang dibatasi

Bidang	Deskripsi	Tipe	Wajib	Default
`name`	Nama unik yang diberikan ke loop. Ini harus unik jika dibandingkan dengan nama loop lain dalam fase yang sama.	String	Tidak	""
`list`	String yang terdiri dari string konstitue n bergabung dengan karakter pembatas umum. Juga menerima ekspresi	Tali	Ya	T/A

Bidang	Deskripsi	Tipe	Wajib	Default
	dirantai. Dalam kasus ekspresi berantai, pastikan bahwa ekspresi tersebut diapit oleh tanda kutip ganda untuk interpret asi yang benar oleh kompiler YAMAL.			

Bidang	Deskripsi	Tipe	Wajib	Default
`delimiter`	Karakter yang digunakan untuk memisahkan string dalam blok. Default adalah karakter koma. Hanya satu karakter pembatas yang diizinkan dari daftar yang diberikan: • Titik: `"."` • Koma: `","` • Titik koma: `";"` • Usus besar: `":"` • Baris baru: `"\n"` • Tab: `"\t"` • Ruang: `" "` • Tanda hubung: `"-"` • Garis bawahi: `"_"`	String	Tidak	Koma: `","`

Bidang	Deskripsi	Tipe	Wajib	Default
	Ekspresi rantai tidak dapat digunakan.			

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

`forEach`loop dengan daftar yang dibatasi contoh 1

Contoh ini menggunakan pola ekspresi rantai berikut untuk merujuk ke output langkah lain:`<phase_name>.<step_name>.[inputs | outputs].<var_name>`.

```
- name: "RunMSIs"
  action: "ExecuteBinary"
  loop:
    forEach:
      list: "{{ build.GetAllMSIPathsForInstallation.outputs.stdout }}"
      delimiter: "\n"
  inputs:
    commands:
      path: "{{ loop.value }}"
```

`forEach`loop dengan contoh daftar terbatas 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 setel 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 ke`Abort`, 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.

    AWSTOE menetapkan kode status untuk fase induk dan dokumen ke`Failed`.

    > **Note**
    >
    > Tidak ada langkah lebih lanjut yang berjalan setelah langkah gagal.

  - Jika onFailure disetel ke`Continue`, 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.

    AWSTOE menetapkan kode status untuk fase induk dan dokumen ke`Failed`.
  - Jika onFailure disetel ke`Ignore`, AWSTOE keluar dari loop dan mencoba ulang langkah sesuai dengan kebijakan coba lagi. Setelah jumlah maksimum percobaan ulang, AWSTOE tandai langkah saat ini sebagai`IgnoredFailure`, dan lanjutkan untuk menjalankan langkah berikutnya.

    AWSTOE menetapkan kode status untuk fase induk dan dokumen ke`SuccessWithIgnoredFailure`.

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

- MaxAttracts — 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,, `ExecuteBash` `ExecutePowerShell`, 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, menggunakan modul AWSTOE tindakan untuk membantu mengonfigurasi EC2 instance 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.

Komponen ditulis dengan dokumen YAMM plaintext. Untuk informasi selengkapnya tentang sintaks dokumen, lihatGunakan kerangka dokumen AWSTOE komponen untuk komponen kustom.

> **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.

Referensi silang berikut mengkategorikan modul tindakan berdasarkan jenis tindakan yang mereka lakukan.

Eksekusi umum

- Menegaskan (Linux, Windows, macOS)
- ExecuteBash (Linux, macOS)
- ExecuteBinary (Linux, Windows, macOS)
- ExecuteDocument (Linux, Windows, macOS)
- ExecutePowerShell (Jendela)

Unduh dan unggah file

- S3Unduh (Linux, Windows, macOS)
- S3Unggah (Linux, Windows, macOS)
- WebDownload (Linux, Windows, macOS)

Operasi sistem file

- AppendFile (Linux, Windows, macOS)
- CopyFile (Linux, Windows, macOS)
- CopyFolder (Linux, Windows, macOS)
- CreateFile (Linux, Windows, macOS)
- CreateFolder (Linux, Windows, macOS)
- CreateSymlink (Linux, Windows, macOS)
- DeleteFile (Linux, Windows, macOS)
- DeleteFolder (Linux, Windows, macOS)
- ListFiles (Linux, Windows, macOS)
- MoveFile (Linux, Windows, macOS)
- MoveFolder (Linux, Windows, macOS)
- ReadFile (Linux, Windows, macOS)
- SetFileEncoding (Linux, Windows, macOS)
- SetFileOwner (Linux, Windows, macOS)
- SetFolderOwner (Linux, Windows, macOS)
- SetFilePermissions (Linux, Windows, macOS)
- SetFolderPermissions (Linux, Windows, macOS)

Tindakan instalasi perangkat lunak

- InstallMSI (Windows)
- UninstallMSI (Windows)

Tindakan sistem

- Reboot (Linux, Windows)
- SetRegistry (Jendela)
- UpdateOS (Linux, Windows)

Modul eksekusi umum

Bagian berikut berisi rincian untuk modul tindakan yang menjalankan perintah dan mengontrol alur kerja eksekusi.

Modul tindakan eksekusi umum

- Menegaskan (Linux, Windows, macOS)
- ExecuteBash (Linux, macOS)
- ExecuteBinary (Linux, Windows, macOS)
- ExecuteDocument (Linux, Windows, macOS)
- ExecutePowerShell (Jendela)

Menegaskan (Linux, Windows, macOS)

Modul tindakan Assert melakukan perbandingan nilai menggunakan Operator perbandingan atau Operator logis sebagai input. Hasil ekspresi operator (benar atau salah) menunjukkan status keberhasilan atau kegagalan keseluruhan untuk langkah tersebut.

Jika perbandingan atau ekspresi operator logis mengevaluasi`true`, langkah ditandai sebagai`Success`. Jika tidak, langkahnya ditandai sebagai`Failed`. Jika langkah gagal, `onFailure` parameter menentukan hasil dari langkah tersebut.

Input

Nama Kunci	Deskripsi	Tipe	Wajib
`input`	Berisi perbandingan tunggal atau operator logis. Catatan,	Ini adalah variabel, tergantung pada operator	Ya

Nama Kunci	Deskripsi	Tipe	Wajib
	operator logika dapat berisi lebih dari satu operator perbandingan.		

Contoh masukan: Perbandingan sederhana menggunakan operator **`stringEquals`** perbandingan

Contoh ini mengevaluasi. `true`

```
- name: StringComparison
  action: Assert
  inputs:
    stringEquals: '2.1.1'
    value: '{{ validate.ApplicationVersion.outputs.stdout }}'
```

Contoh masukan: Perbandingan Regex menggunakan operator perbandingan **`patternMatches`**

Contoh-contoh ini semua mengevaluasi`true`.

```
- name: Letters only
  action: Assert
  inputs:
    patternMatches: '^[a-zA-Z]+$'
    value: 'ThisIsOnlyLetters'

- name: Letters and spaces only
  action: Assert
  inputs:
    patternMatches: '^[a-zA-Z\s]+$'
    value: 'This text contains spaces'

- name: Numbers only
  action: Assert
  inputs:
    patternMatches: '^[0-9]+$'
    value: '1234567890'
```

Contoh masukan: Perbandingan bersarang dengan operator logis dan variabel berantai

Contoh berikut menunjukkan perbandingan bersarang dengan operator logis yang menggunakan perbandingan dengan variabel berantai. `Assert`Mengevaluasi `true` apakah salah satu dari berikut ini benar:

- Yang `ApplicationVersion` lebih besar dari `2.0` dan `CPUArchitecture` sama. `arm64`
- Yang `CPUArchitecture` sama. `x86_64`

```
- name: NestedComparisons
  action: Assert
  inputs:
    or: # <- first level deep
      - and: # <- second level deep
          - numberGreaterThan: 2.0 # <- third level deep
            value: '{{ validate.ApplicationVersion.outputs.stdout }}'
          - stringEquals: 'arm64'
            value: '{{ validate.CPUArchitecture.outputs.stdout }}'
      - stringEquals: 'x86_64'
        value: '{{ validate.CPUArchitecture.outputs.stdout }}'
```

Output:

Output dari sebuah `Assert` adalah keberhasilan atau kegagalan langkah.

ExecuteBash (Linux, macOS)

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 arus`executionstate`, 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

Nama Kunci	Deskripsi	Tipe	Wajib
`commands`	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 (Linux, Windows, macOS)

Modul ExecuteBinarytindakan 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 arus`executionstate`, 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

Nama Kunci	Deskripsi	Tipe	Wajib
`path`	Path ke file biner untuk eksekusi.	Tali	Ya
`arguments`	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 (Linux, Windows, macOS)

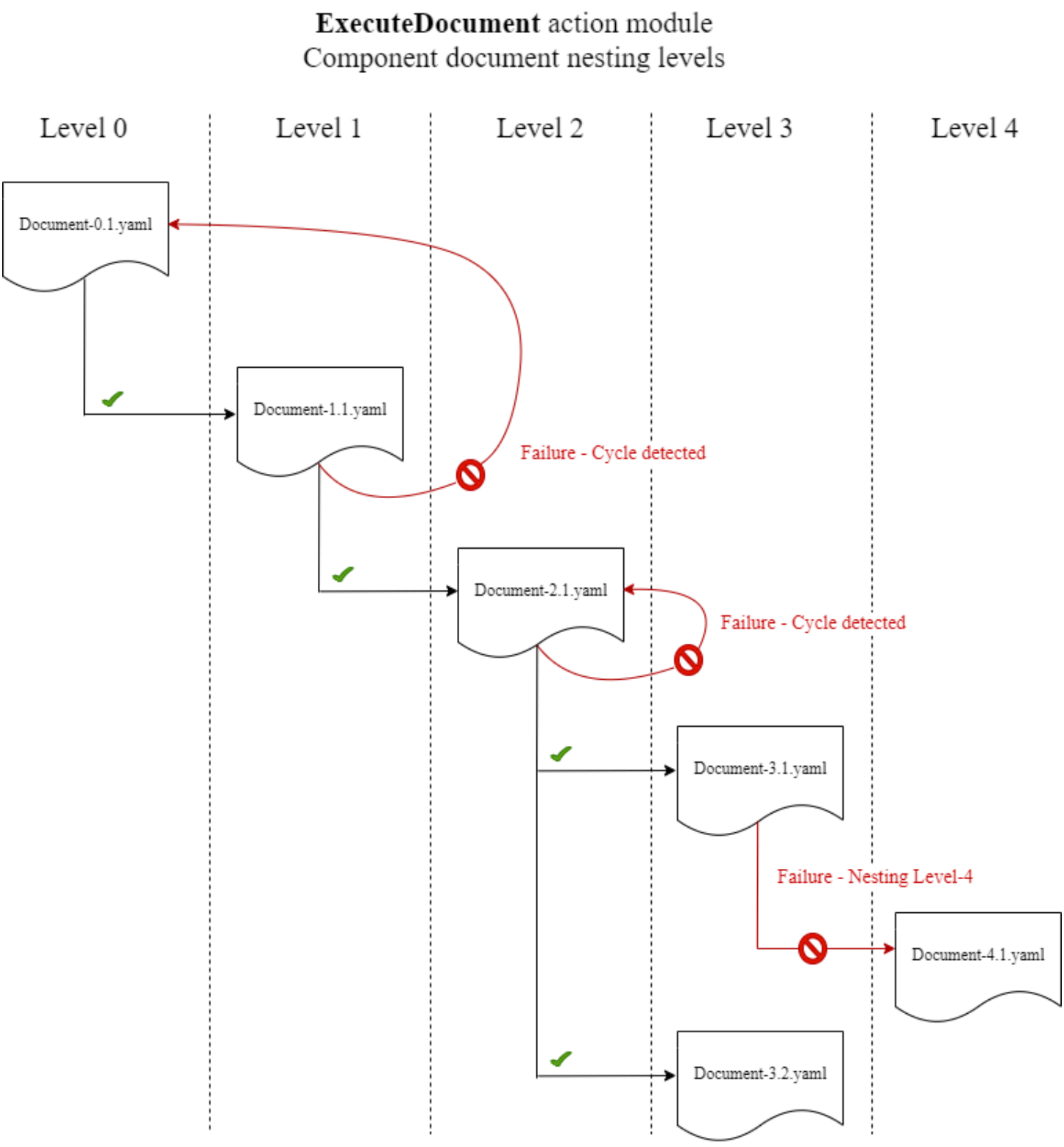
Modul ExecuteDocumenttindakan menambahkan dukungan untuk dokumen komponen bersarang, menjalankan beberapa dokumen komponen dari satu dokumen. AWSTOE memvalidasi dokumen yang diteruskan dalam parameter input pada waktu berjalan.

Pembatasan

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

  > **Note**
  >
  > Anda dapat membiarkan nilai-nilai ini kosong, dan AWSTOE menggunakan 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 AWSTOE mendeteksi eksekusi siklik, itu menghentikan eksekusi dan mencatat kegagalan.



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

Masukan

Nama Kunci	Deskripsi	Tipe	Wajib
`document`	Jalur dokumen komponen. Pilihan yang valid meliputi: • Jalur file lokal • S3 URIs • EC2 Versi pembuatan komponen Image Builder ARNs	Tali	Ya
`document-s3-bucket-owner`	ID akun pemilik bucket S3 untuk bucket S3 tempat dokumen komponen disimpan. (Disarankan jika Anda menggunakan S3 URIs dalam 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

Nama Kunci	Deskripsi	Tipe	Wajib
`parameters`	Parameter masukan yang diteruskan ke dokumen komponen saat runtime sebagai pasangan nilai kunci.	Daftar Peta Parameter	Tidak

Masukan peta parameter

Nama Kunci	Deskripsi	Tipe	Wajib
`name`	Nama parameter input untuk diteruskan ke dokumen komponen yang sedang dijalankan modul ExecuteDocumenttindakan.	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: EC2 Image Builder komponen 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

AWSTOE membuat 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 ExecuteDocumenttindakan, Anda dapat menemukan ringkasan runtime singkat di `outputs` lapangan, dan detail tentang fase, langkah, dan dokumen yang dijalankan di`detailedOutput`.

```
{
 \"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 ExecuteDocumenttindakan 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": [
                                        {
                                            "name": "",
                                            "filePath": "Sample-1.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": "ExecuteBashStep",
                                                            "status": "success",
                                                            "failureMessage": "",
                                                            "timeoutSeconds": 7200,
                                                            "onFailure": "Abort",
                                                            "maxAttempts": 1,
                                                            "action": "ExecuteBash",
                                                            "startTime":
 "2021-08-26T17:20:31-07:00",
                                                            "endTime":
 "2021-08-26T17:20:31-07:00",
                                                            "inputs": "[{\"commands\":
[\"echo \\\"Hello World!\\\"\"]}]",
                                                            "outputs": "[{\"stdout\":
\"Hello World!\"}]",
                                                            "loop": null,
```

```
                                                    "detailedOutput": null
                                                }]
                                            }]
                                        }]
                                    }]
                                }]

                            }]
                        }]
}
```

ExecutePowerShell (Jendela)

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

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

ExecutePowerShellModul menangani sistem restart jika perintah shell keluar dengan kode keluar dari. `3010` 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 arus`executionstate`, mengonfigurasi pemicu 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

Nama Kunci	Deskripsi	Tipe	Wajib
`commands`	Berisi daftar instruksi atau perintah untuk dijalankan sesuai	Daftar String	Ya. Harus menentuka n `commands`

Nama Kunci	Deskripsi	Tipe	Wajib
	PowerShell sintaks. YAML multi-baris diizinkan.		ataufile, tidak keduanya.
file	Berisi path ke file PowerShell script. PowerShell akan berjalan melawan file ini menggunakan argumen baris -file perintah. Jalur harus menunjuk ke .ps1 file.	String	Ya. Harus menentukan commands ataufile, 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
`stdout`	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 detail untuk modul tindakan yang mengunggah atau mengunduh file.

Unduh dan unggah modul tindakan

- S3Unduh (Linux, Windows, macOS)
- S3Unggah (Linux, Windows, macOS)
- WebDownload (Linux, Windows, macOS)

S3Unduh (Linux, Windows, macOS)

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.

`source`Lokasi 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/*`*.

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` melawan bucket/object (misalnya,`arn:aws:s3:::BucketName/*`).
- Beberapa file: `s3:ListBucket` melawan bucket/object (misalnya,`arn:aws:s3:::BucketName`) dan `s3:GetObject` melawan bucket/object (misalnya,`arn:aws:s3:::BucketName/*`).

Input

Kunci	Deskripsi	Tipe	Wajib	Default
`source`		Tali	Ya	N/A

Kunci	Deskripsi	Tipe	Wajib	Default
	Bucket Amazon S3 yang merupakan sumber unduhan Anda. 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.			

Kunci	Deskripsi	Tipe	Wajib	Default
`destination`	Jalur lokal tempat objek Amazon S3 diunduh. Untuk mengunduh satu file, Anda harus menentukan nama file sebagai bagian dari jalur. Misalnya, */myfolder/package.zip* .	Tali	Ya	N/A
`expectedBucketOwner`	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	N/A

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

Kunci	Deskripsi	Tipe	Wajib	Default
	false. 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://amzn-s3-demo-source-bucket/path/to/package.zip
      destination: C:\myfolder\package.zip
      expectedBucketOwner: 123456789022
      overwrite: false
    - source: s3://amzn-s3-demo-source-bucket/path/to/package.zip
      destination: C:\myfolder\package.zip
      expectedBucketOwner: 123456789022
      overwrite: true
    - source: s3://amzn-s3-demo-source-bucket/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://amzn-s3-demo-source-bucket/path/to/*
        destination: C:\myfolder\
        expectedBucketOwner: 123456789022
        overwrite: false
      - source: s3://amzn-s3-demo-source-bucket/path/to/*
        destination: C:\myfolder\
        expectedBucketOwner: 123456789022
        overwrite: true
      - source: s3://amzn-s3-demo-source-bucket/path/to/*
        destination: C:\myfolder\
        expectedBucketOwner: 123456789022
```

Output

Tidak ada.

S3Unggah (Linux, Windows, macOS)

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

Kunci	Deskripsi	Tipe	Wajib	Default
`source`	Jalur lokal tempat sumber files/folders berasal. `source`Mendukun wildcard asterisk ()*.	Tali	Ya	N/A
`destination`	Jalur untuk bucket Amazon S3 tujuan tempat file sumber/folder diunggah.	Tali	Ya	N/A
`recurse`	Saat disetel ke`true`, lakukan S3Upload secara rekursif.	String	Tidak	`false`
`expectedB ucketOwner`	ID akun pemilik yang diharapka n untuk bucket Amazon S3 yang ditentuka n di jalur tujuan. Kami menyarankan Anda memverifi kasi kepemilik an bucket Amazon S3 yang	String	Tidak	N/A

Kunci	Deskripsi	Tipe	Wajib	Default
	ditentukan di tujuan.			

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://amzn-s3-demo-destination-bucket/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://amzn-s3-demo-destination-bucket/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://amzn-s3-demo-destination-bucket/path/to/
        recurse: true
        expectedBucketOwner: 123456789022
```

Output

Tidak ada.

WebDownload (Linux, Windows, macOS)

Modul WebDownloadtindakan memungkinkan Anda mengunduh file dan sumber daya dari lokasi terpencil melalui HTTP/HTTPS protokol (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 dokumen`steps`, yang terkait dengan kegagalan modul tindakan.

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

Input

Nama Kunci	Deskripsi	Tipe	Wajib	Default
`source`	HTTP/HTTPS URL yang valid (HTTPS direkomendasikan), yang mengikuti standar RFC 3986. Ekspresi rantai diizinkan.	Tali	Ya	N/A
`destination`	Sebuah file absolut atau	Tali	Ya	N/A

Nama Kunci	Deskripsi	Tipe	Wajib	Default
	relatif atau jalur folder pada sistem lokal. Jalur folder harus diakhiri dengan/. Jika mereka tidak berakhir dengan/, mereka akan diperlakukan sebagai jalur file. Modul ini membuat file atau folder yang diperlukan untuk unduhan yang berhasil. Ekspresi rantai diizinkan.			

Nama Kunci	Deskripsi	Tipe	Wajib	Default
`overwrite`	Saat diaktifkan, timpa file yang ada di sistem lokal dengan file atau sumber daya yang diunduh. Saat tidak diaktifka n, 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	`true`

Nama Kunci	Deskripsi	Tipe	Wajib	Default
`checksum`	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.	String	Tidak	N/A
`algorithm`	Algoritma yang digunakan untuk menghitung checksum. Pilihannya adalahMD5, SHA1, SHA256, dan SHA512. Agar verifikasi file diaktifkan, checksum dan algoritme harus disediakan. Ekspresi rantai diizinkan.	String	Tidak	N/A

Nama Kunci	Deskripsi	Tipe	Wajib	Default
`ignoreCertificateErrors`	Validasi sertifikat SSL diabaikan saat diaktifkan.	Boolean	Tidak	`false`

Output

Nama Kunci	Deskripsi	Tipe
`destination`	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 operasi sistem file.

Modul tindakan operasi sistem file

- AppendFile (Linux, Windows, macOS)
- CopyFile (Linux, Windows, macOS)
- CopyFolder (Linux, Windows, macOS)
- CreateFile (Linux, Windows, macOS)
- CreateFolder (Linux, Windows, macOS)
- CreateSymlink (Linux, Windows, macOS)
- DeleteFile (Linux, Windows, macOS)
- DeleteFolder (Linux, Windows, macOS)
- ListFiles (Linux, Windows, macOS)
- MoveFile (Linux, Windows, macOS)
- MoveFolder (Linux, Windows, macOS)
- ReadFile (Linux, Windows, macOS)

- SetFileEncoding (Linux, Windows, macOS)
- SetFileOwner (Linux, Windows, macOS)
- SetFolderOwner (Linux, Windows, macOS)
- SetFilePermissions (Linux, Windows, macOS)
- SetFolderPermissions (Linux, Windows, macOS)

AppendFile (Linux, Windows, macOS)

Modul AppendFiletindakan 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

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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
					`LE utf16-BE`,`utf-16-BE ,utf32`,`utf 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 (Linux, Windows, macOS)

Modul CopyFiletindakan 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`source`	Jalur file sumber.	Tali	Ya	N/A	N/A	Ya
`destination`	Jalur file tujuan.	Tali	Ya	N/A	N/A	Ya
`overwrite`	Ketika diatur ke false, file tujuan	Boolean	Tidak	`true`	N/A	Ya

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
	tidak akan diganti ketika sudah ada file di lokasi yang ditentuka n 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 (Linux, Windows, macOS)

Modul CopyFoldertindakan 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`source`	Jalur folder sumber.	Tali	Ya	N/A	N/A	Ya
`destination`	Jalur folder tujuan.	Tali	Ya	N/A	N/A	Ya
`overwrite`	Ketika diatur ke false, folder tujuan tidak akan diganti ketika sudah ada folder di lokasi yang ditentukan dengan nama yang ditentukan.	Boolean	Tidak	`true`	N/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 (Linux, Windows, macOS)

Modul CreateFiletindakan 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`path`	Jalur file.	Tali	Ya	N/A	N/A	Ya
`content`	Isi teks file.	String	Tidak	N/A	N/A	Ya
`encoding`	Standar pengkodean.	String	Tidak	`utf8`	`utf8`,`utf-8`, `utf-16-LE`, `utf16-BE`,`utf-16-BE`,`utf32`,`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	N/A	Tidak didukung di Windows.

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

Contoh masukan: buat file tanpa menimpa (Linux)

```
- name: CreatingFileWithoutOverwriteLinux
  action: CreateFile
  inputs:
    - path: /home/UserName/Sample.txt
      content: The text content of the sample file.
      overwrite: false
```

Contoh masukan: buat file tanpa menimpa (Windows)

```
- name: CreatingFileWithoutOverwriteWindows
  action: CreateFile
  inputs:
    - path: C:\Temp\Sample.txt
      content: The text content of the sample file.
      overwrite: false
```

Contoh masukan: buat file dengan properti file

```
- name: CreatingFileWithFileProperties
  action: CreateFile
  inputs:
    - path: SampleFolder/Sample.txt
      content: The text content of the sample file.
      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, lihat Ganti skrip pembersihan Linux

Output

Tidak ada.

CreateFolder (Linux, Windows, macOS)

Modul CreateFoldertindakan 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`path`	Jalur folder.	Tali	Ya	N/A	N/A	Ya
`owner`	Nama pengguna atau ID.	String	Tidak	Pengguna saat ini.	N/A	Tidak didukung di Windows.

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`group`	Nama grup atau ID.	String	Tidak	Kelompok pengguna saat ini.	N/A	Tidak didukung di Windows.
`permissions`	Izin folder.	String	Tidak	`0777`	N/A	Tidak didukung di Windows.
`overwrite`	Jika nama file yang ditentukan sudah ada, setel nilai ini untuk `false` mencegah file terpotong atau ditimpa secara default.	Boolean	Tidak	`true`	N/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/SampleFoler/
      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 (Linux, Windows, macOS)

Modul CreateSymlinktindakan 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`path`	Jalur file.	Tali	Ya	N/A	N/A	Tidak didukung di Windows.
`target`	Jalur file target yang ditunjuk oleh tautan simbolis.	Tali	Ya	N/A	N/A	Tidak didukung di Windows.
`force`	Memaksa pembuatan tautan ketika tautan dengan nama yang sama sudah ada.	Boolean	Tidak	`false`	N/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 (Linux, Windows, macOS)

Modul DeleteFiletindakan 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`path`	Jalur file.	Tali	Ya	N/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 (Linux, Windows, macOS)

Modul DeleteFoldertindakan menghapus folder.

Jika folder tidak kosong, Anda harus mengatur `force` opsi `true` untuk menghapus folder dan isinya. Jika Anda tidak mengatur `force` opsi ke`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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`path`	Jalur folder.	Tali	Ya	N/A	N/A	Ya
`force`	Menghapus folder apakah folder kosong atau tidak.	Boolean	Tidak	`false`	N/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 (Linux, Windows, macOS)

Modul ListFilestindakan mencantumkan file dalam folder tertentu. Ketika opsi rekursif diatur `ketrue`, 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`path`	Jalur folder.	Tali	Ya	N/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`	N/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

Nama Kunci	Deskripsi	Tipe
`files`	Daftar file.	String

Contoh keluaran

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

```
}
```

MoveFile (Linux, Windows, macOS)

Modul MoveFiletindakan 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`on 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`source`	Jalur file sumber.	Tali	Ya	N/A	N/A	Ya

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`destination`	Jalur file tujuan.	Tali	Ya	N/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`	N/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 (Linux, Windows, macOS)

Modul MoveFoldertindakan 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`source`	Jalur folder sumber.	Tali	Ya	N/A	N/A	Ya
`destination`	Jalur folder tujuan.	Tali	Ya	N/A	N/A	Ya
`overwrite`	Ketika diatur ke false, folder tujuan tidak akan diganti ketika sudah ada folder di lokasi yang ditentukan dengan nama yang ditentukan.	Boolean	Tidak	`true`	N/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 (Linux, Windows, macOS)

Modul ReadFiletindakan 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 simbolik, 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 pengaturan 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform	
`path`	Jalur file.	Tali	Ya	N/A	N/A	Ya	
`encoding`	Standar pengkodean.	String	Tidak	`utf8`	`utf8`,`utf-8`, `utf-16-LE`, `utf16-BE`,`utf-16-BE`,`utf32`,`utf-32-LE`,`utf32-BE`, dan `utf-32-BE`. Nilai opsi pengkodean tidak peka huruf besar/kecil.	Ya	6,utf1
`printFileContent`	Mencetak konten	Boolean	Tidak	false	N/A	Ya.	

Nama Kunci	Deskripsi	Tipe	Wajib	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 (Linux, Windows, macOS)

Modul SetFileEncodingtindakan 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform	
`path`	Jalur file.	Tali	Ya	N/A	N/A	Ya	
`encoding`	Standar pengkodean.	String	Tidak	`utf8`	`utf8`,`utf-8`, `utf-16-LE`, `utf-16-LE` `utf16-BE`,`utf-16-`	Ya	`6`,`utf1`

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
					BE ,utf32,utfLE,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 (Linux, Windows, macOS)

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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`path`	Jalur file.	Tali	Ya	N/A	N/A	Tidak didukung di Windows.
`owner`	Nama pengguna.	string	Ya	N/A	N/A	Tidak didukung di Windows.
`group`	Nama grup pengguna.	String	Tidak	Nama grup tempat pengguna berada.	N/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 (Linux, Windows, macOS)

Modul SetFolderOwnertindakan secara rekursif memodifikasi properti `owner` dan `group` pemilik folder yang ada. Secara default, modul dapat mengubah 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`path`	Jalur folder.	Tali	Ya	N/A	N/A	Tidak didukung di Windows.
`owner`	Nama pengguna.	string	Ya	N/A	N/A	Tidak didukung di Windows.

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`group`	Nama grup pengguna.	String	Tidak	Nama grup tempat pengguna berada.	N/A	Tidak didukung di Windows.
`recursive`	Mengganti perilaku default memodifikasi kepemilikan untuk semua konten folder saat disetel ke. `false`	Boolean	Tidak	`true`	N/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: atur 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 (Linux, Windows, macOS)

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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`path`	Jalur file.	Tali	Ya	N/A	N/A	Tidak didukung di Windows.

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`permissions`	Izin file.	Tali	Ya	N/A	N/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 (Linux, Windows, macOS)

Modul SetFolderPermissionstindakan 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima	Didukung di semua platform
`path`	Jalur folder.	Tali	Ya	N/A	N/A	Tidak didukung di Windows.
`permissions`	Izin folder.	Tali	Ya	N/A	N/A	Tidak didukung di Windows.
`recursive`	Mengganti perilaku default memodifikasi izin untuk semua konten folder saat disetel ke. `false`	Boolean	Tidak	`true`	N/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 berikut menjelaskan modul tindakan yang menginstal atau menghapus 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

- InstallMSI (Windows)
- UninstallMSI (Windows)

InstallMSI (Windows)

Modul `InstallMSI` 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.

AWSTOE menghasilkan 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`path`	Tentukan lokasi file MSI menggunakan salah satu dari berikut ini: •	Tali	Ya	N/A	N/A

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
	Jalur file lokal. Jalannya bisa absolut atau relatif • URI objek S3 yang valid. • HTTP/ HTTPS URL web yang valid (disarankan HTTPS) yang mengikuti standar RFC 3986. Ekspresi rantai diperbolehkan.				

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`reboot`	Konfigurasikan perilaku reboot sistem yang mengikuti keberhasilan menjalankan modul tindakan. Pengaturan: • `Force`— Memulai reboot sistem setelah msiexec perintah berjalan dengan sukses. • `Allow`— Memulai reboot sistem jika msiexec perintah mengembalikan kode keluar yang	String	Tidak	`Allow`	`Allow`, `Force`, `Skip`

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
	menunjukkan reboot diperlukan. • `Skip`— Log pesan informasi ke `console.log` file yang menunjukkan bahwa reboot dilewati. Opsi ini mencegah reboot, bahkan jika msiexec perintah mengembalikan kode keluar yang menunjukkan reboot diperlukan.				

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`logOptions`	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. Untuk informasi selengkapnya tentang opsi log untuk	String	Tidak	`*VX`	`i,w,e,a,r,u,c,m,o,p,v,x,+,!,*`

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
	MSI, lihat Opsi 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	N/A

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`properties`	Pasangan nilai kunci properti logging MSI, misalnya: `TARGETDIR: "C:\target\location"` Catatan: Modifikasi properti berikut tidak diperbolehkan: • `REBOOT="ReallySupress"` • `REINSTALLMODE="ecmus"` • `REINSTALL="ALL"`	Peta [String] String	Tidak	N/A	N/A

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`ignoreAuthenticodeSignatureErrors`	Tandai untuk mengabaikan kesalahan validasi tanda tangan authenticode untuk penginstal yang ditentukan di jalur. Get-AuthenticodeSignaturePerintah ini digunakan untuk memvalidasi installer. Pengaturan: • `true`— Kesalahan validasi diabaikan dan penginstal berjalan. • `false`— Kesalahan validasi	Boolean	Tidak	`false`	`true`, `false`

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
	tidak diabaikan. Penginstal berjalan hanya ketika validasi berhasil. Ini adalah perilaku default.				

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`allowUnsignedInstaller`	Tandai untuk memungkinkan menjalankan penginstal yang tidak ditandatangani yang ditentukan di jalur. Get-AuthenticodeSignaturePerintah ini digunakan untuk memvalidasi installer. Pengaturan: • `true`— Mengabaikan `NotSigned` status yang dikembalikan oleh Get-AuthenticodeSignature perintah	Boolean	Tidak	`false`	`true`, `false`

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
	dan menjalankan installer. • `false`— 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 (Windows)

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.

AWSTOE 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

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`path`	Tentukan lokasi file MSI menggunakan salah satu dari berikut ini: • Jalur file lokal. Jalannya bisa absolut atau relatif.	Tali	Ya	N/A	N/A

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
	• URI objek S3 yang valid. • HTTP/HTTPS URL web yang valid (disarankan HTTPS) yang mengikuti standar RFC 3986. Ekspresi rantai diperbolehkan.				

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`reboot`	Mengkonfigurasi perilaku reboot sistem yang mengikuti keberhasilan menjalankan modul tindakan. Pengaturan: • `Force`— Memulai reboot sistem setelah msiexec perintah berjalan dengan sukses. • `Allow`— Memulai reboot sistem jika msiexec perintah mengembalikan kode	String	Tidak	`Allow`	`Allow`, `Force`, `Skip`

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
	keluar yang menunjukkan reboot diperlukan. • `Skip`— Log pesan informasi ke `console.log` file yang menunjukkan bahwa reboot dilewati. Opsi ini mencegah reboot, bahkan jika msiexec perintah mengembalikan kode keluar yang menunjukkan reboot diperlukan.				

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`logOptions`	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. Untuk informasi selengkapnya tentang opsi log untuk	String	Tidak	`*VX`	`i,w,e,a,r,u,c,m,o,p,v,x,+,!,*`

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
	MSI, lihat Opsi 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	N/A

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`properties`	Pasangan nilai kunci properti logging MSI, misalnya: `TARGETDIR: "C:\target\location"` Catatan: Modifikasi properti berikut tidak diperbolehkan: • `REBOOT="ReallySupress"` • `REINSTALLMODE="ecmus"` • `REINSTALL="ALL"`	Peta [String] String	Tidak	N/A	N/A

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`ignoreAuthenticodeSignatureErrors`	Tandai untuk mengabaikan kesalahan validasi tanda tangan authenticode untuk penginstal yang ditentukan di jalur. Get-AuthenticodeSignaturePerintah ini digunakan untuk memvalidasi installer. Pengaturan: • `true`— Kesalahan validasi diabaikan dan penginstal berjalan. • `false`— Kesalahan validasi	Boolean	Tidak	`false`	`true`, `false`

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
	tidak diabaikan. Penginstal berjalan hanya ketika validasi berhasil. Ini adalah perilaku default.				

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
`allowUnsignedInstaller`	Tandai untuk memungkinkan menjalankan penginstal yang tidak ditandatangani yang ditentukan di jalur. Get-AuthenticodeSignaturePerintah ini digunakan untuk memvalidasi installer. Pengaturan: • `true`— Mengabaikan `NotSigned` status yang dikembalikan oleh Get-AuthenticodeSignature perintah	Boolean	Tidak	`false`	`true`, `false`

Nama Kunci	Deskripsi	Tipe	Wajib	Nilai default	Nilai yang dapat diterima
	dan menjalankan installer. • `false`— 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:\msilogs\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 tindakan sistem atau memperbarui pengaturan sistem.

Modul tindakan sistem

- Reboot (Linux, Windows)
- SetRegistry (Jendela)
- UpdateOS (Linux, Windows)

Reboot (Linux, Windows)

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. AWSTOE membuat entri Penjadwal Tugas Windows dengan pemicu yang berjalan secara otomatis di `SystemStartup`
- Linux. AWSTOE menambahkan 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 sebagaisudo user.

Input

Nama Kunci	Deskripsi	Tipe	Wajib	Default
delaySeconds	Menunda jumlah waktu tertentu sebelum memulai reboot.	Bilangan 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 (Jendela)

Modul SetRegistrytindakan 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

Nama Kunci	Deskripsi	Tipe	Wajib
`path`	Jalur kunci registri.	Tali	Ya
`name`	Nama kunci registri.	Tali	Ya
`value`	Nilai kunci registri.	String/Number/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 (Linux, Windows)

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 (023). AL2 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

Nama Kunci	Deskripsi	Tipe	Wajib
`include`	Untuk Windows, Anda dapat menentukan yang berikut: • Satu atau beberapa artikel Pangkalan Pengetahuan Microsoft (KB) IDs untuk disertaka n dalam daftar pembaruan yang mungkin diinstal. Format yang valid adalah `KB1234567` atau`1234567`. • Nama pembaruan menggunakan nilai wildcard (`*`). Format yang valid adalah `Security*` atau`*Security*` . Untuk Linux, Anda dapat menentukan satu atau lebih paket untuk dimasukka	Daftar String	Tidak

Nama Kunci	Deskripsi	Tipe	Wajib
	n dalam daftar pembaruan untuk instalasi.		

Nama Kunci	Deskripsi	Tipe	Wajib
`exclude`	Untuk Windows, Anda dapat menentukan yang berikut: • Satu atau beberapa artikel Pangkalan Pengetahuan Microsoft (KB) IDs untuk disertakan dalam daftar pembaruan yang akan dikecualikan dari penginstalan. Format yang valid adalah `KB1234567` atau `1234567`. • Nama pembaruan menggunakan nilai wildcard (`*`). Format yang valid adalah: `Security*` atau `*Security*` . Untuk Linux, Anda dapat menentukan satu atau lebih paket yang akan dikecualikan dari daftar pembaruan untuk instalasi.	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.

Konfigurasikan 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:

- Sebuah path file lokal (*`./config.json`*).
- Ember S3 (*`s3://<bucket-path>/<bucket-name>/config.json`*).

Saat Anda memasukkan run perintah, Anda dapat menentukan file konfigurasi input menggunakan --config parameter. Misalnya:

```
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 diutamakan

1. Pengaturan yang diberikan langsung ke run perintah di AWS 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 YAMAL 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:
  - Sebuah path file lokal (*./component-doc-example.yaml*).
  - URI S3 (s3://*bucket/key*).
  - Versi build komponen Image Builder ARN (arn:aws:imagebuilder:us-west-:component/ /2021.12.02/1). *2:123456789012 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 Gunakan variabel dalam dokumen komponen kustom Anda halaman.
- ExecutonId (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 YAMAL. 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. `project`Parameter 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>"
}
```

Sumber daya gambar keluaran 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 di AWS 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 di AWS 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, lihatMengelola alur kerja build dan test untuk image Image Builder.

Daftar Isi

- Buat daftar gambar dan versi build
- Lihat detail sumber daya gambar
- Buat gambar kustom dengan Image Builder
- Impor dan ekspor gambar mesin virtual dengan Image Builder
- Impor image disk Windows ISO terverifikasi dengan Image Builder
- Mengelola temuan keamanan untuk image Image Builder
- Bersihkan sumber daya Image Builder

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 ListImagesdi Referensi API EC2 Image Builder. Untuk permintaan SDK terkait, lihat tautan Lihat Juga di halaman yang sama.

Daftar Isi

- Daftar gambar di konsol
- Buat 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/.
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 operasinya (Windows, Linux, atau macOS).

- 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 (VMIE), atau dari image disk ISO.

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.

Jenis

Jenis gambar keluaran yang dibuat Image Builder untuk sumber daya gambar ini (AMI atau gambar kontainer).

Platform

Platform sistem operasi dari sumber daya gambar, misalnya, “Linux”, “Windows”, atau “macOS”.

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 operasinya (Windows, Linux, atau macOS).

- 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 (VMIE), atau dari image disk ISO.

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.

Jenis

Jenis gambar keluaran yang dibuat Image Builder untuk sumber daya gambar ini (AMI atau gambar kontainer).

Versi

<major>Versi platform sistem operasi untuk sumber daya gambar, biasanya bidang numerik dalam format berikut:. <minor>. <patch>.

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 sumber daya gambar, misalnya, “Linux”, “Windows”, atau “macOS”.

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 operasinya (Windows, Linux, atau macOS).
- 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 (VMIE), atau dari image disk ISO.

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.

Jenis

Jenis gambar keluaran yang dibuat Image Builder untuk sumber daya gambar ini (AMI atau gambar kontainer).

Versi

<major>Versi platform sistem operasi untuk sumber daya gambar, biasanya bidang numerik dalam format berikut:. <minor>. <patch>.

Platform

Platform sistem operasi dari sumber daya gambar, misalnya, “Linux”, “Windows”, atau “macOS”.

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.

Buat daftar gambar dengan AWS CLI perintah

Ketika Anda menjalankan list-images perintah di AWS 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"
  },
  {
    "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/image-recipe-
macos/1.1.1",
    "name": "image-recipe-macos",
    "type": "AMI",
    "version": "1.1.1",
    "platform": "macOS",
    "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, ini akan menghentikan 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 ListWaitingWorkflowStepsdi 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/.
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 yang berikut:

- 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 di AWS 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 SDKs, 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 ListImageBuildVersionsdi 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/.
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 di AWS 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/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 sumber daya 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 SDKs, 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 AWS Sumber daya akses yang dibagikan dengan Anda di Panduan AWS RAM Pengguna.

Daftar Isi

- Melihat detail gambar di konsol Image Builder
- Dapatkan detail kebijakan gambar dari 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. Jika tab tidak berlaku untuk gambar Anda, tab tidak aktif dan tidak menampilkan data.

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 versi build gambar saat ini. 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 (AMIs) di EC2 konsol Amazon.

> **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 wadah

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 Image Builder.

Tab konfigurasi infrastruktur

Tab konfigurasi Infrastruktur menampilkan setelan EC2 infrastruktur Amazon yang digunakan Image Builder untuk membangun 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 EC2 key pair Amazon untuk pemecahan masalah
- Topik Amazon SNS untuk pemberitahuan acara

Untuk informasi selengkapnya, lihat Kelola konfigurasi infrastruktur 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 AWS prinsipal yang dipisahkan koma yang memiliki izin untuk meluncurkan gambar Anda, misalnya, atau grup, Akun AWS atau unit organisasi (). AWS Organizations OUs

> **Note**
>
> Ketika Anda memberikan izin kepada prinsipal lain untuk meluncurkan gambar Anda, Anda masih memiliki gambar tersebut. AWS menagih akun Anda untuk semua contoh yang EC2 diluncurkan Amazon dari gambar Anda.

- Akun target untuk konfigurasi peluncuran yang lebih cepat - Akun AWS Tempat Peluncuran EC2 Cepat mendistribusikan snapshot yang telah disediakan sebelumnya untuk diluncurkan.
- Konfigurasi lisensi terkait - Konfigurasi lisensi License Manager ARNs yang terkait dengan AMI di Wilayah tertentu.
- Luncurkan konfigurasi template - Mengidentifikasi template EC2 peluncuran Amazon yang akan digunakan untuk akun tertentu.
- Setel versi default template peluncuran - Menetapkan template EC2 peluncuran Amazon yang ditentukan sebagai templat peluncuran default untuk yang ditentukan Akun AWS.

Resep wadah

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 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. Wadah 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 atau import 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.

Alur kerja gambar yang menghasilkan keluaran AMI dapat memiliki alur kerja build, import, atau pengujian. Alur kerja kontainer yang menghasilkan keluaran kontainer dapat memiliki alur kerja build, pengujian, atau distribusi.

> **Note**
>
> Jika alur kerja belum dimulai, alur kerja tidak muncul dalam daftar. Misalnya, jika build image Anda yang memiliki alur kerja build dan pengujian dikonfigurasi baru saja dimulai, alur kerja build adalah satu-satunya jenis alur kerja yang muncul dalam daftar. Saat alur kerja pengujian dimulai, 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)
- R ollback-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 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

Pengidentifikasi 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, lihat Konfigurasikan 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 digarisbawahi
- Informasi
- Rendah
- Sedang
- Tinggi
- Kritis

Menemukan ID

Pengenal 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 dari 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 kustom dengan Image Builder

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 atau menjalankan pipeline build untuk membuat gambar. Untuk permintaan SDK yang terkait dengan tindakan API, 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 Wisaya pipa: Buat AMI atauWisaya pipa: Buat gambar wadah, tergantung pada jenis gambar yang ingin Anda buat.

1. Buka konsol EC2 Image Builder di 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 konfigurasikan.

AWS CLI

Sebelum Anda menjalankan create-image perintah di AWS 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 pemindaiannya.

- 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 sumber daya gambar keluaran 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 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 dari 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 dan ekspor gambar mesin virtual dengan Image Builder

Saat Anda mengekspor VM dari lingkungan virtualisasinya, proses tersebut membuat satu set file kontainer 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. Untuk mengekspor, Anda dapat membuat file disk VM sebagai output dari build gambar kustom Anda, dan mendistribusikan file.

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 sebagai bagian dari resep gambar Anda, lihatKonfigurasi impor VM.

Impor VM ke Image Builder

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.

Console

Untuk mengimpor VM dengan konsol Image Builder, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder di https://console.aws.amazon.com/imagebuilder/.
2. Pilih Gambar dari panel navigasi.
3. Untuk membuka dialog impor, pilih Impor gambar.
4. Masukkan informasi umum berikut:
   - Tentukan Nama unik untuk gambar Anda.
   - Tentukan Versi untuk gambar dasar. Gunakan format berikut: *major.minor.patch*.
5. Pilih jenis impor: impor VM.
6. Berikan detail untuk masing-masing bagian berikut di halaman Impor gambar. Kemudian pilih Impor gambar setelah selesai.

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

1. 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 VMs di Image Builder, lihatImpor 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 Penampung disk 1.

a. Sumber — Ini bisa berupa bucket S3, atau snapshot EBS.

b. Pilih lokasi disk S3 — Masukkan lokasi di Amazon S3 tempat gambar disk Anda disimpan. Untuk menelusuri lokasi, pilih Browse S3.

c. Untuk menambahkan wadah disk, pilih Tambahkan wadah disk.

2. Peran IAM

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.

3. 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 tanda, pilih Tambahkan tanda. Untuk menghapus sebuah tag, pilih Hapus tag.

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 Import/Export import-image perintah Amazon EC2 VM 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 VM. Import/Export

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 operasi Image Builder API. Untuk meninjau parameter permintaan operasi API, lihat ImportVmImageoperasi 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>. Misalnya, `1.0.0`. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. Pembuatan versi semantik di Image Builder
- 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.
- 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",
    "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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

Mendistribusikan disk VM dari build gambar Anda dari AWS CLI

Anda dapat mengatur distribusi file format disk VM yang didukung ke bucket S3 di Wilayah target sebagai bagian dari proses pembuatan gambar reguler Anda, menggunakan konfigurasi distribusi Image Builder di file. AWS CLI Lihat informasi yang lebih lengkap di Buat pengaturan distribusi untuk output disk VM dari AWS CLI.

Impor image disk Windows ISO terverifikasi dengan Image Builder

File ISO sistem operasi Windows adalah file gambar disk yang berisi paket instalasi lengkap untuk versi tertentu dari sistem operasi Windows. Microsoft menyediakan file ISO sistem operasi Windows resmi untuk diunduh, baik langsung dari situs web mereka atau melalui pengecer resmi. Penting untuk memastikan bahwa Anda mendapatkan file ISO dari sumber tepercaya dan sah untuk menghindari potensi malware atau versi yang tidak sah.

EC2 Image Builder menggunakan alur kerja `build-image-from-iso` impor untuk mengimpor file disk ISO dan membuat volume sekunder darinya. Setelah konfigurasi selesai, Image Builder mengambil snapshot dari volume yang dibuat dari impor dan menggunakannya untuk membuat Amazon Machine Image (AMI).

Sistem operasi yang didukung untuk impor gambar disk ISO

Image Builder mendukung image disk ISO sistem operasi Windows berikut:

- Windows 11 Perusahaan versi 24H2
- Windows 11 Perusahaan versi 23H2
- Windows 11 Perusahaan versi 22H2

Image Builder tidak mendukung image disk ISO sistem operasi Windows berikut:

- Gambar Saluran Layanan Jangka Panjang (LTSC)
- Gambar disk ISO yang dibuat dari Windows Media Creation Tool
- Gambar evaluasi

Prasyarat untuk mengimpor image disk ISO

Untuk mengimpor image disk ISO, Anda harus terlebih dahulu memenuhi prasyarat berikut:

- Sistem operasi disk image haruslah salah satu yang didukung Image Builder. Untuk daftar sistem operasi yang didukung, lihatSistem operasi yang didukung untuk impor gambar disk ISO.
- Untuk memastikan bahwa Anda dapat mengimpor gambar ISO Anda, unduh dari pusat admin Microsoft 365.
- Sebelum Anda dapat menjalankan proses impor, Anda harus mengunggah file disk ISO Anda ke Amazon S3 di tempat yang sama Akun AWS dan di Wilayah AWS mana impor berjalan.

- Ekstensi file peka huruf besar/kecil untuk proses impor, dan harus `.ISO`. Jika ekstensi file Anda huruf kecil, Anda dapat menjalankan salah satu perintah berikut untuk mengganti namanya:

  Command

  ```
  aws s3 cp s3://amzn-s3-demo-bucket/Win11_24H2_English.iso s3://amzn-s3-demo-bucket/Win11_24H2_English.ISO
  ```

  PowerShell

  ```
  Copy-S3Object -BucketName amzn-s3-demo-bucket -Key Win11_24H2_English.iso -DestinationKey Win11_24H2_English.ISO
  ```

- Lisensi Microsoft tidak secara otomatis disertakan dengan impor. Anda harus membawa lisensi Anda sendiri (BYOL). Untuk informasi selengkapnya tentang lisensi untuk perangkat lunak Microsoft, lihat halaman Lisensi di Amazon Web Services dan Pertanyaan Umum Microsoft.
- Proses impor menggunakan dua peran IAM terpisah, sebagai berikut:

  Peran eksekusi

  Peran ini memberikan izin kepada Image Builder untuk menelepon Layanan AWS atas nama Anda. Anda dapat menentukan peran AWSServiceRoleForImageBuilderterkait layanan, yang mencakup izin yang diperlukan untuk peran eksekusi, atau Anda dapat membuat peran Anda sendiri.

  Peran profil instance

  Peran ini memberikan izin untuk tindakan yang dilakukan layanan pada EC2 instance. Anda dapat menentukan peran profil instance dalam sumber daya konfigurasi infrastruktur Anda. Lampirkan kebijakan terkelola berikut ke peran profil instans Anda untuk memastikan bahwa Anda memiliki semua izin yang diperlukan untuk proses impor.

  - EC2InstanceProfileForImageBuilder
  - AmazonSSMManagedInstanceCore

  Untuk informasi selengkapnya, lihat Kelola konfigurasi infrastruktur Image Builder.

Impor image disk ISO ke Image Builder

Sebelum Anda memulai proses impor, pastikan bahwa Anda telah memenuhi semuaPrasyarat.

Proses impor juga menginstal perangkat lunak dan driver berikut pada gambar Anda:

- EC2Luncurkan v2
- AWS Systems Manager agen
- AWS NVMe sopir
- AWS Driver jaringan ENA
- AWS Driver Serial PCI
- EC2 Utilitas Windows

Console

Untuk mengimpor image disk ISO dengan konsol Image Builder, ikuti langkah-langkah berikut:

1. Buka konsol EC2 Image Builder di https://console.aws.amazon.com/imagebuilder/.
2. Pilih Gambar dari panel navigasi.
3. Untuk membuka dialog impor, pilih Impor gambar.
4. Masukkan informasi umum berikut:
   - Tentukan Nama unik untuk gambar Anda.
   - Tentukan Versi untuk gambar dasar. Gunakan format berikut: *major.minor.patch*.
5. Pilih jenis impor: Impor ISO.
6. Masukkan detail konfigurasi impor ISO berikut. Kemudian pilih Impor gambar setelah selesai.
   - URI S3 — Masukkan lokasi penyimpanan file disk ISO Anda. Untuk menelusuri file, pilih Browse S3.
   - Peran IAM — Untuk mengaitkan peran IAM dengan konfigurasi impor 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.

     Anda dapat menentukan peran AWSServiceRoleForImageBuilderterkait layanan, atau Anda dapat menentukan peran kustom Anda sendiri untuk akses layanan.
7. Anda dapat menambahkan tag secara opsional ke sumber daya gambar Image Builder Anda. Ini tidak menambahkan tag ke AMI Anda.
8. Konfigurasi infrastruktur ISO mendefinisikan pengaturan untuk instance yang diluncurkan Image Builder untuk meng-host proses impor. Anda dapat menggunakan konfigurasi

infrastruktur yang dibuat Image Builder, berdasarkan default layanan, atau Anda dapat menggunakan konfigurasi infrastruktur yang ada. Untuk informasi selengkapnya, lihat Kelola konfigurasi infrastruktur Image Builder.

Untuk membuat konfigurasi infrastruktur baru, pilih Buat konfigurasi infrastruktur. Ini terbuka di tab terpisah. Setelah selesai membuat sumber daya baru, Anda dapat kembali ke konfigurasi impor, dan memilih Gunakan konfigurasi infrastruktur yang ada.

9. Untuk memulai proses impor, pilih Impor gambar.

Setelah impor selesai, gambar Anda muncul di daftar gambar yang Anda miliki. Untuk detail selengkapnya, lihat Daftar gambar.

AWS CLI

Contoh ini menunjukkan cara mengimpor gambar dari file disk ISO dan membuat AMI darinya dengan file AWS CLI.

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>. Misalnya, `1.0.0`. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. Pembuatan versi semantik di Image Builder
- deskripsi (string) — Deskripsi resep gambar.
- ExecutionRole (string) - Nama atau Amazon Resource Name (ARN) untuk peran IAM yang memberikan akses Image Builder untuk melakukan tindakan alur kerja untuk mengimpor gambar dari file ISO Microsoft. Anda dapat menentukan peran AWSServiceRoleForImageBuilderterkait layanan, atau Anda dapat menentukan peran kustom Anda sendiri untuk akses layanan.
- platform (string, required) — Platform sistem operasi untuk image disk ISO. Nilai yang valid termasuk`Windows`.
- osVersion (string, required) — Versi sistem operasi untuk image disk ISO. Nilai yang valid termasuk`Microsoft Windows 11`.

- infrastructureConfigurationArn(string, required) - Nama Sumber Daya Amazon (ARN) dari sumber daya konfigurasi infrastruktur yang digunakan untuk meluncurkan EC2 instance tempat image ISO dibuat.
- uri (string, required) — URI file disk ISO yang disimpan di Amazon S3.

```
aws imagebuilder import-disk-image \
    --name "example-iso-disk-import" \
    --semantic-version "1.0.0" \
    --description "Import an ISO disk image" \
    --execution-role "AWSServiceRoleForImageBuilder" \
    --platform "Windows" \
    --os-version "Microsoft Windows 11" \
    --infrastructure-configuration-arn "arn:aws:imagebuilder:us-
east-1:111122223333:infrastructure-configuration/example-infrastructure-
configuration-123456789abc",
    --uri: "s3://amzn-s3-demo-source-bucket/examplefile.iso"
```

Setelah impor selesai, gambar Anda muncul di daftar gambar yang Anda miliki. Untuk detail selengkapnya, lihat Daftar gambar.

PowerShell

Contoh ini menunjukkan cara mengimpor gambar dari file disk ISO dan membuat AMI darinya PowerShell.

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>. Misalnya, `1.0.0`. Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. Pembuatan versi semantik di Image Builder
- deskripsi (string) — Deskripsi resep gambar.
- ExecutionRole (string) - Nama atau Amazon Resource Name (ARN) untuk peran IAM yang memberikan akses Image Builder untuk melakukan tindakan alur kerja untuk mengimpor gambar dari file ISO Microsoft. Anda dapat menentukan peran

AWSServiceRoleForImageBuilderterkait layanan, atau Anda dapat menentukan peran kustom Anda sendiri untuk akses layanan.

- platform (string, required) — Platform sistem operasi untuk image disk ISO. Nilai yang valid termasuk`Windows`.
- osVersion (string, required) — Versi sistem operasi untuk image disk ISO. Nilai yang valid termasuk`Microsoft Windows 11`.
- infrastructureConfigurationArn(string, required) - Nama Sumber Daya Amazon (ARN) dari sumber daya konfigurasi infrastruktur yang digunakan untuk meluncurkan EC2 instance tempat image ISO dibuat.
- uri (string, required) — URI file disk ISO yang disimpan di Amazon S3.

```
Import-EC2IBDiskImage `
    -Name "example-iso-disk-import" `
    -SemanticVersion "1.0.0" `
    -Description "Import an ISO disk image" `
    -ExecutionRole "AWSServiceRoleForImageBuilder" `
    -Platform "Windows" `
    -OsVersion "Microsoft Windows 11" `
    -InfrastructureConfigurationArn "arn:aws:imagebuilder:us-
east-1:111122223333:infrastructure-configuration/example-infrastructure-
configuration-123456789abc" `
    -Uri "s3://amzn-s3-demo-source-bucket/examplefile.ISO"
```

Setelah impor selesai, gambar Anda muncul di daftar gambar yang Anda miliki. Untuk detail selengkapnya, lihat Daftar gambar.

Luncurkan instance dari output AMI

Saat Anda meluncurkan instance dari AMI yang dibuat oleh proses impor, sistem operasi Windows menjalankan Sysprep Specialize, yang secara otomatis mengunduh dan menginstal EC2 Launch v2 dan Agen Systems Manager dari titik akhir S3 publik. Titik akhir ini membutuhkan akses internet publik. Jika Anda meluncurkan instance dari subnet pribadi, proses Sysprep Specialize tidak dapat mengakses titik akhir S3, dan peluncuran gagal.

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.

Konfigurasikan pemindaian keamanan untuk gambar Image Builder di AWS Management Console

Jika Anda telah mengaktifkan Amazon Inspector untuk akun Anda, Amazon Inspector secara otomatis memindai EC2 instance yang diluncurkan Image Builder untuk membuat dan menguji gambar baru. Contoh-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/.
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: Konfigurasikan 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/.
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 sumber daya 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.

Bersihkan sumber daya Image Builder

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 daya Image Builder yang sudah usang atau tidak digunakan](#).

Mengelola kebijakan siklus hidup untuk image 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 unit lain Akun AWS, organisasi, dan organisasi (OUs) 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 kustom atau gambar kontainer. 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 Anda Akun 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 foto yang didistribusikan AMIs dan terkait. 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 image Image Builder
- Daftar kebijakan manajemen siklus hidup untuk sumber daya gambar Image Builder
- Lihat detail kebijakan siklus hidup
- Buat kebijakan siklus hidup
- Cara kerja aturan manajemen siklus hidup untuk sumber daya gambar Image Builder

Prasyarat manajemen siklus hidup untuk image 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, lihatMembuat 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, lihatMembuat 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.

   JSON

```
{
    "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 konfigurasikan 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/*/*/*"
                }
            }
        }
    ]
}
```

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 `Ec2ImageBuilderCrossAccountLifecycleAccess` sebagai nama Peran.

> **Important**
>
> `Ec2ImageBuilderCrossAccountLifecycleAccess`harus menjadi nama peran ini.

8. Setelah meninjau pengaturan, pilih Buat peran.

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,, atau. SDKs AWS CLI

Anda dapat menggunakan salah satu metode berikut untuk mencantumkan sumber daya kebijakan siklus hidup image Builder Image Builder di file Anda. Akun AWS Untuk tindakan API, lihat ListLifecyclePoliciesdi 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 Anda 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/.
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, lihatBuat 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-policiesperintah 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 Anda Wilayah 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 SDKs, 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.

Jenis

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 konfigurasikan.
  - `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, lihatCara 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: ***type of 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 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. AWS Management Console AWS CLI Anda juga dapat menggunakan tindakan CreateLifecyclePolicyAPI. 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/.
2. Pilih Kebijakan Siklus Hidup dari panel navigasi.
3. Pilih Buat kebijakan siklus hidup.
4. Konfigurasikan setelan kebijakan yang dijelaskan dalam prosedur berikut.
5. Untuk membuat kebijakan siklus hidup setelah Anda mengonfigurasi setelan, pilih Buat kebijakan.

Konfigurasikan 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.

Konfigurasikan 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 terkait AMIs 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 AMIs — Pilih kotak centang untuk menandai Amazon terkait dengan tanggal penghentian EC2 AMIs . AMIs 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 AMIs - Pilih kotak centang untuk menonaktifkan Amazon EC2 AMIs terkait. Anda tidak dapat lagi menggunakan AMIs 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 terkait AMIs atau menghapus snapshot untuk itu. AMIs 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 konfigurasikan. 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 AMIs - Pilih kotak centang untuk membatalkan pendaftaran Amazon terkait. EC2 AMIs Anda tidak dapat lagi menggunakan AMIs atau meluncurkan instance baru dari mereka.
- Simpan gambar AMIs, 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 Amazon EC2 AMIs. Anda dapat memasukkan hingga 50 pasangan nilai kunci.

Tanda (opsional)

Tambahkan tag ke kebijakan siklus hidup Anda.

AWS CLI

Untuk membuat kebijakan lifecle 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. AWS Management Console AWS CLI Anda juga dapat menggunakan tindakan CreateLifecyclePolicyAPI. 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/.
2. Pilih Kebijakan Siklus Hidup dari panel navigasi.
3. Pilih Buat kebijakan siklus hidup.
4. Konfigurasikan setelan kebijakan yang dijelaskan dalam prosedur berikut.
5. Untuk membuat kebijakan siklus hidup setelah Anda mengonfigurasi setelan, pilih Buat kebijakan.

Konfigurasi kebijakan: Pengaturan umum

Konfigurasikan 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.

Konfigurasikan 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 konfigurasikan. 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 lifecle Image Builder baru, Anda dapat menggunakan create-lifecycle-policy perintah di. AWS CLI

Cara kerja aturan manajemen siklus hidup untuk sumber daya gambar Image Builder

Kebijakan siklus hidup gambar menggunakan aturan siklus hidup yang Anda tetapkan 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 output AMIs, atau penyimpanan repositori ECR dan kecepatan transfer data untuk gambar kontainer.

Anda dapat mengonfigurasi jenis aturan berikut untuk kebijakan Anda.

Mengusangkan 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 terkait AMIs tanpa memengaruhi kemampuan Anda untuk meluncurkan instance baru.

Ketika AMI tidak digunakan lagi, AMI akan diabaikan oleh penelusuran umum. Misalnya, jika Anda menjalankan EC2 describe-images perintah Amazon di AWS CLI, perintah itu tidak akan kembali usang AMIs dalam set hasil. Namun, Anda masih dapat menemukan usang dengan ID AMI AMIs 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 terkait AMIs atau menghapus snapshot untuk itu. AMIs 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 siklus hidup AMI
- Melihat detail aturan manajemen siklus hidup untuk kebijakan

Aturan pengecualian siklus hidup AMI

Aturan pengecualian berikut menentukan pengecualian pada aturan siklus hidup untuk. AMIs AMIs 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 publik AMIs 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 AMIs 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-policyperintah. Detail kebijakan dalam respons berisi daftar tindakan (aturan) yang Anda tetapkan untuk kebijakan tersebut, yang mencakup semua setelan yang dikonfigurasi.

Konfigurasikan gambar kustom dengan Image Builder

Sumber daya konfigurasi 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 di AWS CLI, lihat Memberi tanda pada sumber daya bagian panduan ini.

Konten

- Mengelola resep di Image Builder
- Kelola konfigurasi infrastruktur Image Builder
- Mengelola pengaturan distribusi Image Builder

Mengelola resep di Image Builder

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. Secara default, Anda dapat menerapkan hingga 20 komponen ke resep. 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 di AWS CLI, lihat Memberi tanda pada sumber daya bagian panduan ini.

> **Tip**
>
> Anda dapat menggunakan komponen yang dikelola Amazon dalam resep Anda, atau Anda dapat mengembangkan komponen kustom Anda sendiri. Untuk informasi selengkapnya, lihat Mengembangkan komponen kustom untuk image Image Builder Anda. Untuk resep gambar yang membuat output AMIs, Anda juga dapat menggunakan produk dan komponen AWS Marketplace gambar. Untuk informasi selengkapnya tentang integrasi dengan AWS Marketplace produk, lihatAWS Marketplace integrasi dalam Image Builder.

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 dari 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 EC2 gambar Image Builder Anda.

Detail resep gambar

- Daftar resep gambar dari konsol
- Daftar resep gambar dari AWS CLI
- Lihat detail resep gambar dari konsol
- Dapatkan detail resep gambar dari AWS CLI
- Dapatkan detail kebijakan resep gambar dari AWS CLI

Daftar resep gambar dari 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/.
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 dari AWS CLI

Contoh berikut menunjukkan bagaimana untuk daftar semua resep gambar Anda, menggunakan AWS CLI.

```
aws imagebuilder list-image-recipes
```

Lihat detail resep gambar dari konsol

Untuk melihat detail resep gambar tertentu menggunakan konsol Image Builder, pilih resep gambar yang akan ditinjau, menggunakan langkah-langkah yang dijelaskan diDaftar resep gambar dari konsol.

Pada halaman detail resep, Anda dapat:

- Hapus resepnya. Untuk informasi selengkapnya tentang menghapus sumber daya di Image Builder, lihatHapus sumber daya Image Builder yang sudah usang atau tidak digunakan.
- 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 Tutorial: Membuat pipeline gambar dengan output AMI dari wizard konsol Image Builder

> **Note**
>
> Saat Anda membuat pipeline dari resep yang ada, opsi untuk membuat resep baru tidak tersedia.

Dapatkan detail resep gambar dari 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 dari 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/.
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 dalamnyaDaftar 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, lihatHapus sumber daya Image Builder yang sudah usang atau tidak digunakan.
- 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 Tutorial: Membuat pipeline gambar dengan output AMI dari wizard konsol 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 dari konsol
- Buat resep gambar dengan AWS CLI
- Impor VM sebagai gambar dasar Anda di konsol

Buat versi resep gambar baru dari 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. Pembuatan versi semantik di Image Builder

  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

Sumber AMI (Diperlukan) - Masukkan ID AMI atau parameter Penyimpanan Parameter AWS Systems Manager (SSM) yang berisi ID AMI untuk digunakan sebagai gambar dasar. Agen SSM harus sudah diinstal sebelumnya di AMI yang dipilih.

- ID AMI — Pengaturan ini tidak diisi sebelumnya dengan entri asli Anda. Masukkan ID AMI untuk gambar dasar Anda. Contoh:*ami-1234567890abcdef1*.
- Parameter SSM — Masukkan nama atau ARN parameter SSM Parameter Store yang berisi ID AMI untuk gambar dasar Anda. Contoh: */ib/test/param* atau `arn:aws:ssm:`*us-east-1*`:`*111122223333*`:parameter`*/ib/test/param*.

- Konfigurasi instans - Pengaturan sudah dipilih sebelumnya, tetapi Anda dapat mengeditnya.
  - Agen Systems Manager - Anda dapat memilih atau menghapus 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.

    Saat Image Builder meluncurkan instance, skrip data pengguna berjalan selama fase cloud-init, sebelum eksekusi komponen dimulai. Langkah ini dicatat ke file berikut pada contoh:`var/log/cloud-init.log`.

    > **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 lebih lanjut dan contoh skrip, lihatPraktik keamanan terbaik untuk 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 menambahkan komponen, pilih Tambahkan komponen build, dan pilih filter kepemilikan yang berlaku. Misalnya, untuk menambahkan komponen build yang terkait dengan AWS Marketplace produk, pilih`AWS Marketplace`. Ini membuka panel pilihan di sisi kanan antarmuka konsol yang mencantumkan AWS Marketplace komponen.
  > Untuk komponen CIS, pilih`Third party managed`.

  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 komponen. 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 parameter input, nilai default muncul di kotak Nilai dengan teks berwarna abu-abu. Jika tidak ada nilai lain yang dimasukkan, Image Builder menggunakan nilai default.

Jika parameter input diperlukan, tetapi tidak memiliki nilai default yang ditentukan dalam komponen, Anda harus memberikan nilai. Image Builder tidak akan membuat versi resep jika ada parameter yang diperlukan yang hilang dan tidak memiliki nilai default yang ditentukan.

> **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.

  > **Note**
  >
  > Jika Anda menggunakan enkripsi untuk volume Anda, Anda harus memilih kunci untuk setiap volume secara terpisah, bahkan jika kuncinya sama dengan yang digunakan untuk volume root.

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 akhir Anda dapat berisi hingga empat kode produk dari produk dan komponen AWS Marketplace gambar. Jika gambar dasar dan komponen yang Anda pilih berisi lebih dari empat kode produk, Image Builder mengembalikan kesalahan saat Anda mencoba membuat resep.

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 kustom dari AWS CLI bagian panduan ini.

Setelah Anda membuat komponen Anda, atau jika Anda menggunakan komponen yang ada, perhatikan ARNs 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 operasi Image Builder API. Untuk meninjau parameter permintaan operasi API, lihat CreateImageRecipeperintah 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 khusus Anda. Anda dapat menentukan gambar induk menggunakan salah satu opsi berikut:
  - ID AMI
  - Sumber daya gambar Image Builder ARN
  - AWS Systems Manager (SSM) Parameter Menyimpan parameter, diawali oleh`ssm:`, diikuti oleh nama parameter atau ARN.
  - AWS Marketplace ID produk

  Note

  Contoh Linux dan macOS 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. Pembuatan versi semantik di Image Builder
- 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 ARNs dengan komponen yang Anda gunakan untuk resep Anda. ARNs

- parameter (array objek) — Berisi array `ComponentParameter` objek. Jika parameter input diperlukan, tetapi tidak memiliki nilai default yang ditentukan dalam komponen, Anda harus memberikan nilai. Image Builder tidak akan membuat versi resep jika ada parameter yang diperlukan yang hilang dan tidak memiliki nilai default yang ditentukan.

  > **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 kefalse, 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:
>
> - `userDataOverride`Kosong 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 berdasarkan 64. Misalnya, perintah berikut dikodekan sebagai:
> `IyEvYmluL2Jhc2gKbWtkaXIgLXAgL3Zhci9iYi8KdG91Y2ggL3Zhcg==`
>
> ```
> #!/bin/bash
> mkdir -p /var/bb/
> touch /var
> ```
>
> Contoh Linux menggunakan nilai yang dikodekan ini.

Linux

Gambar dasar (`parentImage`properti) 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-1234567890abcdef1",
"semanticVersion": "1.0.0",
"components": [
 {
  "componentArn": "arn:aws:imagebuilder:us-west-2:111122223333: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 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-2:111122223333:component/my-example-component/2019.12.02/1"
 },
 {
  "componentArn": "arn:aws:imagebuilder:us-west-2:111122223333:component/my-imported-component/1.0.0/1"
 }
]
}
```

> **Note**
>
> Untuk mempelajari lebih lanjut tentang pembuatan versi semantik untuk sumber daya Image Builder, lihat. Pembuatan versi semantik di Image Builder

macOS

Gambar dasar (`parentImage`properti) 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": "macOS Catalina Image recipe",
"description": "Hello World image recipe for macOS.",
"parentImage": "ami-1234567890abcdef1",
"semanticVersion": "1.0.0",
"components": [
 {
  "componentArn": "arn:aws:imagebuilder:us-
west-2:111122223333:component/catalina$"
 }
],
"additionalInstanceConfiguration": {
 "systemsManagerAgent": {
   "uninstallAfterBuild": true
 },
 "userDataOverride": "IyEvYmluL2Jhc2gKbWtkaXIgLXAgL3Zhci9iYi8KdG91Y2ggL3Zhcg=="
}
}
```

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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

Gambar akhir Anda dapat berisi hingga empat kode produk dari produk dan komponen AWS Marketplace gambar. Jika gambar dasar dan komponen yang Anda pilih berisi lebih dari empat kode produk, Image Builder mengembalikan kesalahan saat menjalankan `create-image-recipe` perintah.

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, lihatWisaya pipa: Buat AMI. Untuk langkah-langkah tambahan untuk membuat resep gambar atau versi resep baru, lihatBuat 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 VMs 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.

Peran IAM

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 tanda, pilih Tambahkan tanda. Untuk menghapus sebuah tag, pilih Hapus tag.

Buat versi baru dari resep wadah

Bagian ini menunjukkan kepada Anda cara membuat versi baru dari resep wadah.

Daftar Isi

- Buat versi resep kontainer baru dengan konsol
- Buat resep wadah dengan AWS CLI

Buat versi resep kontainer 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 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. 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 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. Pembuatan versi semantik di Image Builder

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

- Sumber AMI (Diperlukan) — Identifikasi AMI kustom untuk digunakan sebagai image dasar untuk build container dan instance pengujian Anda. Ini bisa berupa ID AMI atau parameter Penyimpanan Parameter AWS Systems Manager (SSM) yang berisi ID AMI.
  - ID AMI — Pengaturan ini tidak diisi sebelumnya dengan entri asli Anda. Masukkan ID AMI untuk gambar dasar Anda. Contoh:*ami-1234567890abcdef1*.
  - Parameter SSM — Masukkan nama atau ARN parameter SSM Parameter Store yang berisi ID AMI untuk gambar dasar Anda. Contoh: */ib/test/param* atau `arn:aws:ssm:`*us-east-1*`:`*111122223333*`:parameter`*/ib/test/param*.
- 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.

> **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 menambahkan komponen, pilih Tambahkan komponen build, dan pilih filter kepemilikan yang berlaku. Misalnya, untuk menambahkan komponen build yang terkait dengan AWS Marketplace produk, pilih`AWS Marketplace`. Ini membuka panel pilihan di sisi kanan antarmuka konsol yang mencantumkan AWS Marketplace komponen.
  > Untuk komponen CIS, pilih`Third party managed`.

  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 komponen. 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 parameter input, nilai default muncul di kotak Nilai dengan teks berwarna abu-abu. Jika tidak ada nilai lain yang dimasukkan, Image Builder menggunakan nilai default.

Jika parameter input diperlukan, tetapi tidak memiliki nilai default yang ditentukan dalam komponen, Anda harus memberikan nilai. Image Builder tidak akan membuat versi resep jika ada parameter yang diperlukan yang hilang dan tidak memiliki nilai default yang ditentukan.

> **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.

Templat Dockerfile

- Template Dockerfile - Pra-diisi, tetapi dapat diedit. Anda dapat menentukan salah satu variabel kontekstual berikut yang diganti Image Builder dengan informasi build saat runtime.

  ParentImage (wajib)

  Pada waktu pembuatan, variabel ini menyelesaikan gambar dasar untuk resep Anda.

  Contoh:

  ```
  FROM
  {{{ imagebuilder:parentImage }}}
  ```

  lingkungan (diperlukan jika komponen ditentukan)

  Variabel ini akan menyelesaikan skrip yang menjalankan komponen.

  Contoh:

```
{{{ imagebuilder:environments }}}
```

komponen (opsional)

Image Builder menyelesaikan skrip komponen build dan test untuk komponen yang disertakan dalam resep container. Variabel ini dapat ditempatkan di mana saja di Dockerfile, setelah variabel lingkungan.

Contoh:

```
{{{ imagebuilder:components }}}
```

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 kontainer 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 wadah 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 kustom dari AWS CLI bagian panduan ini.

Setelah Anda membuat komponen Anda, atau jika Anda menggunakan komponen yang ada, perhatikan ARNs 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 operasi Image Builder API. Untuk meninjau parameter permintaan operasi API, lihat CreateContainerRecipeperintah 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 ARNs dengan komponen yang Anda gunakan untuk resep Anda,. ARNs Ini termasuk Wilayah AWS, nama, dan nomor versi untuk masing-masing.

- parameter (array objek) — Berisi array `ComponentParameter` objek. Jika parameter input diperlukan, tetapi tidak memiliki nilai default yang ditentukan dalam komponen, Anda harus memberikan nilai. Image Builder tidak akan membuat versi resep jika ada parameter yang diperlukan yang hilang dan tidak memiliki nilai default yang ditentukan.

  > **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.
- 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 kontainer Docker untuk digunakan dalam resep wadah sebagai dasar untuk gambar khusus Anda.
  - Gambar publik dihosting di DockerHub

    - Gambar kontainer yang ada di Amazon ECR
    - Gambar kontainer yang dikelola Amazon
- 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. Pembuatan versi semantik di Image Builder
- 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) - Gambar dasar untuk container build dan test instance. Ini dapat berisi ID AMI atau dapat menentukan parameter Parameter Store AWS Systems Manager (SSM), diawali oleh`ssm:`, diikuti oleh nama parameter atau ARN. Jika Anda menggunakan parameter SSM, nilai parameter harus berisi ID AMI. Jika Anda tidak menentukan gambar dasar, 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-west-2:111122223333: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-1234567890abcdef1",
   "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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

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, lihatHapus sumber daya Image Builder yang sudah usang atau tidak digunakan.

Kelola konfigurasi infrastruktur Image Builder

Anda dapat menggunakan konfigurasi infrastruktur untuk menentukan EC2 infrastruktur Amazon yang digunakan Image Builder untuk membangun dan menguji EC2 image 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.

Untuk gambar Mac, Anda mungkin ingin memilih jenis instans yang mendukung sistem operasi macOS secara native. Untuk informasi selengkapnya, lihat instans Amazon EC2 Mac di Panduan EC2 Pengguna Amazon.

- Pengaturan penempatan instans di mana Anda dapat menentukan host, grup penempatan host, atau Availability Zone tempat instance yang diluncurkan dari gambar Anda harus pergi.
- Profil instans yang memberikan 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 instance juga memerlukan seperangkat izin minimal agar EC2 Image Builder berhasil berkomunikasi dengan instance. Untuk informasi selengkapnya, lihat Siapkan untuk membangun gambar kustom dengan Image Builder.
- 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/*`).
- Amazon EC2 key pair yang memungkinkan Anda masuk ke instans untuk memecahkan masalah jika build gagal dan Anda `terminateInstanceOnFailure` menyetelnya. `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 di AWS CLI.

Daftar Isi

- Daftar dan lihat detail konfigurasi infrastruktur
- Membuat konfigurasi infrastruktur
- Memperbarui konfigurasi infrastruktur

- Image Builder dan AWS PrivateLink antarmuka titik akhir VPC

> **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 di AWS 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 dari AWS CLI
- Dapatkan detail konfigurasi infrastruktur dari AWS CLI

Daftar konfigurasi infrastruktur dari AWS CLI

Contoh berikut menunjukkan cara membuat daftar semua konfigurasi infrastruktur Anda, menggunakan list-infrastructure-configurations perintah di AWS CLI.

```
aws imagebuilder list-infrastructure-configurations
```

Dapatkan detail konfigurasi infrastruktur dari 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/.
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 AWS infrastruktur, Anda dapat mengonfigurasi 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.

     > **Note**
     >
     > Instance Mac berjalan pada tipe `.metal` instans pada Host Khusus. Jenis instans Anda harus cocok dengan salah satu tipe yang ditentukan untuk host yang dijalankannya. Untuk informasi selengkapnya tentang instance Mac dan daftar jenis instans yang mendukung sistem operasi macOS secara native, lihat instans Amazon EC2 Mac di Panduan Pengguna Amazon. EC2

   - Topik SNS (opsional) - Pilih topik SNS untuk menerima pemberitahuan dan peringatan dari Image Builder EC2 .

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. Image Builder dan AWS PrivateLink antarmuka titik akhir VPC
- 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 EC2 instance untuk memecahkan masalah. Jika Anda ingin instance Anda terus berjalan setelah kegagalan build, kosongkan kotak centang.
  - Pasangan kunci — Jika EC2 instance 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 agar diterapkan pada EC2 instance yang digunakan Image Builder untuk membuat 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 EC2 instance yang diluncurkan Image Builder dari build pipeline untuk digunakan 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.

- Di bagian Pengaturan penempatan Instans, Anda dapat mengonfigurasi nilai berikut untuk diterapkan ke EC2 instance yang digunakan Image Builder untuk membangun dan menguji gambar Anda:
  - Anda dapat memilih Availability Zone tempat Image Builder meluncurkan instance selama pembuatan gambar.
  - Secara opsional pilih Tenancy untuk server yang menjalankan instance yang Anda luncurkan. Secara default, EC2 instance berjalan pada perangkat keras penyewaan bersama. Ini berarti bahwa beberapa Akun AWS mungkin berbagi perangkat keras fisik yang sama. Sebuah instance dengan `dedicated` tenancy berjalan pada perangkat keras penyewa tunggal. Sebuah instance dengan `host` tenancy berjalan pada Dedicated Host.

    Instans Mac memerlukan Host Khusus yang dibuat sebagai prasyarat sebelum Anda membuat gambar khusus. Pilih `host` untuk gambar macOS Anda. Anda kemudian dapat memilih host target atau grup sumber daya host untuk meluncurkan instance, tetapi tidak diperlukan jika Host Khusus Anda mengaktifkan penempatan otomatis. Untuk informasi selengkapnya, lihat Penempatan otomatis di Panduan EC2 Pengguna Amazon.
    - ID host penyewaan — ID Host Khusus tempat instans berjalan.
    - Grup sumber daya host penyewaan - Nama Sumber Daya Amazon (ARN) dari grup sumber daya host untuk meluncurkan instance.

6. Di bagian tag Infrastructure (opsional), Anda dapat menetapkan tag metadata ke EC2 instance Amazon yang diluncurkan Image Builder selama proses pembuatan. Tag dimasukkan sebagai pasangan nilai kunci.
7. Di bagian Tag (opsional), Anda dapat menetapkan tag metadata ke sumber daya konfigurasi infrastruktur yang dibuat Image Builder sebagai output. Tag dimasukkan sebagai pasangan nilai kunci.

AWS CLI

Prosedur berikut menunjukkan cara mengkonfigurasi infrastruktur untuk gambar Anda dengan create-infrastructure-configuration perintah Image Builder di AWS CLI. Perintah pada langkah 2 mengambil file yang Anda buat di langkah 1. Untuk contoh ini, file dari langkah 1 dipanggil`create-infrastructure-configuration.json`.

1. Buat file JSON input CLI

   Contoh berikut menunjukkan variasi file JSON yang mungkin Anda buat untuk konfigurasi infrastruktur. Gunakan alat pengeditan file untuk membuat file JSON Anda sendiri.

   Contoh 1: Konfigurasi untuk mempertahankan instance dari build yang gagal

   Contoh ini menentukan dua jenis contoh, `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 instance juga memerlukan seperangkat izin minimal agar EC2 Image Builder berhasil berkomunikasi dengan instance. Untuk informasi selengkapnya, lihat Siapkan untuk membangun gambar kustom dengan Image Builder.

```
{
    "name": "ExampleInfraConfigDontTerminate",
    "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"
}
```

Contoh 2: konfigurasi macOS dengan penempatan otomatis

Contoh ini menentukan jenis dan penempatan instance untuk instance Mac di mana Host Khusus mengaktifkan penempatan otomatis.

```
{
   "name": "macOSInfraConfigAutoPlacement",
   "description": "An example infrastructure configuration for macOS.",
   "instanceProfileName": "EC2InstanceProfileForImageBuilder",
   "instanceTypes": ["mac1.metal, mac2.metal"],
   "terminateInstanceOnFailure": false,
   "placement": {
      "tenancy": "host"
   }
}
```

Contoh 3: konfigurasi macOS dengan ID Host ditentukan

Contoh ini menentukan jenis dan penempatan instance untuk instance Mac yang menargetkan Host Khusus tertentu.

```
{
   "name": "macOSInfraConfigHostPlacement",
   "description": "An example infrastructure configuration for macOS.",
   "instanceProfileName": "EC2InstanceProfileForImageBuilder",
   "instanceTypes": ["mac2-m1ultra.metal"],
   "terminateInstanceOnFailure": false,
   "placement": {
      "tenancy": "host",
      "hostId" : "h-1234567890abcdef0"
   }
}
```

2. 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. Untuk melacak sumber daya Anda, Anda dapat menerapkan tag sebagai berikut. Tag dimasukkan sebagai pasangan nilai kunci.

- Tag sumber daya menetapkan tag metadata ke EC2 instance Amazon yang diluncurkan Image Builder selama proses pembuatan.
- Tag menetapkan tag metadata ke sumber daya konfigurasi infrastruktur yang dibuat Image Builder sebagai output.

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/.
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
```

Image Builder dan AWS PrivateLink antarmuka titik akhir VPC

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 secara pribadi APIs 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. APIs 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 selengkapnya, lihat Membuat titik akhir antarmuka dalam Panduan Pengguna Amazon VPC.

Buat endpoint VPC untuk Image Builder menggunakan nama layanan berikut:

- com.amazonaws. *region*.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 selengkapnya, lihat Mengakses layanan melalui titik akhir antarmuka dalam 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 titik akhir menentukan informasi berikut:

- Prinsipal yang dapat melakukan tindakan.
- Tindakan yang dapat dilakukan.
- Sumber daya yang menjadi target 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**
>
> Ketika kebijakan non-default diterapkan ke titik akhir VPC antarmuka untuk Image EC2 Builder, permintaan API tertentu yang gagal, seperti yang gagal`RequestLimitExceeded`, mungkin tidak dicatat atau Amazon. AWS CloudTrail CloudWatch

Untuk informasi selengkapnya, lihat Mengontrol Akses ke Layanan dengan titik akhir VPC dalam 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 Region (*region*) dan lingkungan aplikasi (*environment*). 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.

JSON

```
{
    "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 Image Builder yang dikelola Amazon. Ganti *regionprincipal-org-id*,, dan *resource-org-id* dengan nilai-nilai organisasi Anda.

JSON

```
{
  "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 *region* dan *environment* dengan nilai-nilai organisasi Anda. Tambahkan izin lain yang diperlukan ke kebijakan berdasarkan persyaratan aplikasi Anda.

> **Note**
>
> Untuk gambar Linux, jika Anda tidak menentukan data pengguna dalam resep gambar Anda, Image Builder menambahkan skrip untuk mengunduh dan menginstal agen Systems Manager pada instance build dan pengujian untuk image Anda. Untuk mengunduh agen, Image Builder mengakses bucket S3 untuk Wilayah build Anda.
> Untuk memastikan bahwa Image Builder dapat mem-bootstrap instance build dan test, tambahkan resource tambahan berikut ke kebijakan endpoint S3 Anda: `"arn:aws:s3:::amazon-ssm-region/*"`

JSON

```
{
  "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 Image Builder

Sebelum mengonfigurasi setelan distribusi untuk gambar keluaran, sebaiknya Anda memverifikasi ketersediaan infrastruktur dasar atau persyaratan lain untuk instance yang diluncurkan dari gambar keluaran Anda di Wilayah target distribusi. Misalnya, tidak semua Wilayah mendukung Host Khusus

EC2 Mac, yang diperlukan untuk meluncurkan instance dari image macOS. Untuk informasi selengkapnya tentang jenis instans dan harga untuk Host Khusus, lihat Harga Host EC2 Khusus Amazon.

Setelah membuat setelan distribusi dengan Image Builder, Anda dapat mengelolanya menggunakan konsol Image Builder, Image Builder API, atau imagebuilder perintah di AWS CLI. Dengan pengaturan distribusi, Anda dapat melakukan tindakan berikut:

Distribusi AMI

- Tentukan nama dan deskripsi AMI keluaran Anda.
- Otorisasi organisasi lain Akun AWS, dan OUs 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`. Untuk informasi dan contoh, lihat ModifyImageAttribute di Referensi Amazon EC2 API.

- Buat salinan output AMI untuk setiap akun target yang ditentukan, organisasi, dan OUs di Wilayah tujuan. Akun target, organisasi, dan OUs memiliki salinan AMI mereka, dan ditagih untuk biaya terkait. Untuk informasi selengkapnya tentang mendistribusikan AMI Anda ke AWS Organizations dan OUs, lihat Berbagi AMI dengan organisasi atau OUs.
- 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 disk VM dari 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 target, akun, AWS Organizations dan unit organisasi (OUs) satu kali, atau dengan setiap build pipeline:

- Untuk secara otomatis mengirimkan gambar yang diperbarui ke Wilayah tertentu, akun, Organizations, dan OUs, gunakan setelan distribusi dengan pipeline Image Builder yang berjalan sesuai jadwal.

- Untuk membuat gambar baru dan mengirimkannya ke Wilayah tertentu, akun, Organizations, dan OUs, gunakan pengaturan distribusi dengan pipeline Image Builder yang Anda jalankan satu kali dari konsol Image Builder, menggunakan Run pipeline dari menu Actions.
- Untuk membuat gambar baru dan mengirimkannya ke Wilayah, akun, Organizations, dan, yang ditentukan OUs, gunakan setelan distribusi dengan tindakan API berikut atau perintah Image Builder di AWS CLI:
  - CreateImageTindakan di Image Builder API.
  - create-imagePerintah di AWS 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 di AWS CLI, lihat Memberi tanda pada sumber daya bagian panduan ini.

Daftar Isi

- Daftar dan lihat detail konfigurasi distribusi
- Membuat dan memperbarui konfigurasi distribusi AMI
- Membuat dan memperbarui pengaturan distribusi untuk gambar kontainer
- Siapkan distribusi AMI lintas akun dengan Image Builder
- Konfigurasikan distribusi AMI dengan template EC2 peluncuran

Daftar dan lihat detail konfigurasi distribusi

Bagian ini menjelaskan berbagai cara agar Anda dapat menemukan informasi dan melihat detail untuk konfigurasi distribusi EC2 Image Builder Anda.

Detail konfigurasi distribusi

- Daftar konfigurasi distribusi dari konsol
- Lihat detail konfigurasi distribusi dari konsol

- Daftar distribusi dari AWS CLI
- Dapatkan detail konfigurasi distribusi dari AWS CLI

Daftar konfigurasi distribusi dari 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/.
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 dari konsol

Untuk melihat detail konfigurasi distribusi tertentu menggunakan konsol Image Builder, pilih konfigurasi yang akan ditinjau, menggunakan langkah-langkah yang dijelaskan dalamDaftar konfigurasi distribusi dari konsol.

Pada halaman detail distribusi, Anda dapat:

- Hapus konfigurasi distribusi. Untuk informasi selengkapnya tentang menghapus sumber daya di Image Builder, lihatHapus sumber daya Image Builder yang sudah usang atau tidak digunakan.
- Edit detail distribusi.

Daftar distribusi dari 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 dari 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

- Prasyarat untuk parameter keluaran SSM
- Buat konfigurasi distribusi AMI
- Perbarui konfigurasi distribusi AMI
- Buat pengaturan distribusi untuk mengaktifkan Peluncuran EC2 Cepat untuk output AMIs
- Buat pengaturan distribusi untuk output disk VM dari AWS CLI

Prasyarat untuk parameter keluaran SSM

Sebelum Anda membuat konfigurasi distribusi AMI baru yang menetapkan AWS Systems Manager parameter Parameter Store (parameter SSM), pastikan Anda telah memenuhi prasyarat berikut.

Peran eksekusi

> Saat membuat pipeline atau menggunakan perintah create-image di dalamnya AWS CLI, Anda hanya dapat menentukan satu peran eksekusi Image Builder. Jika Anda telah menentukan peran eksekusi alur kerja Image Builder, Anda akan menambahkan izin fitur tambahan apa pun ke peran tersebut. Jika tidak, Anda akan membuat peran kustom baru yang menyertakan izin yang diperlukan.

- Untuk menyimpan ID AMI keluaran dalam parameter SSM selama distribusi, Anda harus menentukan `ssm:PutParameter` tindakan dalam peran eksekusi Image Builder, dengan parameter yang tercantum sebagai sumber daya.
- Saat Anda menyetel tipe data parameter `AWS EC2 Image` ke sinyal Systems Manager untuk memvalidasi nilai parameter sebagai ID AMI, Anda juga harus menambahkan `ec2:DescribeImages` tindakan.

Buat konfigurasi distribusi AMI

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

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 EC2 mengkonfigurasi Fast Launch untuk output Windows. AMIs Untuk membuat publik AMI, atur akun terotorisasi izin peluncuran ke `all`. Lihat contoh untuk membuat AMI publik di EC2 ModifyImageAttribute.

Console

Ikuti langkah-langkah berikut untuk membuat konfigurasi distribusi AMI baru di AWS Management Console:

1. Buka konsol EC2 Image Builder di 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 tersebut.

> **Prasyarat**
>
> Untuk menyalin gambar di seluruh akun, Anda harus membuat `EC2ImageBuilderDistributionCrossAccountRole` peran di semua akun target distribusi, 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:
  - AWS akun — Berikan akses ke AWS akun tertentu
  - Unit Organisasi (OU) — Memberikan akses ke OU, dan semua entitas anaknya. Entitas anak termasuk OUs dan AWS akun.
  - Organisasi — Berikan akses ke Anda AWS Organizations, dan semua entitas anaknya. Entitas anak termasuk OUs 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 IDs jenis yang berbeda.

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 template EC2 peluncuran yang akan digunakan untuk meluncurkan instance dari AMI yang Anda buat.

Jika Anda menggunakan template EC2 peluncuran, Anda dapat menginstruksikan Image Builder untuk membuat versi baru template peluncuran yang menyertakan ID AMI terbaru setelah build selesai. Untuk memperbarui template peluncuran, konfigurasikan 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.

iv. Anda dapat memperluas bagian konfigurasi parameter SSM untuk mengonfigurasi parameter SSM yang akan menyimpan ID AMI keluaran untuk gambar yang

didistribusikan ke Wilayah tujuan. Anda dapat secara opsional menentukan akun distribusi di Wilayah.

Nama parameter — Masukkan nama untuk parameter Anda. Sebagai contoh, `/output/image/param`.

Tipe data - Pertahankan nilai default (`AWS EC2 Image`). Ini memberitahu Systems Manager untuk memvalidasi nilai parameter untuk memastikan bahwa itu adalah ID AMI yang valid.

b. Untuk menambahkan setelan distribusi untuk Wilayah lain, pilih Tambah Wilayah.

7. Pilih Buat pengaturan setelah Anda selesai.

AWS CLI

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 mana Akun AWS, AWS Organizations atau unit organisasi (OUs) yang memiliki izin untuk meluncurkan AMI yang Anda distribusikan ke Wilayah tertentu. Beri nama file`create-ami-distribution-configuration.json`, untuk digunakan pada langkah berikutnya:

   Contoh 1: Distribusikan ke Akun AWS

   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"
     ]
    }
   }
  }
 ]
}
```

Contoh 2: Distribusikan ke Organizations dan 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"
     ]
    }
   }
  }
 ]
}
```

Contoh 3: Simpan ID AMI keluaran dalam parameter SSM

Contoh ini menyimpan ID AMI keluaran dalam AWS Systems Manager parameter Parameter Store di Region distribusi.

```
{
 "name": "SSMParameterOutputAMI",
 "description": "Updates an SSM parameter with the output AMI ID for the
 distribution.",
 "distributions": [
  {
   "region": "us-west-2",
   "amiDistributionConfiguration": {
    "name": "Name {{ imagebuilder:buildDate }}"
   },
   "ssmParameterConfigurations": [
    {
     "amiAccountId": "111122223333",
     "parameterName": "/output/image/param",
     "dataType": "aws:ec2:image"
    }
   ]
  }
 ]
}
```

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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

Untuk informasi lebih rinci, lihat create-distribution-configuration di AWS CLI Command Reference.

Perbarui konfigurasi distribusi AMI

Anda dapat mengubah konfigurasi distribusi AMI Anda. Namun, perubahan yang Anda buat tidak berlaku untuk sumber daya apa pun yang telah didistribusikan oleh 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.

AWS Management Console

Ikuti langkah-langkah berikut ke konfigurasi distribusi AMI di AWS Management Console:

1. Buka konsol EC2 Image Builder di 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, lihatBuat konfigurasi distribusi AMI.

5. Setelah selesai, pilih Simpan perubahan.

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 untuk membuat file JSON dengan kunci yang ditunjukkan pada contoh berikut, dan nilai yang valid untuk lingkungan Anda. Contoh ini menggunakan file bernamaupdate-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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

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.

Buat pengaturan distribusi untuk mengaktifkan Peluncuran EC2 Cepat untuk output AMIs

Contoh berikut menunjukkan cara menggunakan create-distribution-configuration perintah untuk membuat pengaturan distribusi yang memiliki EC2 Fast Launch dikonfigurasi untuk AMI Anda, dari AWS CLI.

> **Note**
>
> Image Builder tidak mendukung distribusi lintas akun AMIs dengan Peluncuran EC2 Cepat yang telah diaktifkan sebelumnya. EC2 Peluncuran Cepat harus diaktifkan dari akun tujuan.

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.",
   "amiTags": {
    "KeyName": "Some Value"
   }
  },
  "fastLaunchConfigurations": [{
   "enabled": true,
   "snapshotConfiguration": {
    "targetResourceCount": 5
   },
   "maxParallelLaunches": 6,
   "launchTemplate": {
    "launchTemplateId": "lt-0ab1234c56d789012",
    "launchTemplateVersion": "1"
    }
  }],
  "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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

Untuk informasi lebih rinci, lihat create-distribution-configuration di AWS CLI Command Reference.

Buat pengaturan distribusi untuk output disk VM dari 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 operasi Image Builder API. Untuk meninjau parameter permintaan operasi 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 Import/Export izin 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 ESX VMware dan vSphere versi 4, 5, dan VMware 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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

Untuk informasi lebih rinci, lihat create-distribution-configuration di AWS CLI Command Reference.

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 dari AWS CLI
- Perbarui setelan distribusi untuk gambar kontainer Anda dari AWS CLI

Buat pengaturan distribusi untuk gambar kontainer Image Builder dari 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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

Untuk informasi lebih rinci, lihat create-distribution-configuration di AWS CLI Command Reference.

Perbarui setelan distribusi untuk gambar kontainer Anda dari 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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

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, lihatBuat resep gambar dengan AWS CLI. Untuk membuat file JSON untuk konfigurasi infrastruktur, lihatMembuat konfigurasi infrastruktur.

Prasyarat untuk distribusi AMI lintas akun

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 dienkripsi AMIs, 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, konfigurasikan 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 Image Builder AMIs 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. Contoh berikut menunjukkan kebijakan kepercayaan di akun tujuan yang menentukan ID akun dari akun sumber.

   JSON

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::444455556666:root"
        },
        "Action": "sts:AssumeRole"
    }]
}
```

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` `Principal`Bagian 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.

   JSON

   ```
   {
    "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 templat EC2 peluncuran Amazon, Anda juga harus menambahkan kebijakan berikut ke akun Anda `EC2ImageBuilderDistributionCrossAccountRole` di setiap 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"
                }
            }
        }
    ]
}
```

6. Jika Anda menggunakan AWS Systems Manager parameter Parameter Store untuk menyimpan ID AMI keluaran AMI untuk akun distribusi dan Wilayah, Anda harus menambahkan kebijakan berikut ke akun Anda `EC2ImageBuilderDistributionCrossAccountRole` di setiap akun tujuan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:111122223333:parameter/ImageBuilder-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeImages"
            ],
            "Resource": "*"
        }
    ]
}
```

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 virtualisasi PV. AMIs 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 CreateDistributionConfigurationdi Referensi API EC2 Image Builder.

Konfigurasikan distribusi lintas akun untuk Image Builder AMI dari konsol

Bagian ini menjelaskan cara membuat dan mengonfigurasi setelan distribusi untuk distribusi lintas akun Image Builder Anda AMIs menggunakan. AWS Management Console Mengkonfigurasi distribusi lintas akun memerlukan izin IAM tertentu. Anda harus menyelesaikan Prasyarat untuk distribusi AMI lintas akun 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/.
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.

Konfigurasikan distribusi lintas akun untuk Image Builder AMI dari 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 distribusi AMI lintas akun untuk bagian ini sebelum Anda menjalankan create-image perintah.

1. Konfigurasikan file pengaturan distribusi

   Sebelum Anda menggunakan create-image perintah di AWS CLI untuk membuat Image Builder AMI yang didistribusikan ke akun lain, Anda harus membuat struktur `DistributionConfiguration` JSON yang menentukan akun target IDs dalam pengaturan. `AmiDistributionConfiguration` Anda harus menentukan setidaknya satu `AmiDistributionConfiguration` di Wilayah sumber.

   File contoh berikut, bernama`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-configurationperintah 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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

Anda juga dapat memberikan JSON langsung dalam perintah, menggunakan *`--distributions`* parameter.

Konfigurasikan distribusi AMI dengan template EC2 peluncuran

Untuk membantu memastikan pengalaman peluncuran yang konsisten untuk Image Builder AMI di akun target dan Wilayah, Anda dapat menentukan templat EC2 peluncuran Amazon di setelan distribusi, menggunakan`launchTemplateConfigurations`. Ketika `launchTemplateConfigurations` hadir selama proses distribusi, Image Builder membuat versi baru dari template peluncuran yang mencakup semua pengaturan asli dari template, dan ID AMI baru dari build. Untuk informasi selengkapnya tentang meluncurkan EC2 instance 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

> **Note**
>
> Ketika Anda menyertakan template peluncuran untuk mengaktifkan Windows Fast Launch dalam gambar Anda, template peluncuran harus menyertakan tag berikut sehingga Image Builder dapat mengaktifkan Windows Fast Launch atas nama Anda.
> `CreatedBy: EC2 Image Builder`

Tambahkan template EC2 peluncuran ke pengaturan distribusi AMI dari 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/.
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 EC2 peluncuran dari daftar. Jika tidak ada template peluncuran di akun Anda, pilih Buat templat peluncuran baru, yang membuka Template Peluncuran di EC2Dasbor.

   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 EC2 peluncuran ke setelan distribusi AMI dari 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. Konfigurasikan 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-configurationperintah 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-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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

Anda juga dapat memberikan JSON langsung dalam perintah, menggunakan *`--distributions`* parameter.

Bagikan sumber daya Image Builder dengan AWS RAM

EC2 Image Builder terintegrasi dengan AWS Resource Access Manager (AWS RAM) sehingga Anda dapat berbagi jenis sumber daya Image Builder berikut dengan salah satu Akun AWS atau melalui AWS Organizations.

- Komponen-komponen
- Citra
- Resep

Untuk berbagi sumber daya AWS RAM, Anda harus 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. Daftar berikut mencakup jenis akun dan organisasi yang dapat Anda bagikan.

- 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 OUs di luar organisasinya.

Dalam model ini, Akun AWS yang memiliki sumber daya (pemilik) membaginya dengan orang lain Akun AWS atau melalui AWS Organizations (konsumen) dalam Wilayah yang sama. Ketika sumber daya bersama diperbarui, konsumen mendapatkan pembaruan tersebut secara otomatis.

> **Note**
>
> 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 dibagikan dengan mereka.

Topik

- Pemilik sumber daya
- Konsumen sumber daya
- Buat pembagian AWS RAM sumber daya untuk sumber daya Image Builder

- Membatalkan pembagian sumber daya Image Builder dari AWS RAM

Pemilik sumber daya

Sumber daya Image Builder hanya dapat dibagikan di Wilayah AWS tempat mereka dibuat. Ketika Anda berbagi sumber daya ini, mereka tidak akan mereplikasi di seluruh Wilayah.

Untuk mendapatkan daftar sumber daya Image Builder yang Anda miliki dan dapat berbagi, tentukan filter kepemilikan di konsol atau ketika Anda menjalankan perintah di AWS CLI.

- Daftar komponen Image Builder
- Daftar gambar
- Daftar dan lihat detail resep gambar
- Daftar dan lihat detail resep wadah

Untuk informasi selengkapnya AWS RAM, lihat Panduan AWS RAM Pengguna.

Prasyarat untuk berbagi sumber daya Image Builder

Untuk berbagi sumber daya Image Builder, seperti komponen, gambar, atau resep:

- Anda Akun AWS harus memiliki sumber daya Image Builder yang ingin Anda bagikan. 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. OUs
- Untuk berbagi sumber daya Image Builder dengan AWS Organizations dan OUs menggunakan 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 Wilayah AWS, dan ke, akun, dan organisasi atau unit organisasi (OUs) yang Anda tentukan dalam konfigurasi distribusi.

- Gambar Mesin Amazon (AMIs)
- Gambar kontainer yang berada di Amazon ECR

Untuk informasi selengkapnya tentang setelan distribusi untuk AMI Anda, lihatMembuat dan memperbarui konfigurasi distribusi AMI. Untuk informasi selengkapnya tentang setelan distribusi untuk image container Anda di Amazon ECR, lihatMembuat dan memperbarui pengaturan distribusi untuk gambar kontainer.

Untuk informasi selengkapnya tentang berbagi AMI dengan AWS Organizations dan OUs, lihat Berbagi AMI dengan organisasi atau OUs.

Konsumen sumber daya

Konsumen dapat menggunakan sumber daya bersama, tetapi tidak dapat memodifikasinya dengan cara apa pun. Saat mereka membuat resep Image Builder, mereka dapat menentukan gambar bersama sebagai gambar dasar, dan mereka dapat menambahkan komponen bersama. Mereka juga dapat menentukan resep bersama saat mereka membuat pipeline image Image Builder, atau saat mereka menggunakan create-image perintah di file AWS CLI.

Jika Anda termasuk dalam organisasi di AWS Organizations, dan berbagi dalam organisasi Anda diaktifkan, konsumen di organisasi Anda secara otomatis diberikan akses ke sumber daya bersama. Jika tidak, konsumen menerima undangan untuk bergabung dengan pembagian sumber daya dan diberikan akses ke sumber daya bersama setelah menerima undangan.

Buat pembagian AWS RAM sumber daya untuk sumber daya Image Builder

Untuk membagikan komponen, gambar, atau resep Image Builder, Anda harus menambahkannya ke pembagian AWS Resource Access Manager sumber daya. Pembagian sumber daya menentukan sumber daya untuk dibagikan dan konsumen dengan siapa mereka dibagikan.

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 file AWS CLI.

Opsi 2: Terapkan kebijakan sumber daya dan promosikan ke pembagian sumber daya yang ada

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. Terapkan 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 di AWS 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 di AWS 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 di AWS 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 di AWS 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 pembagian sumber daya Image Builder dari AWS RAM

Untuk membatalkan pembagian sumber daya Image Builder yang Anda miliki, seperti komponen bersama, gambar, atau resep, Anda harus menghapusnya dari pembagian AWS Resource Access Manager sumber daya. Anda dapat melakukan ini menggunakan AWS RAM konsol atau AWS CLI.

> **Note**
>
> Pemilik tidak dapat menghapus sumber daya bersama hingga tidak lagi dibagikan. Pemilik tidak dapat membatalkan pembagian sumber daya ini sampai tidak ada konsumen yang bergantung padanya.

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.

Tag sumber daya keluaran 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 build date/time pada waktu build.
- `- {{imagebuilder:buildVersion}}`

  Menyelesaikan ke versi build, yang merupakan nomor yang terletak di akhir Image Builder Amazon Resource Name (ARN.) Misalnya, `"arn:aws:imagebuilder:us-west-2:123456789012:component/myexample-component/2019.12.02/1"` menampilkan versi build sebagai`1`.

Untuk membantu Anda melacak Amazon Machine Images (AMIs) yang telah didistribusikan, Image Builder secara otomatis menambahkan tag berikut ke output Anda AMIs.

- `"CreatedBy":"EC2 Image Builder"`
- `"Ec2ImageBuilderArn":"arn:aws:imagebuilder:`*`us-west-2`*`:`*`123456789012`*`:image/`*`simple-recipe-linux`*`/`*`1.0.0/10`*`"`. Tag ini berisi ARN sumber daya image Image Builder yang digunakan untuk membuat AMI.

Daftar Isi

- Tag sumber daya dari AWS CLI
- Hapus tag sumber daya dari AWS CLI
- Daftar semua tag untuk sumber daya tertentu dari AWS CLI

Tag sumber daya dari 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
```

Hapus tag sumber daya dari 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 dari 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 daya Image Builder yang sudah usang atau tidak digunakan

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 resource tidak menghapus image container Amazon EC2 AMIs atau 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 AWS CLI perintah.

> **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 dari 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 pipa gambar dari 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 dari 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 dari 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 image 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.

Mengelola alur kerja build dan test untuk image Image Builder

Alur kerja image 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 melewatkan 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.

Topik alur kerja

- Kerangka alur kerja: Tahapan
- Akses layanan
- Menggunakan alur kerja terkelola untuk gambar Anda
- Daftar alur kerja gambar
- Buat alur kerja gambar
- Membuat dokumen alur kerja YAMB

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 tahapan berbeda berikut.

1. Tahap build (pre-snapshot) — Selama tahap build, Anda membuat perubahan pada instance EC2 build Amazon yang menjalankan image dasar Anda, untuk membuat baseline untuk gambar 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 dibuat AMIs dan gambar kontainer. Untuk alur kerja AMI, Image Builder meluncurkan EC2 instance 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 AWSServiceRoleForImageBuilderterkait 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 CreateImagetindakan, 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.

Menggunakan alur kerja terkelola untuk gambar Anda

Alur kerja terkelola dibuat dan dikelola oleh AWS. Saat Anda menggunakan alur kerja terkelola di pipeline gambar atau untuk pembuatan gambar satu kali, Anda dapat memilih Nama Sumber Daya Amazon (ARN) dari alur kerja terkelola yang ingin Anda gunakan. Amazon menyediakan versi terbaru yang memiliki tambalan dan pembaruan lainnya yang diterapkan. Untuk mendapatkan daftar alur kerja terkelola, lihatDaftar alur kerja gambar, dan filter di Owner = Amazon (konsol).

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 APISDKs, 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 ListWorkflowsdi Referensi API EC2 Image Builder. Untuk permintaan SDK terkait, lihat tautan Lihat Juga di halaman yang sama.

Anda dapat memfilter detail berikut untuk merampingkan daftar hasil. Misalnya, jika Anda memfilter Owner = Amazon di konsol, daftar hanya menampilkan alur kerja AWS terkelola.

- Alur kerja
- Versi
- Tipe
- Pemilik

Console

Rincian alur kerja

Detail pada halaman daftar alur kerja Image di konsol Image Builder mencakup hal-hal berikut:

- Alur kerja - 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/.
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. Defaultnya 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 di AWS 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, lihatEnkripsi dan manajemen kunci di 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/.
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. Konfigurasikan 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.

Memulai

- 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 YAMAL 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 di AWS 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 CLI create-workflow. `--data`Parameter 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 CLI create-workflow. `--data`Parameter 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 dokumen alur kerja
- Parameter masukan dokumen alur kerja
- Langkah-langkah dokumen alur kerja
- Output dokumen alur kerja

Identifikasi dokumen alur kerja

Mengidentifikasi alur kerja secara unik. Bagian ini dapat mencakup atribut berikut.

Bidang	Deskripsi	Tipe	Wajib
name	Nama dokumen alur kerja.	String	Tidak
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 masukan dokumen alur kerja

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	Wajib	Batasan
name	Nama parameter.	Tali	Ya	
deskripsi	Deskripsi parameter.	String	Tidak	
default	Nilai default parameter, jika tidak ada nilai yang disediakan. Jika Anda tidak menyertakan nilai default dalam definisi parameter, nilai parameter diperlukan saat runtime.	Cocokkan dengan tipe data parameter.	Tidak	
jenis	Tipe data parameter. Jika Anda tidak menyertakan tipe data dalam definisi parameter,	Tali	Ya	Tipe data parameter harus salah satu dari yang berikut: • `string` •

Bidang	Deskripsi	Tipe	Wajib	Batasan
	tipe parameter default ke nilai string yang diperlukan saat runtime.			• `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 dokumen alur kerja

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 JSONPath pemilih. Misalnya:

```
$.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	Wajib	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.
`if`, diikuti oleh serangkaian pernyataan bersyarat yang memodifikasi `if` operator.	Pernyataan bersyarat menambahkan alur titik keputusan kontrol ke badan langkah alur kerja Anda.	Dikte	Tidak		Image Builder mendukung pernyataan bersyarat berikut sebagai pengubah operator: `if` • Kondisi percabangan dan pengubah: `if`,,`and`,`or`.

Bidang	Deskripsi	Tipe	Wajib	Nilai default	Batasan
					not Kondisi percabangan ditentukan pada garis sendiri. • Operator perbandingan:booleanEquals ,numberEquals ,numberGreaterThan ,numberGreaterThanEquals ,numberLessThan ,numberLessThanEquals ,stringEquals .
deskripsi	Deskripsi langkah.	String	Tidak		String kosong tidak diperbolehkan. Jika disertakan, panjangnya harus 1-1024 karakter.

Bidang	Deskripsi	Tipe	Wajib	Nilai default	Batasan
masukan	Berisi parameter yang perlu dijalankan oleh tindakan langkah. Anda dapat menentukan nilai kunci sebagai nilai statis, atau dengan JSONPath variabel yang menyelesaikan ke tipe data yang benar.	Dikte	Ya		
name	Nama langkah. Nama ini harus unik dalam dokumen alur kerja.	Tali	Ya		Panjangnya harus antara 3-128 karakter. Dapat menyertakan karakter alfanumerik dan. _ Tidak ada spasi.

Bidang	Deskripsi	Tipe	Wajib	Nilai default	Batasan
onFailure	Mengkonfigurasi tindakan yang akan diambil jika langkah gagal, sebagai berikut. Perilaku • `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	Wajib	Nilai default	Batasan
	gagal, dan berlanjut hingga semua langkah yang memungkinkannya diputar kembali. • `Continue`—Gagal langkah, tetapi terus menjalankan langkah-langkah yang tersisa setelah langkah yang gagal. Dalam hal ini, tidak ada rollback.				

Bidang	Deskripsi	Tipe	Wajib	Nilai default	Batasan
RollBackenabled	Mengkonfigurasi apakah langkah akan diputar kembali jika terjadi kegagalan. Anda dapat menggunakan nilai Boolean statis atau JSONPath variabel dinamis yang menyelesaikan nilai Boolean.	Boolean	Tidak	`true`	`true` | `false` | atau JSONPath variabel yang menyelesaikan menjadi benar atau salah.

Bidang	Deskripsi	Tipe	Wajib	Nilai default	Batasan
timeoutSeconds	Waktu maksimum, dalam hitungan detik, langkah berjalan sebelum gagal dan mencoba lagi, jika percobaan ulang berlaku.	Bilangan 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 dokumen alur kerja

Mendefinisikan output untuk alur kerja. Setiap output adalah pasangan nilai kunci yang secara spesifik nama output dan nilainya. Anda dapat menggunakan output untuk mengekspor data saat runtime yang dapat digunakan alur kerja berikutnya. Bagian ini opsional.

Setiap output yang Anda tentukan mencakup atribut berikut.

Bidang	Deskripsi	Tipe	Wajib
name	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 Manager Menjalankan 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	Wajib	Default	Batasan
instanceId	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 dikembali kan dari Systems ManagersendCommand.	String
output	Output dikembali kan 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 menjalankan Agen AWS Systems Manager (Systems Manager) pada EC2 instance 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 dibuat AMIs.

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	Wajib	Default	Batasan
instanceId	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	Wajib	Default	Batasan
instanceId	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 dikembali kan dari Systems ManagersendCommand.	String
output	Output dikembali kan 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 `CreateImage` API. 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	Wajib	Default	Batasan
instanceId	Contoh untuk membuat gambar baru dari.	Tali	Ya		Instance untuk ID instance yang disediakan harus dalam `running` 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 Manager Menjalankan 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	Wajib	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 sendCommand yang menjalankan komponen pada instance.	String

Nama output	Deskripsi	Tipe
status	Status dikembali kan dari Systems ManagersendCommand.	String
output	Output dikembali kan 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	Wajib	Default	Batasan
imageIdOverride	Gambar yang akan digunakan untuk meluncurkan instance	String	Tidak	Membangun panggung: Gambar dasar resep gambar Tahap pengujian: Keluarkan AMI dari tahap pembuatan	Harus berupa ID AMI yang valid
instanceTypesOverride	Image Builder mencoba setiap jenis instance dalam daftar hingga menemukan satu yang berhasil diluncurkan	Daftar String	Tidak	Jenis instans yang ditentukan dalam Konfigurasi Infrastruktur	Harus jenis instance yang valid
Menunggu	Kondisi untuk menunggu sebelum menyelesaikan langkah alur kerja dan	Tali	Ya		Image Builder mendukung `ssmAgent`.

Nama masukan	Deskripsi	Tipe	Wajib	Default	Batasan
	melanjutkan ke langkah berikutnya				

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 Manager Menjalankan 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	Wajib	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 untuk parameter apa pun yang diperlukan dokumen perintah.	<string>kamus <string, daftar>	Bersyarat		
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 dikembali kan dari Systems ManagersendCommand.	String
output	Output dikembali kan 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	Wajib	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 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

Nama output	Deskripsi	Tipe
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, lihatDiperlukan 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 Manager Menjalankan 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	Wajib	Default	Batasan
instanceId	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
runCommandId	ID Systems Manager sendCommand yang menjalankan skrip sanitasi 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: 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	Wajib	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	Wajib	Default	Batasan
snsTopicArn	Topik SNS opsional ARN untuk mengirim pemberitahuan saat langkah alur kerja tertunda.	String	Tidak		

Output: Tabel berikut mencakup output untuk tindakan langkah ini.

Nama output	Deskripsi	Tipe
tindakan	Tindakan yang dikembalikan oleh tindakan SendWorkflowStepAction 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 JSONPath pemilih dengan node struktural yang secara unik mengidentifikasi variabel target.

JSONPath struktur variabel alur kerja dinamis

```
$.<document structure>.[<step name>.]<variable name>
```

Node pertama setelah root ($) mengacu pada struktur dokumen alur kerja, sepertistepOutputs, atau dalam kasus variabel sistem Image Builder,imageBuilder. Daftar berikut berisi node struktur dokumen JSONPath alur kerja 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 dalam JSONPath adalah nama variabel target, seperti`instanceId`.

Setiap langkah dapat merujuk ke output dari setiap tindakan langkah sebelumnya dengan variabel JSONPath dinamis 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
```

Ketika parameter input mengacu pada variabel dinamis, indikator rantai (`.$`) harus dilampirkan ke akhir nama parameter, seperti yang ditunjukkan pada contoh berikut.

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/<recipe-name>`	String	`/aws/imagebuilder/sampleImageRecipe`
cloudWatchLogStreaming	Nama aliran CloudWatch Log untuk log keluaran.	String	`1.0.0/1`

Nama variabel	Deskripsi	Tipe	Nilai contoh
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	Nomor versi build gambar.	Bilangan Bulat	*1*
imageId	Id AMI dari gambar dasar.	String	*ami-12345 67890abcdef1*
ImageName	Nama gambar.	String	*sampleImage*
ImageType	Jenis output gambar.	String	`AMI` | `Docker`
imageVersionNumber	Nomor versi gambar.	String	*1.0.0*
instanceProfileName	Nama peran profil instance yang digunakan Image Builder untuk meluncurkan instance build dan test.	String	*SampleIma geBuilder InstanceP rofileRole*

Nama variabel	Deskripsi	Tipe	Nilai contoh
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	{'S3logs': {'s3 BucketName ':' ', 's3 *sample-bucket* KeyPrefix': '}} *ib-logs*
securityGroups	Grup keamanan IDs yang berlaku untuk membangun dan menguji instance.	Daftar [String]	*[sg-1234567890abcdef1, sg-11112222333344445]*
SourceImagearn	Nama Sumber Daya Amazon (ARN) dari sumber daya image Image Builder yang digunakan alur kerja untuk tahap pembuatan dan pengujian.	String	arn:aws:imagebuilder: ::image//*us-east-1 111122223333 sampleImage 1.0.0/1*
subnetId	ID subnet untuk meluncurkan instance build dan test.	String	*subnet-1234567890abcdef1*

Nama variabel	Deskripsi	Tipe	Nilai contoh
terminateInstanceOnKegagalan	Nilai setelan saat ini yang mengarahkan Image Builder untuk menghentikan instance jika gagal atau menyimpannya untuk pemecahan masalah.	Boolean	`true` | `false`
WorkflowPhase	Tahap saat ini yang berjalan untuk eksekusi alur kerja.	String	`Build` | `Test`
workingDirectory	Jalur ke direktori kerja.	String	`/tmp`

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 terselesaikan`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 operator perbandingan berikut yang melakukan perbandingan nilai (sama, kurang dari, lebih besar dari) berdasarkan tipe data yang dibandingkannya (string atau angka).

Operator perbandingan 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, lihatLevel bersarang dalam pernyataan bersyarat.
- Setiap pernyataan percabangan harus memiliki setidaknya satu pernyataan bersyarat anak, tetapi tidak lebih dari sepuluh.
- Pengubah bersyarat beroperasi hanya pada satu pernyataan bersyarat anak.

Level bersarang dalam pernyataan bersyarat

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)
```

`if`atribut

`if`Atribut 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 pernyataan bersyarat

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

Operator `booleanEquals` perbandingan 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

Operator `stringEquals` perbandingan membandingkan dua string dan mengembalikan nilai true jika string 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

Operator `numberEquals` perbandingan membandingkan dua angka dan mengembalikan nilai true jika angkanya sama. Angka-angka untuk membandingkan harus menjadi salah satu format berikut.

- Bilangan Bulat
- Desimal
- String yang cocok dengan pola regex berikut:. `^-?[0-9]+(\.)?[0-9]+$`

Contoh perbandingan berikut semuanya mengevaluasi. `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
```

Mengelola pembuatan gambar kustom di Image Builder melalui proses pipeline berulang

Image Builder image pipelines menyediakan kerangka kerja otomatisasi untuk membuat dan memelihara gambar kustom AMIs dan kontainer. Pipa memberikan fungsionalitas berikut:

- Merakit gambar dasar, komponen untuk bangunan dan pengujian, 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
- Konfigurasikan alur kerja pipa gambar di Image Builder
- Jalankan pipeline gambar Anda
- Gunakan ekspresi cron di 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 EC2 gambar Image Builder Anda.

Detail saluran pipa

- Daftar pipa gambar dari AWS CLI
- Dapatkan detail pipeline gambar dari AWS CLI

Daftar pipa gambar dari 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 pipeline gambar dari 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 terkelola Amazon yang dimulai cepat, gambar yang Anda buat atau yang dibagikan dengan Anda, atau gambar yang Anda berlangganan melalui. AWS Marketplace Resep ini juga mencakup komponen yang melakukan

tugas-tugas berikut pada EC2 instance 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 setelan 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 distribusi yang AMIs 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, lihatTutorial: Membuat pipeline gambar dengan output AMI dari wizard konsol Image Builder.

Daftar Isi

- Buat pipeline gambar AMI dari AWS CLI
- Perbarui pipeline gambar AMI dari konsol
- Perbarui pipeline gambar AMI dari AWS CLI

Buat pipeline gambar AMI dari 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, lihatGunakan ekspresi cron di 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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

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 AMI dari 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, lihat Perbarui pipeline gambar AMI dari 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/.
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 di AWS CLI. Untuk informasi selengkapnya, lihat Mengelola resep di Image Builder.
- 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, lihatGunakan 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 setelan 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. 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
  - 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 Image Builder

Perbarui pipeline gambar AMI dari 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 Amazon Resource Names (ARNs) 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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

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
- Menambah dan menghapus 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, lihatTutorial: Buat pipeline gambar dengan output gambar kontainer Docker dari wizard konsol Image Builder.

Daftar Isi

- Buat pipeline gambar kontainer dari AWS CLI
- Perbarui pipeline gambar kontainer dari konsol
- Perbarui pipa gambar kontainer dari AWS CLI

Buat pipeline gambar kontainer dari 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 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, lihatGunakan ekspresi cron di 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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

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 dari 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, lihat Perbarui pipa gambar kontainer dari 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/.
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 dari 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, lihatGunakan 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 setelan 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 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 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 Image Builder

Perbarui pipa gambar kontainer dari 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 Amazon Resource Names (ARNs) 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 di AWS 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. Misalnya, Windows menggunakan garis miring terbalik (\) untuk merujuk ke jalur direktori, sedangkan Linux dan macOS menggunakan garis miring maju (/).

2. Jalankan perintah berikut, menggunakan file yang Anda buat sebagai input.

```
aws imagebuilder update-image-pipeline --cli-input-json file://update-image-pipeline.json
```

Konfigurasikan alur kerja pipa gambar di 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, lihatMengelola alur kerja build dan test untuk image 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 untuk menentukan alur kerja pengujian.

> **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 Anda

Mengatur parameter alur kerja dalam pipeline Image Builder dari 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

Untuk menjalankan alur kerja gambar, Image Builder memerlukan izin untuk melakukan tindakan alur kerja. Anda dapat menentukan peran AWSServiceRoleForImageBuilderterkait 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 CreateImagetindakan, 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, lihatSumber daya dibuat.

Gunakan ekspresi cron di Image Builder

Gunakan ekspresi cron untuk EC2 Image Builder untuk menyiapkan jendela waktu guna menyegarkan gambar Anda dengan pembaruan yang berlaku pada 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 zona waktu Universal Coordinated Time (UTC) secara default, atau Anda dapat menentukan zona waktu. 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 Hari dan Day-of-week bidang, 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), Day-of-week posisi 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 kolom Day-of-month menentuka n hari kerja. Di Day-of-month lapangan, jika Anda memasukkan nomor sebelum`W`, itu berarti Anda ingin menargetkan hari kerja yang paling

Wildcard	Deskripsi
	dekat dengan hari itu. Misalnya, jika Anda menentukan`3W`, 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 `fri#2` untuk bidang hari dalam seminggu.

Pembatasan

- Anda tidak dapat menentukan kolom Day-of-month dan Day-of-week 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 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 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.

Sintaksis

```
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.

Untuk ringkasan peristiwa yang dihasilkan sistem yang dikirimkan Image Builder EventBridge, lihatPesan peristiwa yang dikirim Image Builder.

> **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/.
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 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 Image Builder

EC2 Image Builder terintegrasi dengan AWS Marketplace Layanan AWS dan aplikasi 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 image dasar.
- Komponen Image Builder — 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 Organizations— AWS Organizations memungkinkan Anda untuk 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 dalam. AWS Organizations AWS menyediakan template SCPs untuk skenario umum, seperti memberlakukan batasan pada akun anggota untuk meluncurkan instance dengan hanya disetujui. AMIs
- AWS CloudTrail— Monitor peristiwa Image Builder yang dikirim ke CloudTrail. Untuk informasi selengkapnya tentang CloudTrail integrasi dengan Image Builder, lihatPanggilan API Log Image Builder menggunakan CloudTrail.

  Untuk mempelajari selengkapnya CloudTrail, termasuk cara mengaktifkannya dan menemukan file log Anda, lihat Panduan AWS CloudTrail Pengguna.
- Amazon CloudWatch Logs — Memantau, menyimpan, dan mengakses file log Image Builder Anda dengan CloudWatch. Secara opsional, Anda dapat menyimpan log Anda ke ember S3. Untuk mempelajari lebih lanjut tentang CloudWatch integrasi dengan Image Builder, lihatMemantau log Image Builder dengan Amazon CloudWatch Logs.

  Untuk informasi selengkapnya tentang CloudWatch Log, lihat Apa itu CloudWatch Log Amazon? di Panduan Pengguna CloudWatch Log Amazon.
- Amazon Elastic Container Registry (Amazon ECR) - Amazon ECR adalah AWS layanan registri gambar kontainer terkelola yang aman, terukur, dan andal. Gambar kontainer yang Anda buat dengan Image Builder disimpan di Amazon ECR di Wilayah sumber (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.
- 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 EC2 pengujian 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 License Manager— Anda dapat melampirkan lisensi License Manager yang dikelola sendiri ke output AMI selama proses distribusi. Lisensi yang Anda tentukan untuk Wilayah tujuan harus sudah ada di Wilayah tersebut. Untuk informasi selengkapnya tentang lisensi yang dikelola sendiri, lihat Lisensi yang dikelola sendiri di License Manager.
- 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 Membeli produk di Panduan AWS Marketplace Pembeli.
- AWS Resource Access Manager (AWS RAM) — Dengan AWS RAM, Anda dapat berbagi sumber daya 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 sumber daya berikut: komponen, gambar, dan resep gambar. Untuk informasi selengkapnya AWS RAM, lihat Panduan AWS Resource Access Manager Pengguna. Untuk informasi tentang berbagi sumber daya Image Builder, lihatBagikan sumber daya Image Builder dengan AWS RAM.
- 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

- 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

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 yang masuk untuk memulai tindakan. Misalnya, mengirim acara ke fungsi Lambda saat status EC2 instance 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:

- Memanggil fungsi AWS Lambda
- Memanggil Amazon EC2 Run Command
- Menyampaikan peristiwa ke Amazon Kinesis Data Streams
- Aktifkan mesin AWS Step Functions negara
- 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 aturan diintegrasikan dengan pipeline image Image Builder, lihatMenggunakan EventBridge aturan dengan pipeline Image Builder.

Pesan peristiwa yang dikirim Image Builder

Image Builder mengirimkan pesan peristiwa EventBridge ketika ada perubahan signifikan dalam status untuk sumber daya Image Builder. Misalnya, ketika ada perubahan status untuk gambar. Contoh berikut menunjukkan pesan peristiwa JSON khas yang mungkin dikirim Image Builder.

EC2 Perubahan Status Gambar Image Builder

Image Builder mengirimkan peristiwa ini ketika status berubah untuk sumber daya gambar selama pembuatan gambar. Misalnya, ketika status gambar berubah dari satu status ke negara lain, sebagai berikut:

- Dari `building` ke `testing`
- Dari `testing` ke `distribution`
- Dari `testing` ke `failed`
- Dari `integrating` ke `available`

```
{
    "version": "0",
    "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "detail-type": "EC2 Image Builder Image State Change",
    "source": "aws.imagebuilder",
    "account": "111122223333",
    "time": "2024-01-18T17:50:56Z",
    "region": "us-west-2",
    "resources": ["arn:aws:imagebuilder:us-west-2:111122223333:image/
cmkencryptedworkflowtest-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222/1.0.0/1"],
    "detail": {
        "previous-state": {
```

```
            "status": "TESTING"
        },
        "state": {
            "status": "AVAILABLE"
        }
    }
}
```

EC2 Image Builder CVE Terdeteksi

Jika deteksi CVE diaktifkan untuk gambar Anda, Image Builder mengirimkan pesan dengan hasil setiap kali pemindaian gambar selesai.

```
{
    "version": "0",
    "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "detail-type": "EC2 Image Builder CVE Detected",
    "source": "aws.imagebuilder",
    "account": "111122223333",
    "time": "2023-03-01T16:59:09Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:imagebuilder:us-east-1:111122223333:image/test-image/1.0.0/1",
        "arn:aws:imagebuilder:us-east-1:111122223333:image-pipeline/test-pipeline"
    ],
    "detail": {
        "resource-id": "i-1234567890abcdef0",
        "finding-severity-counts": {
            "all": 0,
            "critical": 0,
            "high": 0,
            "medium": 0
        }
    }
}
```

EC2 Langkah Alur Kerja Image Builder Menunggu

Image Builder mengirimkan pesan saat langkah `WaitForAction` alur kerja berhenti untuk menunggu tindakan asinkron selesai.

```
{
```

```
    "version": "0",
    "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "detail-type": "EC2 Image Builder Workflow Step Waiting",
    "source": "aws.imagebuilder",
    "account": "111122223333",
    "time": "2024-01-18T16:54:44Z",
    "region": "us-west-2",
    "resources": ["arn:aws:imagebuilder:us-west-2:111122223333:image/
workflowstepwaitforactionwithvalidsnstopictest-a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222/1.0.0/1", "arn:aws:imagebuilder:us-west-2:111122223333:workflow/build/
build-workflow-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333/1.0.0/1"],
    "detail": {
        "workflow-execution-id": "wf-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "workflow-step-execution-id": "step-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "workflow-step-name": "TestAutoSNSStop"
    }
}
```

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.

Ketika Amazon Inspector menemukan kerentanan dalam 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.

Konfigurasikan pemindaian keamanan

Jika Anda telah mengaktifkan Amazon Inspector untuk akun Anda, Amazon Inspector secara otomatis memindai EC2 instance 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, lihatKonfigurasikan 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. Ketika penjual mengirimkan produk AWS Marketplace, mereka menentukan harga produk, dan 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?

AWS Marketplace fitur integrasi

Image Builder terintegrasi dengan AWS Marketplace untuk menyediakan kemampuan berikut langsung dari konsol Image Builder:

- Cari produk gambar yang tersedia di AWS Marketplace.
- Cari produk AWS Marketplace gambar yang mengirimkan komponen.
- Lihat daftar langganan AWS Marketplace produk Anda saat ini.
- Gunakan produk AWS Marketplace gambar yang telah Anda berlangganan sebagai gambar dasar untuk resep Image Builder.
- Gunakan AWS Marketplace komponen yang telah Anda berlangganan dalam resep Image Builder.

Image Builder terintegrasi dengan AWS Marketplace untuk menampilkan produk gambar dan komponen yang telah Anda berlangganan. Anda juga dapat mencari produk dan komponen AWS Marketplace gambar dari halaman Temukan produk tanpa meninggalkan konsol Image Builder.

Output AMI yang dibuat Image Builder menyertakan kode produk dari produk dan komponen AWS Marketplace gambar. Anda dapat memiliki hingga empat kode produk untuk gambar akhir yang disesuaikan.

AWS Marketplace langganan di Image Builder

Halaman Langganan di AWS Marketplace bagian konsol Image Builder menampilkan daftar AWS Marketplace produk yang saat ini Anda berlangganan. Setiap produk berlangganan menunjukkan detail berikut:

- Nama produk. Ini terkait dengan halaman detail produk di AWS Marketplace. Halaman detail produk untuk produk berlangganan Anda terbuka di tab baru di browser Anda.
- Penerbit. Ini terkait dengan halaman detail penerbit di AWS Marketplace. Halaman detail penerbit terbuka di tab baru di browser Anda.

- Versi yang Anda berlangganan.
- Jika ada komponen Terkait yang disertakan dengan produk berlangganan Anda, Image Builder menampilkan tautan ke detail 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 gambar berlangganan dalam 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.

Temukan produk AWS Marketplace gambar dari konsol Image Builder

Bagian ini berfokus pada produk AWS Marketplace gambar untuk digunakan sebagai gambar dasar dalam resep Anda. Untuk produk yang menyertakan komponen perangkat lunak terkait, Anda dapat memfilter pemilik produk di konsol dan di API, SDK, dan CLI. Untuk informasi selengkapnya, lihat Daftar komponen Image Builder. Untuk informasi selengkapnya tentang menemukan, berlangganan, dan menggunakan AWS Marketplace komponen, lihatGunakan AWS Marketplace komponen untuk menyesuaikan gambar Anda.

Temukan produk

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/.
2. Dari panel navigasi, pilih Temukan produk di AWS Marketplacebagian.
3. Anda dapat mencari produk gambar di tab Produk gambar di halaman Temukan produk.

   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.

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 terkait dengan 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 di AWS 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.

Gunakan produk AWS Marketplace gambar dalam resep Image Builder

Buka halaman Buat resep dan pilih produk AWS Marketplace gambar untuk digunakan sebagai gambar dasar Anda, sebagai berikut.

1. Buka konsol EC2 Image Builder di https://console.aws.amazon.com/imagebuilder/.
2. Dari panel navigasi, pilih Resep gambar di AWS Marketplacebagian. Ini menunjukkan 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 Marketplacetab. Pilih Tambahkan produk, atau buka AWS Marketplacetab secara langsung. Untuk informasi selengkapnya tentang cara mengatur filter dan mencari di AWS Marketplace, lihatTemukan produk AWS Marketplace gambar dari konsol Image Builder.
6. Masukkan detail yang tersisa seperti biasa. Jika ada atau langganan produk Anda yang menyertakan komponen build, Anda dapat memilihnya dari daftar komponen Build. Pilih `AWS Marketplace` dari daftar tipe pemilik komponen untuk melihatnya, atau pilih `Third party managed` komponen CIS.
7. Pilih Buat resep.

   Gambar akhir Anda dapat berisi hingga empat kode produk dari produk dan komponen AWS Marketplace gambar. Jika gambar dasar dan komponen yang Anda pilih berisi lebih dari empat kode produk, Image Builder mengembalikan kesalahan saat Anda mencoba membuat resep.

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, lihatFormat 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 IAM untuk 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. Di 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": "amzn-s3-demo-bucket"
      }
    },
    "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": 12379400392853802748991242 31,
  "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": "1fecd23a-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 AWS Marketplace kepatuhan dan komponen Image Builder.

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, lihatAmazon mengelola komponen pengerasan STIG untuk Image Builder.

Memantau peristiwa dan log di Image Builder

Untuk menjaga keandalan, ketersediaan, dan kinerja pipeline EC2 Image Builder, 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

- Panggilan API Log Image Builder menggunakan CloudTrail
- Memantau log Image Builder dengan Amazon CloudWatch Logs

Panggilan API Log Image Builder menggunakan CloudTrail

EC2 Image Builder terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Image Builder. CloudTrail menangkap semua panggilan API untuk 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 peristiwa secara terus menerus ke bucket Amazon 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 acara terbaru

di situs Anda Akun AWS. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan Riwayat CloudTrail acara.

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

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

Semua tindakan Image Builder dicatat oleh CloudTrail dan didokumentasikan dalam Referensi API EC2 Image Builder. Misalnya, panggilan ke`CreateImagePipeline`,`UpdateInfrastructureConfiguration`, dan `StartImagePipelineExecution` tindakan menghasilkan entri dalam file CloudTrail log.

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

- Apakah permintaan itu dibuat dengan kredenal pengguna root atau AWS Identity and Access Management (IAM).
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan itu dibuat oleh AWS layanan lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Memantau log Image Builder dengan Amazon CloudWatch Logs

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 (xx.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 Lihat informasi yang lebih lengkap di Memecahkan masalah Image Builder.

Keamanan di Image Builder

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

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang berjalan Layanan AWS di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk EC2 Image Builder, lihat Layanan AWS di Cakupan oleh AWS Layanan Program Kepatuhan .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan 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 dan model tanggung jawab AWS bersama di Image Builder
- Integrasi Identity and Access Management untuk Image Builder
- Sumber daya validasi kepatuhan untuk Image Builder
- Redundansi dan ketahanan data di Image Builder
- Keamanan infrastruktur di Image Builder
- Manajemen Patch untuk gambar Image Builder
- Praktik keamanan terbaik untuk Image Builder

Perlindungan data dan model tanggung jawab AWS bersama di Image Builder

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di EC2 Image Builder. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

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

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat Bekerja dengan CloudTrail jejak di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Image Builder atau lainnya Layanan AWS

menggunakan konsol, API AWS CLI, atau AWS SDKs. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi dan manajemen kunci di 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 khusus 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 untuk Image Builder untuk 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.

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.

AWS CloudTrail acara 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-EXAMPLEaaaaa",
  "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-EXAMPLEzzzzz"
   }
  ],
  "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-EXAMPLEbbbbb",
 "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 Image Builder

Image Builder tidak menyimpan log Anda di layanan. Semua log disimpan di EC2 instans Amazon Anda yang digunakan untuk membuat gambar, atau di log otomatisasi Systems Manager Anda.

Privasi Lalu Lintas Antar Jaringan di Image Builder

Koneksi diamankan antara Image Builder dan lokasi lokal, antara AZs dalam AWS Wilayah, dan antar AWS Wilayah melalui HTTPS. Tidak ada koneksi langsung antar akun.

Integrasi Identity and Access Management untuk Image Builder

Topik

- Audiens
- Mengautentikasi dengan identitas
- Cara Image Builder bekerja dengan kebijakan dan peran IAM
- Kelola perimeter data untuk akses unduhan bucket S3 di Image Builder
- Kebijakan berbasis identitas Image Builder
- Kebijakan berbasis sumber daya Image Builder
- Menggunakan kebijakan AWS terkelola untuk EC2 Image Builder
- Menggunakan peran terkait layanan IAM untuk Image Builder
- Memecahkan masalah IAM di 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, lihatMemecahkan masalah IAM di 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 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, lihatCara Image Builder bekerja dengan kebijakan dan peran 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 Anda Akun AWS, lihat Identitas di Panduan Pengguna IAM.

Cara Image Builder bekerja dengan kebijakan dan peran 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 AWS layanan 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, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya

tindakan tersebut. Anda tidak dapat menentukan secara spesifik prinsipal dalam sebuah kebijakan berbasis identitas karena prinsipal berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

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: Ya

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

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

Tindakan kebijakan untuk Image Builder

Mendukung tindakan kebijakan: Ya

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

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

Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi 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 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 AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, prinsipal manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk

tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

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

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

Untuk melihat daftar jenis sumber daya Image Builder dan jenisnya ARNs, 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 kondisi kebijakan khusus layanan: Yes

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

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

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

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

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Untuk melihat daftar 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

ACLs di Image Builder

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan Image Builder

Mendukung ABAC (tag dalam kebijakan): Sebagian

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut. Dalam AWS, atribut ini disebut tag. Anda dapat melampirkan tag ke entitas IAM (pengguna atau peran) dan ke banyak AWS sumber daya. Penandaan ke entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian rancanglah kebijakan ABAC untuk mengizinkan operasi ketika tanda milik prinsipal cocok dengan tanda yang ada di sumber daya yang ingin diakses.

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

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, atau `aws:TagKeys`.

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

Untuk informasi selengkapnya tentang ABAC, lihat Tentukan izin dengan otorisasi ABAC dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Menggunakan kredensil sementara dengan Image Builder

Mendukung kredensial sementara: Ya

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

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

Anda dapat membuat kredensil sementara secara manual menggunakan API AWS CLI atau AWS . Anda kemudian dapat menggunakan kredensi sementara tersebut untuk mengakses. AWS AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat Kredensial keamanan sementara di IAM.

Izin utama lintas layanan untuk Image Builder

Mendukung sesi akses maju (FAS): Ya

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

Peran layanan untuk Image Builder

Mendukung peran layanan: Ya

Peran layanan adalah peran IAM yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari

dalam IAM. Untuk informasi selengkapnya, lihat Buat sebuah 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 terkait layanan: Ya

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

Untuk detail tentang peran terkait layanan Image Builder, lihat. Menggunakan peran terkait layanan IAM untuk 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 Amazon EC2 Image Builder 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 akses menggunakan kebijakan

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

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

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

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

ARN terdiri dari beberapa node yang membantu mengidentifikasi sumber daya dan memastikan bahwa nama itu unik. Node terakhir dalam nama mencakup beberapa variasi dalam pemformatan untuk jenis sumber daya, nama, dan ID. Saat Image Builder membuat sumber daya, ia menggunakan format berikut:

`arn:aws:imagebuilder:`*`region`*`:`*`owner`*`:`*`resource-type/resource-name/version/build-version`*

> **Note**
>
> Versi build tidak selalu disertakan dalam ARN sumber daya. Namun, beberapa operasi API, seperti GetComponent, memerlukan versi build untuk mengidentifikasi sumber daya secara unik untuk diambil.

Untuk sumber daya yang digunakan Image Builder dalam resepnya, seperti image dasar atau komponen, node pemilik dapat berupa salah satu dari berikut ini:

- Nomor akun pemilik sumber daya
- Untuk sumber daya yang dikelola Amazon: `aws`
- Untuk AWS Marketplace sumber daya: `aws-marketplace`

Contoh berikut menunjukkan ARN untuk komponen terkelola untuk menginstal CloudWatch agen Amazon di Linux:

```
arn:aws:imagebuilder:us-east-1:aws:component/amazon-cloudwatch-agent-linux/1.0.1/1
```

Contoh ini menunjukkan ARN untuk komponen terkelola fiktif dari: AWS Marketplace

```
arn:aws:imagebuilder:us-east-1:aws-marketplace:component/example-linux-software-component/1.0.1
```

Untuk informasi selengkapnya tentang mendapatkan daftar komponen, termasuk penggunaan filter kepemilikan, lihatDaftar komponen Image Builder.

Contoh ARNs

Berikut ini adalah beberapa contoh sumber daya ARNs yang mungkin Anda tentukan dalam kebijakan IAM:

- Contoh ARN

```
"Resource": "arn:aws:imagebuilder:us-
east-1:111122223333:instance/i-1234567890abcdef0"
```

- Wildcard (*) contoh untuk menentukan semua instance untuk akun tertentu

```
"Resource": "arn:aws:imagebuilder:us-east-1:111122223333:instance/*"
```

- Wildcard (*) contoh untuk menentukan semua versi alur kerja gambar terkelola

```
"Resource": "arn:aws:imagebuilder:us-east-1:aws:workflow/build/build-image/*"
```

- Gambar terkelola ARN

```
"Resource": "arn:aws:imagebuilder:us-east-1:aws:image/amazon-linux-2-
arm64/2024.12.17/1"
```

- Wildcard (*) contoh untuk menentukan semua versi dari gambar terkelola

```
"Resource": "arn:aws:imagebuilder:us-east-1:aws:image/amazon-linux-2-arm64/x.x.x"
```

Banyak tindakan EC2 Image Builder API melibatkan banyak sumber daya. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARNs 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 kondisi AWS global, lihat Kunci Konteks Kondisi AWS Global di 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 `CreateInfrastrucutreConfiguration` dan `UpdateInfrastructureConfiguration` APIs.

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 `CreateInfrastrucutreConfiguration` dan `UpdateInfrastructureConfiguration` APIs.

imagebuilder: LifecyclePolicyResourceType

Bekerja dengan operator string.

Gunakan kunci ini untuk memfilter akses menurut jenis sumber daya Siklus Hidup yang ditentukan dalam permintaan.

Nilai untuk kunci ini dapat berupa `AMI_IMAGE` atau`CONTAINER_IMAGE`.

Ketersediaan — Kunci ini hanya tersedia untuk `CreateLifecyclePolicy` dan `UpdateLifecyclePolicy`APIs.

ImageBuilder: EC2 MetadataHttpTokens

Bekerja dengan operator string.

Gunakan kunci ini untuk memfilter akses dengan Persyaratan Token HTTP Metadata EC2 Instans yang ditentukan dalam permintaan.

Nilai untuk kunci ini dapat berupa `optional` atau`required`.

Ketersediaan — Kunci ini hanya tersedia untuk `CreateInfrastrucutreConfiguration` dan `UpdateInfrastructureConfiguration` APIs.

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 `CreateInfrastrucutreConfiguration` dan `UpdateInfrastructureConfiguration` APIs.

Contoh

Untuk melihat contoh kebijakan berbasis identitas Image Builder, lihat. Kebijakan berbasis identitas Image Builder

Kebijakan berbasis sumber daya Image Builder

Kebijakan berbasis sumber daya 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 informasi tentang cara melampirkan kebijakan berbasis sumber daya ke komponen, gambar, atau resep gambar, lihat. Bagikan sumber daya Image Builder dengan AWS RAM

> **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 tanda, berikan informasi tentang tanda di elemen kondisi dari kebijakan menggunakan kunci kondisi `imagebuilder:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`. Untuk informasi selengkapnya tentang menandai resource Image Builder, lihatTag sumber daya dari AWS CLI.

Peran IAM Image Builder

Peran IAM adalah entitas di dalam Anda Akun AWS yang memiliki izin khusus.

Menggunakan kredensil sementara dengan Image Builder

Anda dapat menggunakan kredensial sementara untuk masuk dengan gabungan, menjalankan IAM role, atau menjalankan peran lintas akun. Anda memperoleh kredensil keamanan sementara dengan memanggil operasi AWS STS API seperti AssumeRoleatau. GetFederationToken

Peran terkait 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 IAM untuk 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.

Kelola perimeter data untuk akses unduhan bucket S3 di Image Builder

EC2 Image Builder mengelola dua kelas bucket S3 AWS milik layanan yang berisi sumber daya yang dapat diunduh yang diperlukan untuk menjalankan beban kerja Image Builder di akun Anda.

Jika Anda menggunakan perimeter data untuk mengontrol akses ke Amazon S3 di lingkungan Anda, Anda mungkin perlu secara eksplisit mengizinkan akses ke bucket ini. Anda dapat menggunakan bucket ARN atau URL bucket untuk mengizinkan daftar bucket ini, tergantung pada cara Anda mengontrol akses ke Amazon S3.

Skrip bootstrap manajemen komponen (Wajib)

Bucket S3 ini berisi skrip bootstrap untuk mengatur AWSTOE aplikasi pada EC2 instance yang digunakan untuk membuat gambar. Image Builder memerlukan akses untuk mengunduh skrip untuk mendukung pembuatan dan pengujian gambar baru.

- Ember S3 ARN: `arn:<AWS partition>:s3:::ec2imagebuilder-managed-resources-<AWS Region>-prod`
- URL ember S3: `https://ec2imagebuilder-managed-resources-<AWS Region>.s3.<AWS Region>.<AWS partition-specific domain name>`

Komponen yang dikelola

Bucket S3 ini berisi muatan paket untuk komponen yang dikelola Amazon. Image Builder memerlukan akses untuk mengunduh komponen terkelola apa pun yang dikonfigurasi dalam resep Anda.

- Ember S3 ARN: `arn:<AWS partition>:s3:::ec2imagebuilder-toe-<AWS Region>-prod`
- URL ember S3: `https://ec2imagebuilder-toe-<AWS Region>.s3.<AWS Region>.<AWS partition-specific domain name>`

Kebijakan berbasis identitas 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 membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam 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 melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam 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 kepada mereka:

- AWSImageBuilderReadOnlyAccesskebijakan
- AWSImageBuilderFullAccesskebijakan

Untuk informasi selengkapnya tentang kebijakan terkelola Image Builder, lihatMenggunakan kebijakan AWS 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:

JSON

```
{
    "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 alternatif, hanya izinkan akses ke tindakan yang cocok dengan operasi API yang sedang Anda coba lakukan.

Kebijakan berbasis sumber daya Image Builder

Untuk informasi tentang cara membuat komponen, lihatGunakan komponen untuk menyesuaikan image 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.

`Condition`Blok menggunakan `IpAddress` dan `NotIpAddress` kondisi dan kunci `aws:SourceIp` kondisi, yang merupakan kunci kondisi AWS-wide. Untuk informasi selengkapnya tentang kunci kondisi ini, lihat Menentukan Ketentuan dalam Kebijakan. `aws:sourceIp` IPv4 Nilai-nilai menggunakan notasi CIDR standar. Untuk informasi lebih lanjut, lihat Operator Syarat Alamat IP dalam Panduan Pengguna IAM.

JSON

```
{
  "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 AWS terkelola untuk EC2 Image Builder

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

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

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi

semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam 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 tindakan Amazon EC2 Deskripsikan 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”.

Untuk melihat izin kebijakan ini, lihat AWSImageBuilderFullAccessdi Referensi Kebijakan AWS Terkelola.

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`

Untuk melihat izin kebijakan ini, lihat AWSImageBuilderReadOnlyAccessdi Referensi Kebijakan AWS Terkelola.

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 informasi selengkapnya tentang peran terkait layanan Image Builder, lihat. Menggunakan peran terkait layanan IAM untuk 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, mengambil snapshot, dan mendaftarkan image (AMIs) yang dibuat dan meluncurkan EC2 instance di akun Anda. Image Builder menggunakan snapshot terkait, volume, antarmuka jaringan, subnet, grup keamanan, konfigurasi lisensi, dan pasangan kunci 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 EC2 gambar Amazon, atribut instance, status instans, jenis instans yang tersedia untuk akun Anda, templat peluncuran, subnet, host, dan tag di EC2 sumber daya Amazon Anda.

Image Builder dapat memperbarui pengaturan gambar untuk mengaktifkan atau menonaktifkan peluncuran instance Windows yang lebih cepat di akun Anda, tempat 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, menjelaskan host sebagai dukungan penempatan instance, 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 EC2ImageBuilderDistributionCrossAccountRoledari akun Anda ke akun mana pun yang diizinkan oleh kebijakan Trust tentang peran tersebut. Ini digunakan untuk distribusi gambar lintas akun.

Untuk melihat izin kebijakan ini, lihat AWSServiceRoleForImageBuilderdi Referensi Kebijakan AWS Terkelola.

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 EC2 gambar Amazon 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 EC2 gambar Amazon apa pun di akun.

Untuk melihat izin kebijakan ini, lihat Ec2ImageBuilderCrossAccountDistributionAccessdi Referensi Kebijakan AWS Terkelola.

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 (AMIssnapshot) 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 (AMIs) di akun yang diberi `CreatedBy: EC2 Image Builder` tag.
  - 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`
- AWS Grup 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.

Untuk melihat izin kebijakan ini, lihat EC2ImageBuilderLifecycleExecutionPolicydi Referensi Kebijakan AWS Terkelola.

EC2InstanceProfileForImageBuilderkebijakan

EC2InstanceProfileForImageBuilderKebijakan ini memberikan izin minimum yang diperlukan untuk sebuah EC2 instans untuk 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/`.
- Amazon EC2 — Akses diberikan untuk mendeskripsikan volume dan snapshot, untuk membuat snapshot volume atau sumber daya snapshot yang dibuat Image Builder, dan untuk membuat tag untuk sumber daya Image Builder.
- Image Builder - Akses diberikan untuk mendapatkan Image Builder atau AWS Marketplace komponen 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 `ec2imagebuilder-` dengan, atau sumber daya yang memiliki ekstensi file ISO.

Untuk melihat izin kebijakan ini, lihat EC2InstanceProfileForImageBuilderdi Referensi Kebijakan AWS Terkelola.

EC2InstanceProfileForImageBuilderECRContainerBuildskebijakan

EC2InstanceProfileForImageBuilderECRContainerBuildsKebijakan ini memberikan izin minimum yang diperlukan untuk EC2 instance 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-`

Untuk melihat izin kebijakan ini, lihat EC2InstanceProfileForImageBuilderECRContainerBuildsdi Referensi Kebijakan AWS Terkelola.

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
AWSServiceRoleForImageBuilder — Permbaruan ke kebijakan yang sudah ada	Image Builder membuat perubahan berikut pada peran layanan untuk mendukung impor file ISO OS klien Microsoft sebagai image dasar. • Menambahkan ec2: RegisterImage untuk memungkinkan Image Builder membuat snapshot dan membuat serta mendaftarkan AMI yang sistem operasi dasarnya diimpor dari file disk ISO terverifikasi.	Desember 30, 2024

Perubahan	Deskripsi	Tanggal
EC2InstanceProfileForImageBuilder – Pembaruan ke kebijakan yang ada	Image Builder membuat perubahan berikut pada kebijakan profil instance untuk mendukung pembuatan gambar dari file image disk. • Menambahkan tindakan ec2 berikut: DescribeVolumes dan DescribeSnapshots pada semua sumber daya untuk mengambil detail. Juga menambahk an tindakan berikut untuk sumber daya yang dibuat oleh Image Builder: CreateSnapshot untuk sumber daya volume dan snapshot, dan CreateTags. Ditambahkan S3: GetObject untuk sumber daya dengan ekstensi file “ISO”.	Desember 30, 2024
EC2InstanceProfileForImageBuilder— Kebijakan yang diperbarui	Image Builder memperbar ui `EC2InstanceProfile ForImageBuilder` kebijakan untuk memungkin kan Image Builder mendapatk an AWS Marketplace komponen.	Desember 2, 2024

Perubahan	Deskripsi	Tanggal
EC2ImageBuilderLifecycleExecutionPolicy – Kebijakan baru	Image Builder menambahkan `EC2ImageBuilderLifecycleExecutionPolicy` kebijakan baru yang berisi izin untuk manajemen siklus hidup gambar.	17 November 2023
AWSServiceRoleForImageBuilder – Pembaruan ke kebijakan yang ada	Image Builder membuat perubahan berikut pada peran layanan untuk memberikan dukungan penempatan instance. • Menambahkan ec2:DescribeHosts memungkinkan 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.	19 Oktober 2023

Perubahan	Deskripsi	Tanggal
AWSServiceRoleForImageBuilder – Pembaruan ke kebijakan yang ada	Image Builder membuat perubahan berikut pada peran layanan untuk memberikan dukungan penempatan instance. • 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.	28 September 2023

Perubahan	Deskripsi	Tanggal
AWSServiceRoleForImageBuilder – Pembaruan ke kebijakan yang ada	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. • Menambahkan inspector2:ListCoverage dan inspector2: 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
	tambahan yang memerlukan nama repositori untuk memulai. `image-builder-*` Batasan nama mencegah eskalasi hak istimewa dan mencegah perubahan pada repositori yang tidak dibuat oleh Image Builder. • Ditambahkan ecr:BatchDeleteImage untuk repositori ECR ditandai dengan. `CreatedBy: EC2 Image Builder` Izin ini memerlukan nama repositori untuk memulai. `image-builder-*` • Menambahkan izin acara untuk Image Builder untuk membuat dan mengelola aturan EventBridge terkelola Amazon yang disertakan `ImageBuilder-*` dalam nama.	

Perubahan	Deskripsi	Tanggal
AWSServiceRoleForImageBuilder – Pembaruan ke kebijakan yang ada	Image Builder membuat perubahan berikut pada peran layanan: • Menambahkan lisensi License Manager sebagai sumber daya untuk RunInstance panggilan ec2: untuk memungkinkan pelanggan menggunakan image dasar AMIs yang terkait dengan konfigurasi lisensi.	22 Maret 2022
AWSServiceRoleForImageBuilder – Pembaruan ke kebijakan yang ada	Image Builder membuat perubahan berikut pada peran layanan: • Menambahkan izin untuk tindakan EC2 EnableFastLaunch API, 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
AWSServiceRoleForImageBuilder – Pembaruan ke kebijakan yang ada	Image Builder membuat perubahan berikut pada peran layanan: • 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.	20 November 2021
AWSServiceRoleForImageBuilder – Pembaruan ke kebijakan yang ada	Image Builder menambahkan izin baru untuk memperbaiki masalah di mana lebih dari satu asosiasi inventaris menyebabkan pembuatan gambar macet.	Agustus 11, 2021

Perubahan	Deskripsi	Tanggal
AWSImageBuilderFullAccess – Pembaruan ke kebijakan yang ada	Image Builder membuat perubahan berikut pada peran akses penuh: • Menambahkan izin untuk mengizinkan `ec2:DescribeInstanceTypeOfferings` . • Menambahkan izin untuk memanggil `ec2:DescribeInstanceTypeOfferings` 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 IAM untuk Image Builder

EC2 Image Builder menggunakan AWS Identity and Access Management 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 terkait layanan untuk layanan tersebut.

Izin peran terkait layanan untuk Image Builder

Image Builder menggunakan peran AWSServiceRoleForImageBuilderterkait 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-imagedalam AWS CLI).
  - Tindakan ImportVmImage API (import-vm-imagedalam AWS CLI).
  - Tindakan StartImagePipelineExecution API (start-image-pipeline-executiondalam 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, 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, the AWS 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 file AWS 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, lihatHapus sumber daya Image Builder yang sudah usang atau tidak digunakan.

Hapus peran terkait layanan menggunakan IAM

Anda dapat menggunakan konsol IAM, the AWS 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 Image Builder

Image Builder mendukung penggunaan peran terkait layanan di semua AWS Wilayah tempat layanan tersedia. Untuk daftar AWS Wilayah yang didukung, lihatAWS Wilayah dan Titik Akhir.

Memecahkan masalah IAM di 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 kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan 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 memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

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.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol 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 memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses sumber daya Image Builder saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mengetahui apakah Image Builder mendukung fitur ini, lihatCara Image Builder bekerja dengan kebijakan dan peran IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Sumber daya validasi kepatuhan untuk Image Builder

EC2 Image Builder tidak dalam lingkup program AWS kepatuhan apa pun.

Untuk daftar cakupan program kepatuhan tertentu, lihat AWS Layanan dalam Lingkup menurut Program Kepatuhan Layanan AWS . Layanan AWS Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di Laporan AWS Pengunduhan Artefak .

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.
- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya 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.

Redundansi dan ketahanan data di Image Builder

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang 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. AMIs 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 High Availability (HA). Anda dapat mendistribusikan gambar ke beberapa Wilayah untuk membuat gambar lebih tersedia.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat Infrastruktur AWS Global.

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 kredensil keamanan sementara untuk permintaan Anda.

Selain itu, EC2 instance yang digunakan Image Builder untuk membangun dan menguji gambar harus memiliki akses ke AWS Systems Manager.

Manajemen Patch untuk gambar Image Builder

AWS menyediakan dikelola diperbarui AMIs setiap bulan yang memiliki pembaruan terbaru dan patch keamanan diterapkan untuk sistem operasi berikut. Anda dapat menggunakan ini AMIs sebagai gambar dasar untuk penyesuaian Anda. Untuk informasi selengkapnya, lihat Sistem operasi yang didukung.

- Distribusi Linux termasuk Amazon Linux 2, AL2 023, Red Hat Enterprise Linux (RHEL), CentOS, Ubuntu, SUSE Linux Enterprise Server
- Windows Server 2016 dan setelahnya
- macOS 10.14.x dan yang lebih baru

Setelah Anda membuat gambar khusus, Anda bertanggung jawab atas penambalan EC2 sistem Amazon, sesuai dengan model tanggung jawab bersama. Jika EC2 instance dalam beban kerja aplikasi Anda dapat dengan mudah diganti, maka mungkin paling efisien untuk memperbarui AMI dasar dan menerapkan ulang semua node komputasi berdasarkan gambar ini.

> **Note**
>
> Untuk patch macOS, kami sarankan Anda membuat versi baru resep Anda yang menggunakan AMI terkelola terbaru untuk gambar dasar, lalu buat gambar kustom yang diperbarui dari resep dan sumber daya pembuatan gambar Anda yang lain. Jika instans Mac Anda tidak mudah diganti, lihat halaman Perbarui sistem operasi dan perangkat lunak pada instans Mac di Panduan EC2 Pengguna Amazon untuk informasi selengkapnya.

Berikut ini adalah dua cara agar Image Builder tetap AMIs up to date.

- AWS-komponen patching yang disediakan - EC2 Image Builder menyediakan komponen build berikut yang menginstal semua pembaruan sistem operasi yang tertunda:
  - `update-linux`
  - `update-windows`

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 yang didukung AMIs, 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 saja. Modul `UpdateOS` tindakan tidak didukung untuk macOS.)

```
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 keamanan terbaik untuk 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 tidak memadai untuk lingkungan Anda, perlakukan itu sebagai pertimbangan yang bermanfaat, bukan sebagai resep.

- 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.
- Terapkan pembaruan AMI terkelola secara berkala ke resep macOS Anda dan buat gambar baru untuk meluncurkan instance yang memiliki tambalan keamanan terbaru.

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 saluran pipa Image Builder

Saat pipeline Image Builder Anda berjalan, ia mengirimkan permintaan HTTP untuk meluncurkan EC2 instance 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 EC2 instance yang diluncurkan Image Builder dari build pipeline untuk digunakan IMDSv2 sehingga permintaan pengambilan metadata instance memerlukan header token yang ditandatangani.

Untuk informasi selengkapnya tentang konfigurasi infrastruktur Image Builder, lihatKelola konfigurasi infrastruktur Image Builder. Untuk informasi selengkapnya tentang opsi metadata EC2 instance untuk image Linux, lihat Mengonfigurasi opsi metadata instans di Panduan Pengguna Amazon. EC2 Untuk gambar Windows, lihat Mengonfigurasi opsi metadata instans di EC2 Panduan Pengguna Amazon.

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_cloudinit_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.aliasesdb-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.

macOS

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 rm -f $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_cloudinit_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 rm -f {} \;
  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.aliasesdb-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 rm -f {} \;
  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"
  "/var/log/amazon/ec2/ec2-macos-init.log"
  "/var/log/amazon/ec2/ena-ethernet.log"
  "/var/log/amazon/ec2/system-monitoring.log"
)
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 rm -f {} \;
  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 rm -f {} \;
  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 rm -f /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 rm -f /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 rm -f {} \;
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 rm -f /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 rm -f {} \;
      sudo rm -rf /var/log/journal/*
fi

sudo touch /etc/machine-id
```

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**
>
> Melewatkan 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.

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 bernama`skip_cleanup_cloudinit_files.txt`, Image Builder tidak akan mengenali file skip.

Input

Bersihkan bagian	File dihapus	Lewati nama file bagian

`CLOUD_INIT_FILES`	`/etc/sudoers.d/90-cloud-init-users`
	`/etc/locale.conf`
	`/var/log/cloud-init.log`
	`/var/log/cloud-init-output.log`
		`skip_cleanup_cloudinit_files`

`INSTANCE_FILES`	`/etc/.updated`
	`/etc/aliases.db`
	`/etc/hostname`
		`skip_cleanup_instance_files`

Bersihkan bagian	File dihapus	Lewati nama file bagian
	/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/.gpgkeyschecked.yum	
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 /home/<all users>/.ssh/authorized_keys;	skip_cleanup_ssh_files

Bersihkan bagian	File dihapus	Lewati nama file bagian
INSTANCE_LOG_FILES	/var/log/audit/audit.log /var/log/boot.log /var/log/dmesg /var/log/cron	skip_cleanup_instance_log_files
TOE_FILES	{{workingDirectory}}/TOE_*	skip_cleanup_toe_files
SSM_LOG_FILES	/var/log/amazon/ssm/*	skip_cleanup_ssm_log_files

Memecahkan masalah Image Builder

EC2 Image Builder terintegrasi dengan Layanan AWS 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 GetWorkflowExecutiondan ListWorkflowStepExecutionsAPI 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 EC2 instans Amazon yang dikembalikan di 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 EC2 instance 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 ini`application.log`, dimulai dengan, untuk mendiagnosis penyebab masalah pada instance.

Secara default, Image Builder mematikan instance EC2 build atau pengujian Amazon yang berjalan saat pipeline gagal. Anda dapat mengubah setelan instans untuk sumber daya konfigurasi infrastruktur yang digunakan pipeline, untuk mempertahankan instans 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. Lihat perinciannya di 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 pada saat yang sama.

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 APIs 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 menyiapkan PrivateLink titik akhir, lihat Mengakses AWS layanan melalui AWS PrivateLink.

- Izin tidak ada - Tambahkan kebijakan terkelola berikut ke peran terkait layanan IAM untuk Image Builder:
  - EC2InstanceProfileForImageBuilder
  - EC2InstanceProfileForImageBuilderECRContainerMembangun
  - Amazon SSMManaged InstanceCore

  Untuk informasi selengkapnya tentang peran terkait layanan Image Builder, lihat. Menggunakan peran terkait layanan IAM untuk 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 sedang 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 jenis 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 instance diubah 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

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 ImageRecipedata.

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 EC2 instans. 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, lihatPerbarui pipeline gambar AMI dari konsol. Untuk informasi selengkapnya tentang memperbarui pipeline image container menggunakan konsol EC2 Image Builder, lihatPerbarui pipeline gambar kontainer dari konsol.

- Anda juga dapat memperbarui pipeline gambar Anda dengan update-image-pipeline perintah di file AWS 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 `EnhancedImageMetadataEnabled` properti dalam file JSON `false` saat Anda membuat pipeline menggunakan. AWS CLI

Riwayat dokumentasi perubahan pada panduan pengguna Image Builder

Tabel berikut menjelaskan perubahan penting pada dokumentasi berdasarkan tanggal. Untuk notifikasi tentang pembaruan dokumentasi ini, Anda dapat berlangganan ke umpan RSS.

- Versi API: 2025-04-30

Perubahan	Deskripsi	Tanggal
Pembaruan STIG Q2 2025	Memperbarui versi Windows STIG dan menerapkan STIGS untuk rilis kuartal kedua 2025.	Juni 26, 2025
Pembaruan STIG Q1 2025	Memperbarui versi Windows STIG dan menerapkan versi komponen STIGS dan SCAP untuk rilis kuartal pertama 2025.	5 Mei 2025
Rilis fitur: Mendukung Parameter SSM	Image Builder sekarang mendukung penggunaan Parameter Penyimpanan Parameter AWS Systems Manager(SSM) dalam resep dan selama distribusi gambar.	April 30, 2025
Pembaruan STIG Q4	Memperbarui versi Windows STIG dan menerapkan STIGS untuk rilis kuartal keempat 2024.	Februari 4, 2025
Pembaruan kebijakan IAM: kebijakan peran layanan	Memperbarui kebijakan peran terkait layanan untuk mendukung pembuatan gambar dari file ISO OS klien	Desember 30, 2024

	Microsoft resmi yang diimpor. Untuk informasi selengkap nya, lihat AWSServiceRoleForImageBuilderkebijakan.	
Pembaruan kebijakan IAM: kebijakan profil instance	Memperbarui kebijakan peran profil instance untuk mendukung pembuatan gambar dari file gambar disk. Untuk informasi selengkap nya, lihat EC2InstanceProfileForImageBuilderkebijakan.	Desember 30, 2024
Rilis fitur: ISO ke AMI	Image Builder sekarang dapat membuat gambar dari file disk ISO resmi untuk Microsoft Windows 11 dan sistem operasi klien yang lebih baru.	Desember 30, 2024
Pembaruan STIG Q4	Memperbarui versi STIG Linux dan menerapkan STIGS untuk rilis kuartal keempat 2024. Menambahkan informasi tentang dua parameter input baru untuk komponen Linux.	Desember 10, 2024
Pembaruan kebijakan IAM: kebijakan profil instance	Memperbarui kebijakan profil instance untuk memberika n akses untuk mendapatkan AWS Marketplace komponen. Untuk informasi selengkap nya, lihat EC2InstanceProfileForImageBuilderkebijakan.	Desember 2, 2024
Rilis fitur: AWS Marketplace komponen	Menambahkan dukungan untuk komponen AWS Marketplace perangkat lunak.	Desember 2, 2024

Rilis fitur: dukungan macOS	Menambahkan dukungan untuk gambar macOS.	Oktober 22, 2024
Dukungan dokumen untuk operator logis di AWS Task Orchestrator and Executor	Gunakan operator logika untuk menambahkan atau memodifikasi ekspresi bersyarat dalam dokumen komponen Anda.	Agustus 16, 2024
Dukungan dokumen untuk konstruksi bersyarat di AWS Task Orchestrator and Executor	Gunakan konstruksi bersyarat seperti pernyataan “if” untuk mengarahkan aliran kontrol untuk tindakan komponen dalam dokumen komponen Anda.	Agustus 16, 2024
Dukungan dokumen untuk operator perbandingan di AWS Task Orchestrator and Executor	Gunakan operator perbandingan untuk membandingkan nilai dalam dokumen komponen Anda.	Agustus 16, 2024
Ditambahkan modul tindakan Assert	Ditambahkan dokumentasi untuk modul `ExecuteDocument` tindakan di bawah`General execution` .	Agustus 14, 2024
Dukungan sistem operasi	Menambahkan dukungan untuk RHEL 9 dan Ubuntu 24.04 LTS.	Agustus 2, 2024
Pembaruan dokumen: Mengatur ulang konten	Dokumentasi yang direorganisasi untuk meningkatkan presentasi dan navigasi.	Juni 21, 2024

Pembaruan STIG Q2	Memperbarui versi STIG Linux dan menerapkan STIGS untuk rilis kuartal kedua 2024. Menambahkan dukungan untuk RHEL 9, CentOS Stream 9, dan Ubuntu 22.04. Tidak ada perubahan pada versi Windows.	10 Mei 2024
Pembaruan STIG Q1	Memperbarui versi STIG Linux dan menerapkan STIGS untuk rilis kuartal pertama 2024. Tidak ada perubahan pada versi Windows.	Februari 23, 2024
Pembaruan STIG Q1	Memperbarui versi STIG Linux dan menerapkan STIGS untuk rilis kuartal pertama 2024. Tidak ada perubahan pada versi Windows.	Februari 23, 2024
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

Pembaruan STIG Q4	Memperbarui versi STIG Linux dan menerapkan STIGS untuk rilis kuartal keempat 2023. Tidak ada perubahan pada versi Windows. Juga memperbarui Linux dan Windows SCAP untuk komponen baru, perangkat lunak, dan nomor benchmark.	Desember 7, 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 kebijakan IAM: peran layanan	Memperbarui kebijakan peran terkait layanan untuk dukungan penempatan misalnya. Untuk informasi selengkapnya, lihat AWSServiceRoleForImageBuilderkebijakan.	19 Oktober 2023

Pembaruan STIG Q3	Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal ketiga 2023. Selain itu pesan yang diperbarui untuk mengklarifikasi bahwa paket pihak ketiga tidak diinstal secara otomatis, dengan sangat sedikit pengecualian. Semua yang dilewati STIGs dicatat.	5 Oktober 2023
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 AL2 023.	April 14, 2023
Perbarui Wilayah yang didukung untuk AWSTOE	AWSTOE Menambahkan dukungan untuk hal-hal berikut Wilayah AWS: Asia Pasifik (Hyderabad), Asia Pasifik (Jakarta), Eropa (Zurich), Eropa (Spanyol), dan Timur Tengah (UEA).	13 April 2023
AWSTOE pembaruan 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.	31 Maret 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.	1 Februari 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.	13 Januari 2023
Komponen pengerasan CIS	Ditambahkan komponen pengerasan CIS yang dimiliki dan dipelihara oleh CIS.	13 Januari 2023
Versi STIG Baru	Memperkenalkan dukungan Ubuntu, versi STIG yang diperbarui, dan menerapka n 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.	7 Juni 2022
Versi STIG Baru	Versi STIG yang diperbarui dan menerapkan STIGS untuk rilis kuartal pertama 2022.	April 25, 2022
Ditambahkan modul ExecuteDocument tindakan	Ditambahkan dokumenta si untuk modul `ExecuteDo cument` tindakan di bawah`General execution` .	28 Maret 2022

Rilis fitur: Support untuk peluncuran Windows AMI yang lebih cepat	Menambahkan pengaturan konfigurasi distribusi untuk mendukung peluncuran lebih cepat untuk Windows AMIs.	Februari 21, 2022
Rilis pemeliharaan: Perbarui sidik AWSTOE jari biner	Cetak jempol biner yang diperbarui untuk sertifikat AWSTOE penandatangan.	18 Februari 2022
Rilis fitur: Konfigurasikan input untuk AWSTOE	Ditambahkan dukungan untuk menggunakan file konfigura si JSON sebagai input untuk AWSTOE run perintah.	3 Februari, 2022
Versi STIG Baru	Versi STIG yang diperbaru i dan menerapkan STIGS untuk rilis kuartal keempat 2021. Juga menambahkan bagian untuk komponen SCAP Compliance Checker (SCC) baru.	Desember 22, 2021
Rilis fitur: Integrasi VM Import/ Export (VMIE)	Menambahkan dukungan untuk impor VM melalui semua saluran (konsol,API/CLI, etc.), and for VM export via API/ CLI. Ekspor VM saat ini tidak tersedia dari konsol Image Builder.	Desember 20, 2021
Rilis fitur: AMI berbagi untuk AWS Organizations dan OUs	Konfigurasi distribusi yang diperbarui untuk menambahk an dukungan untuk berbagi output AMIs dengan AWS Organizations dan OUs.	24 November 2021

Pembaruan dokumen: Perbarui tahapan dan fase komponen	Konten yang diperluas untuk tahapan komponen di Image Builder, dan bagaimana konten tersebut berintera ksi 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.	10 September 2021
Rilis fitur: EventBridge Integrasi Amazon	Menambahkan EventBridge dukungan yang memungkin kan Anda menghubun gkan Image Builder dengan peristiwa dari terkait Layanan AWS, dan memulai peristiwa berdasarkan aturan yang ditentukan dalam EventBridge.	18 Agustus 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 yang diperluas dari EC2 instance 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 disempurnakan seputar penandaan sumber daya.	25 Juni 2021
Rilis fitur: Luncurkan integrasi template	Menambahkan dukungan untuk menggunakan templat EC2 peluncuran Amazon untuk distribusi AMI di pengaturan Distribusi.	7 April 2021
Rilis fitur: Peningkatan pembuatan kontainer	Menambahkan dukungan untuk mengonfigurasi pemetaan perangkat blok dan menentukan AMIs 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 STIGs 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
Terenkripsi AMIs	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 secara pribadi APIs 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. APIs 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.